

The background of the cover features a technical line drawing of several robotic arms. The top half of the image has an orange background, while the bottom half is white. The robotic arms are depicted in a stylized, schematic manner, with various joints, actuators, and end-effectors. Some arms are shown in a more detailed, shaded style, while others are simple line drawings. The overall composition suggests a focus on robotics and automation.

DESIGNING SELF-ORGANIZATION IN THE PHYSICAL REALM

EDITED BY: Heiko Hamann, Melanie Schranz, Wilfried Elmenreich, Vito Trianni,
Carlo Pincioli, Nicolas Bredeche and Eliseo Ferrante

PUBLISHED IN: Frontiers in Robotics and AI



frontiers

Frontiers eBook Copyright Statement

The copyright in the text of individual articles in this eBook is the property of their respective authors or their respective institutions or funders. The copyright in graphics and images within each article may be subject to copyright of other parties. In both cases this is subject to a license granted to Frontiers.

The compilation of articles constituting this eBook is the property of Frontiers.

Each article within this eBook, and the eBook itself, are published under the most recent version of the Creative Commons CC-BY licence.

The version current at the date of publication of this eBook is CC-BY 4.0. If the CC-BY licence is updated, the licence granted by Frontiers is automatically updated to the new version.

When exercising any right under the CC-BY licence, Frontiers must be attributed as the original publisher of the article or eBook, as applicable.

Authors have the responsibility of ensuring that any graphics or other materials which are the property of others may be included in the CC-BY licence, but this should be checked before relying on the CC-BY licence to reproduce those materials. Any copyright notices relating to those materials must be complied with.

Copyright and source acknowledgement notices may not be removed and must be displayed in any copy, derivative work or partial copy which includes the elements in question.

All copyright, and all rights therein, are protected by national and international copyright laws. The above represents a summary only. For further information please read Frontiers' Conditions for Website Use and Copyright Statement, and the applicable CC-BY licence.

ISSN 1664-8714
ISBN 978-2-88966-311-8
DOI 10.3389/978-2-88966-311-8

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews. Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: researchtopics@frontiersin.org

DESIGNING SELF-ORGANIZATION IN THE PHYSICAL REALM

Topic Editors:

Heiko Hamann, University of Lübeck, Germany

Melanie Schranz, Lakeside Labs GmbH, Austria

Wilfried Elmenreich, University of Klagenfurt, Austria

Vito Trianni, Italian National Research Council, Italy

Carlo Pinciroli, Worcester Polytechnic Institute, United States

Nicolas Bredeche, Université Pierre et Marie Curie, France

Eliseo Ferrante, Vrije Universiteit Amsterdam, Netherlands

Citation: Hamann, H., Schranz, M., Elmenreich, W., Trianni, V., Pinciroli, C., Bredeche, N., Ferrante, E., eds. (2021). Designing Self-Organization in the Physical Realm. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88966-311-8

Table of Contents

04	<i>Editorial: Designing Self-Organization in the Physical Realm</i> Heiko Hamann, Melanie Schranz, Wilfried Elmenreich, Vito Trianni, Carlo Pinciroli, Nicolas Bredeche and Eliseo Ferrante
07	<i>Automatic Off-Line Design of Robot Swarms: A Manifesto</i> Mauro Birattari, Antoine Ligot, Darko Bozhinoski, Manuele Brambilla, Gianpiero Francesca, Lorenzo Garattoni, David Garzón Ramos, Ken Hasselmann, Miquel Kegeleirs, Jonas Kuckling, Federico Pagnozzi, Andrea Roli, Muhammad Salman and Thomas Stützle
13	<i>A Comparison of Individual Learning and Social Learning in Zebrafish Through an Ethorobotics Approach</i> Yanpeng Yang, Romain J. G. Clément, Stefano Ghirlanda and Maurizio Porfiri
26	<i>A Survey on Swarming With Micro Air Vehicles: Fundamental Challenges and Constraints</i> Mario Coppola, Kimberly N. McGuire, Christophe De Wagter and Guido C. H. E. de Croon
52	<i>Attraction, Dynamics, and Phase Transitions in Fire Ant Tower-Building</i> Gary K. Nave Jr., Nelson T. Mitchell, Jordan A. Chan Dick, Tyler Schuessler, Joaquin A. Lagarrigue and Orit Peleg
65	<i>Phenotypic Plasticity Provides a Bioinspiration Framework for Minimal Field Swarm Robotics</i> Edmund R. Hunt
74	<i>Swarm Robotic Behaviors and Current Applications</i> Melanie Schranz, Martina Umlauft, Micha Sende and Wilfried Elmenreich
94	<i>Guiding the Self-Organization of Cyber-Physical Systems</i> Carlos Gershenson
107	<i>Scalable and Robust Fabrication, Operation, and Control of Compliant Modular Robots</i> Nialah Jenae Wilson, Steven Ceron, Logan Horowitz and Kirstin Petersen
125	<i>Mutual Shaping in Swarm Robotics: User Studies in Fire and Rescue, Storage Organization, and Bridge Inspection</i> Daniel Carrillo-Zapata, Emma Milner, Julian Hird, Georgios Tzoumas, Paul J. Vardanega, Mahesh Sooriyabandara, Manuel Giuliani, Alan F. T. Winfield and Sabine Hauert
144	<i>Blockchain Technology Secures Robot Swarms: A Comparison of Consensus Protocols and Their Resilience to Byzantine Robots</i> Volker Strobel, Eduardo Castelló Ferrer and Marco Dorigo
166	<i>Sparse Robot Swarms: Moving Swarms to Real-World Applications</i> Danesh Tarapore, Roderich Groß and Klaus-Peter Zauner
175	<i>Adaptive Foraging in Dynamic Environments Using Scale-Free Interaction Networks</i> Ilja Rausch, Pieter Simoens and Yara Khaluf



Editorial: Designing Self-Organization in the Physical Realm

Heiko Hamann^{1*}, Melanie Schranz², Wilfried Elmenreich³, Vito Trianni⁴, Carlo Pinciroli⁵, Nicolas Bredeche⁶ and Eliseo Ferrante⁷

¹ Institute of Computer Engineering, University of Lübeck, Lübeck, Germany, ² Lakeside Labs GmbH, Klagenfurt, Austria, ³ Faculty of Technical Sciences, Institute of Networked and Embedded Systems, University of Klagenfurt, Klagenfurt, Austria, ⁴ Institute of Cognitive Sciences and Technologies, Italian National Research Council, Rome, Italy, ⁵ Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA, United States, ⁶ Institut des Systèmes Intelligents et de Robotique, Université Pierre et Marie Curie, Paris, France, ⁷ Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, Netherlands

Keywords: self-organization, cyber-physical system, swarm robotics, resilience, scalability

Editorial on the Research Topic

Designing Self-Organization in the Physical Realm

The design and deployment of decentralized systems can benefit from self-organization as it introduces key features, such as resilience, scalability, and adaptivity to dynamic environments. However, whenever self-organization was demonstrated on physical platforms (e.g., robot swarms), this was performed mostly within controlled laboratory conditions. The real world comes with severe requirements, calling for robust design methodologies, their standardization, and validation via benchmarking toolsets. With this Research Topic, we collect, benchmark, and survey novel approaches to push self-organization toward real-world applications, focusing on embodied artificial systems, such as multi-robot, cyber-physical, and socio-technical systems.

We start with six perspective and survey papers that give a good overview of the state of the art and challenges of real-world implementations.

Gershenson studies the complexity of cyber-physical systems. After reviewing basic concepts that are useful to design self-organizing systems, he introduces approaches to implement self-organization in cyber-physical systems. Gershenson reviews three case studies from different domains. Crowd control is related to a passive control approach using signs to mediate passenger boarding and descent in Mexico City Metro. In a traffic light case study, traffic lights and vehicles interact closely as agents, resulting in a network of streets and crossings with self-organized coordination of traffic flows. The third case study is related to public transport and addresses the equal headway instability. Trains use bio-inspired pheromone systems to keep equal distance to the vehicles in front and behind. The result is a flexible system where trains can quickly adapt and respond to service delays. Gershenson provides an outlook for cyber-physical and cyber-social systems controlled by guided self-organization.

Based on the above-mentioned benefits of self-organization the motivation is strong to apply swarm robotics in industrial applications. However, many industrial applications still rely on centralized control. In cases where a multi-robot solution is employed, the main idea of swarm robotics of distributed decision-making is often not implemented. Schranz et al. provide a collection and categorization of swarm robotic behaviors. The paper gives a comprehensive overview of research platforms and industrial projects and products, separated into terrestrial, aerial, aquatic, and outer space. The authors identify several open issues including dependability, emergent characteristics, security and safety, and communication as hindrances for the implementation of fully distributed autonomous swarm systems.

OPEN ACCESS

Edited and reviewed by:

Herbert Glenn Tanner,
University of Delaware, United States

*Correspondence:

Heiko Hamann
hamann@iti.uni-luebeck.de

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 22 August 2020

Accepted: 08 October 2020

Published: 02 November 2020

Citation:

Hamann H, Schranz M, Elmenreich W,
Trianni V, Pinciroli C, Bredeche N and
Ferrante E (2020) Editorial: Designing
Self-Organization in the Physical
Realm. *Front. Robot. AI* 7:597859.
doi: 10.3389/frobt.2020.597859

To deploy swarm robots to the physical realm, one requirement is the ability to cope with environments that lack human infrastructures. Two key mechanisms, namely cognition and sensing, have to take place “on-board” on the robot and should not be offloaded to external devices. Physical mobile robots that operate on land do have the required hardware capabilities for onboard computation and sensing, and have successfully been used to demonstrate basic collective behaviors and to a more limited extent been used in real applications. However, Coppola et al. convincingly argue that swarm robotics approaches so far cannot be applied to Micro Aerial Vehicles (MAVs). The most impressive MAVs demonstrations have been executed requiring external computation, sensing, or both. The main challenge is related to local sensing, which they divide into the following sub-challenges: MAV hardware design, ego-state estimation, intra-swarm relative sensing, and swarm behaviors. This paper presents how advanced we are in terms of autonomy of swarms of MAVs, and presents a roadmap to overcome the challenges in the near future.

One of the main challenges for the design of self-organizing systems is the gap between the rules followed by individual system components and the desired collective behavior of the system as a whole. Especially for practical application scenarios, it is difficult to conceive and optimize the system behavior by acting at the level of the individual rules. The paper by Birattari et al. champions a methodology that optimizes the system behavior offline (e.g., in simulation) and that ensures sufficient performance when deployed in the real world. The central aspect is the “class of interest” of the problems to be addressed. Every new problem instance is sampled from the same class of interest (e.g., gardening with robot swarms), and the solution is optimized to maximize performance, according to relevant metrics defined for the given class. It is within the same class of interest that the offline automatic design approach gives its best results, and the manifesto highlights the most important questions that should drive future research in this area.

The following eight papers study concepts, methods, hardware designs, and natural systems with high potential to support future real-world applications of self-organizing systems.

There have been many contributions using either simulation or relatively simple robots, often in controlled environments of limited size. Tarapore et al. question the very definition of swarm robotics by focusing on the question of how sparse is a robot swarm for a realistic task. Tarapore et al. argue that real swarm robotics applications will need to be addressed, and they introduce the idea of “sparse swarm robotics”: robots are spread over the environment such that the opportunity for communication must be explicitly addressed, as opposed to being naturally forced in smaller environment where density is high. They propose a clean and straight-forward formalization of this problem in mathematical terms. Also, they illustrate the concept of sparse swarm robotics by describing several realistic problems and their implications, including a step-by-step description of the specific issues that arise for one such problem. Considering a monitoring task for soil sampling in a forest, they discuss both low-level hardware issues and high-level communication/coordination issues.

A particular threat for real-world robot swarms is a possible attack by malicious agents that could be introduced into the swarm. The paper by Strobel et al. makes a significant contribution toward the use of swarm robotics in the real world by presenting a framework for a secure decentralized database. The presented framework uses smart contracts, a way to decentrally execute programs based on an Ethereum blockchain. Individual malicious robots aim to disrupt the collective decision-making process of a simulated swarm of e-puck robots by spreading misinformation. The robot swarm successfully disregards the wrong information. The authors indicate that blockchain networks can be used for robot swarms, and the low processing and memory capacity of swarm robots does not prohibit the use of blockchains in real-world scenarios.

When developing the swarm robot controller and hardware, it is difficult to anticipate all future situations that this robot swarm may experience. Hunt claims that nature provides an example solution that we can follow: phenotypic plasticity. The idea is to train robot swarms in (simulated) heterogeneous environments, for example, using methods of evolutionary computation. The general swarm robot design should allow for flexibility such that they can be adapted and shaped ideally in three dimensions: behavioral, physiological, and morphological plasticity. Behavioral plasticity of the swarm members introduces diversity that can be exploited, for example, to increase fault tolerance and decision accuracy. Physiological plasticity in robots could be modes of operation that have different energy consumption. Morphological plasticity could be known implementations of self-assembling swarm robots. In summary, Hunt opens a door to more flexible and dynamic ways of drafting, developing, and optimizing robot swarms for the real world mainly based on a systematic behavioral and morphological diversity.

Rausch et al. propose an empirical case study of the impact of network topology over the spread of information in a robot swarm. Specifically, they consider the possible benefits of scale-free communication topology. They experimentally show that there is actually a trade-off in using scale-free (rather than random) topology: information spreads faster, enabling quicker reactions to changes in dynamic environments, but at the cost of a decreased stability as the emergence of consensus is hindered by communication pathways of different lengths.

To ensure a smooth transition from lab to market, it is necessary to recognize user needs and to evaluate the acceptability of robot swarms. The paper by Carrilo-Zapata et al. conducts a study against three application domains wherein robot swarms are considered as game changing tools. The mutual shaping methodology proposed entails a bi-directional knowledge exchange between swarm designers and final users, raising awareness of the possibility offered by the technology but also allowing to collect important design and interaction features that can drive the deployment. Overall, the study reveals that robot swarms can play an important role within the considered application domains, above all when they work in support of human operations, rather than as entire replacements.

Another important hurdle to deploying swarms in the physical realm is robustness. Contrary to the adage “there is safety in

numbers,” robustness is not an inherent benefit of robot swarms that results from redundancy. Robustness is a challenging design goal, made complex by the interplay between the benefits of redundancy and the need for scalability. Wilson et al. argue that achieving robustness through redundancy involves a careful co-design of hardware, fabrication processes, and control software. To investigate this idea, the authors present an approach to achieving robustness that involves a novel hardware-software co-design of a modular robotic platform called “DONUTs” (Deformable Self-Organizing Nomadic Units). The modules are inexpensive, flexible printed circuit boards, and designed to move as a collective through magnetic interaction. Wilson et al. study several control strategies that explore the design space of inter-module connectivity to shed light on the interplay between robustness, scalability, and controllability.

Nave et al. investigate on a biological model related to social insects—the tower building behavior of red imported fire ants. Results show that individuals moving under the influence of local attraction can form large towers. The system shows a sudden density-dependent phase transition as the attraction parameter is varied. The resulting towers of simulated agents are constantly rebuilt and move over time—a feature that has to be considered for robotic applications. There is for future robotic studies, where robots build towers out of themselves in a manner similar as the fire ants. In a real-world application, a tower of robots could be useful for seeing over obstacles, providing scaffolding for climbing, or marking a location of interest. Robotic tower-builders would need capabilities for sensing neighbors, climbing onto and off one another, and supporting appropriate loads. Building such robotic tower-builders would be an interesting step for future robotics research.

While engineers take robots to the real world to automate tasks currently done by humans or impossible for humans, biologists take robots to the real world to study animal behavior. Yang et al. study a robotics-based experimental test paradigm where a robotic replica is used to influence the behavior of Zebrafish. Two setups were studied. In the individual training condition, a single fish learned to open the correct of two doors by itself. In the social training condition, a fish observes the replica approaching both doors with the correct one opening after a certain period of time. Main contributions are the technical innovation of this robot-supported experiment and the negative

result indicating that there is no improvement by social learning. Yang et al. claim that their setup can generalize to other species, such as guppies and mollies but also insects, mammals, and even invertebrates. It seems promising that with ongoing technological progress we will see more of these bio-hybrid systems with robots and animals interacting closely in the real world.

In summary, all the above papers that study an engineering approach to take self-organizing robots to the field, struggle with a technological bottleneck: local sensing, coordinated actuation, and means of communication that work reliably in field environments. This is a common challenge of robotics and will require designing smart control algorithms with minimal requirements for sensing, actuation, and communication. Common to all papers in this Research Topic are deviations between model abstractions and the physical realm. We still do not know well-enough what deviations are caused by which abstraction in swarm and multi-robot models and simulations. The intrinsic stochastic nature of self-organizing systems adds to this challenge. In future work, this will require an effort toward more robust hardware, as well as verifiable swarm and robot behaviors to achieve certification. Our Research Topic covers a wide range of fields, concepts, and methods that will hopefully help to kick our robots out of the lab, pushing toward a novel “field swarm robotics,” to establish cyber-physical systems in the wild and to design distributed systems for radically novel applications using self-organization in the physical realm.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Hamann, Schranz, Elmenreich, Trianni, Pincioli, Bredeche and Ferrante. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Automatic Off-Line Design of Robot Swarms: A Manifesto

Mauro Birattari^{1*}, Antoine Ligot¹, Darko Bozhinoski¹, Manuele Brambilla¹,
Gianpiero Francesca¹, Lorenzo Garattoni¹, David Garzón Ramos¹, Ken Hasselmann¹,
Miquel Kegeleirs¹, Jonas Kuckling¹, Federico Pagnozzi¹, Andrea Roli²,
Muhammad Salman¹ and Thomas Stützle¹

¹ Université libre de Bruxelles, Brussels, Belgium, ² Alma Mater Studiorum, Università di Bologna, Bologna, Italy

OPEN ACCESS

Edited by:

Vito Trianni,
Istituto di Scienze e Tecnologie della
Cognizione (ISTC), Italy

Reviewed by:

Anders Lyhne Christensen,
University Institute of Lisbon (ISCTE),
Portugal

Nicolas Bredeche,
Université Pierre et Marie Curie,

France
Sabine Hauert,
University of Bristol, United Kingdom

*Correspondence:

Mauro Birattari
mbiro@ulb.ac.be

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 16 April 2019

Accepted: 03 July 2019

Published: 19 July 2019

Citation:

Birattari M, Ligot A, Bozhinoski D,
Brambilla M, Francesca G,
Garattoni L, Garzón Ramos D,
Hasselmann K, Kegeleirs M,
Kuckling J, Pagnozzi F, Roli A,
Salman M and Stützle T (2019)
Automatic Off-Line Design of Robot
Swarms: A Manifesto.
Front. Robot. AI 6:59.
doi: 10.3389/frobt.2019.00059

Designing collective behaviors for robot swarms is a difficult endeavor due to their fully distributed, highly redundant, and ever-changing nature. To overcome the challenge, a few approaches have been proposed, which can be classified as manual, semi-automatic, or automatic design. This paper is intended to be the manifesto of the automatic off-line design for robot swarms. We define the off-line design problem and illustrate it via a possible practical realization, highlight the core research questions, raise a number of issues regarding the existing literature that is relevant to the automatic off-line design, and provide guidelines that we deem necessary for a healthy development of the domain and for ensuring its relevance to potential real-world applications.

Keywords: swarm robotics, automatic design, collective behaviors, design methodology, evolutionary robotics

Although swarm robotics is widely recognized as a promising approach to coordinating large groups of robots (Dorigo et al., 2014; Yang et al., 2018) and has already gained a prominent position in the scientific literature (e.g., see Rubenstein et al., 2014; Werfel et al., 2014; Garattoni and Birattari, 2018; Slavkov et al., 2018; Yu et al., 2018; Li et al., 2019; Xie et al., 2019), a general methodology for designing collective behaviors for robot swarms is still missing (Brambilla et al., 2013). The design problem is particularly challenging because it aims at producing a system that is autonomous, fully distributed, and highly redundant: robots do not have any predefined role and do not rely on any external infrastructure (Beni, 2004; Şahin, 2004). A robot swarm is a loosely coupled system in which the collective behavior of the system results from the local interactions between individuals, and between them and the environment. These interactions cannot be explicitly defined at design time due to the high uncertainty that characterizes the operation of a swarm. As a result, at least in the general case, it is impossible to tell what the individual robots should do so that a desired collective behavior is achieved. This rules out the application of traditional multi-robot systems and software engineering techniques, which rely on formally deriving the individual behaviors of the robots from specifications expressed at the collective level (Brugali, 2007; Di Ruscio et al., 2014; Bozhinoski et al., 2015; Schlegel et al., 2015).

A few methods/tools have been proposed that, under a number of restrictive hypotheses and constraints, support the designer for specific classes of missions (Hamann and Wörn, 2008; Kazadi, 2009; Berman et al., 2011; Beal et al., 2012; Brambilla et al., 2015; Reina et al., 2015; Lopes et al., 2016; Pinciroli and Beltrame, 2016). Also, a few automatic (and semi-automatic) design methods have been proposed that operate under various assumptions (Nolfi and Floreano, 2000; Watson et al., 2002; Duarte et al., 2014; Francesca et al., 2014). For recent discussions, see Francesca and Birattari (2016) and Bredeche et al. (2018).

This paper is intended to be the manifesto of the automatic off-line design of robot swarms. In this approach, the design problem is cast into an optimization problem that is solved

off-line—that is, before the swarm is deployed in the target environment. An optimization algorithm searches a space of possible designs with the goal of maximizing an appropriate mission-specific performance measure. Within the design process, the performance of candidate designs explored by the optimization algorithm is assessed via computer-based simulations. Once the optimization algorithm terminates, the selected design is uploaded to the robots and the swarm is deployed in its target environment. In the following, we focus mostly on the development of software but the discussion can be directly extended to the hardware. For example, the automatic off-line design process might optimize the number of robots in the swarm; if the swarm is heterogeneous, select the fraction of robots of type A, B, C, ...; fine-tune parameters of hardware or firmware; activate/deactivate or add/remove hardware modules; design chassis, shell, or attachments.

Our vision is that, in a relatively close future, automatic off-line design will be a practically relevant way of realizing robot swarms. Likely, it will not be the only one: other approaches will be available, each with its specific advantages and ideal areas of application, as well as its disadvantages and limitations. Among them, we foresee that a relevant role will be played by manual design, semi-automatic design, automatic on-line design, and hybrid approaches that combine the previous ones. Nonetheless, we expect that the automatic off-line approach will play a major role, both on its own and also as a component of hybrid approaches.

In the automatic off-line approach, robot swarms are generated to perform missions that are sampled from a given class of interest and are sufficiently different from one another to possibly require (or benefit from) a tailored design. An automatic off-line method must operate on the missions of the given class without requiring either mission-specific adjustments, or per-mission human intervention. The notion of a class of missions plays here a central role. It refers to a set of missions, together with a probability measure defined on them, which determines their relative frequency of appearance. Typically, an explicit, closed-form definition of the set of missions and of the probability measure is not available—and is not even needed. Instead, what we have is a stream of missions sampled from the class of interest according to the aforementioned probability measure. The assumption that missions are sampled according to a probability measure gives a formal meaning to the notion of expected performance, as well as to any other statistics one might wish to adopt to describe the aggregate behavior of a design method across the missions of interest. To illustrate the automatic off-line design of robot swarms, it is convenient to sketch one of its possible practical applications.

Fiorella's swarm gardening

(for an artist's rendition, see **Figure 1**)

Fiorella owns a robot-swarm gardening business and offers her individually-tailored service to her many customers in the Brussels area. She has a busy schedule: every day, she visits three or four customers with her gardening

swarm. Customers book Fiorella's service via a form on her website. Through the form, they ask for one or more specific interventions—e.g., cutting grass, watering flowers. They also provide relevant information on their garden—e.g., size, shape, orientation. The interventions requested and the characteristics of the garden specify the mission that Fiorella's swarm must perform for a specific customer. As the list of possible interventions and characteristics of the garden is huge, the class of possible missions is overwhelmingly large and rather diverse. To provide her customers with the best gardening experience—but also to cut costs and maximize her benefit—Fiorella relies on an automatic off-line method that designs and fine-tunes the behaviors of her swarm specifically for each mission. The design process takes place while Fiorella drives her swarm to the customer's garden: her powerful computers run simulations using the information provided by the customer on the interventions and on the garden. The design process must be performed within a limited amount of time—the time of the ride to the customer's. As Fiorella arrives on the spot, the selected design is uploaded to the robots and the swarm is deployed in the garden. Fiorella cannot intervene in the design process—she drives the van in the meantime. Moreover, due to her tight schedule, Fiorella cannot either test the selected design on the robots before deployment and possibly re-run the design process: once she reaches the customer, robots must be operational. Any per-mission human intervention and any test on the robots in the target environment would be too time consuming and expensive: they would increase costs dramatically and Fiorella would be unavoidably out of business.

Missions in the class of interest can be relatively minor variations of each other—e.g., cut the grass in a small garden; in a large one; in one with a central flower bed. In this case, the behaviors to be produced will be similar, with some minor differences to increase performance or reduce execution time. Missions can also be substantially different in the nature of their goals and require major differences in the behaviors to be realized—e.g., cut the grass; gather dead leaves in a specific place; locate and map mole tunnels.

In Fiorella's example, the central role of the notion of class of missions emerges clearly. Fiorella faces a stream of missions sampled from the possible missions for which customers might demand her intervention—and which her swarm can hopefully accomplish. It is in the repetitive nature of the design problems faced by Fiorella that the significance of automatic design lies. Indeed, if Fiorella had to solve a single design problem (instead of a stream thereof) she could more profitably solve it either manually or via a semi-automatic design method¹. It is only when one has to solve a stream of design problems that the human intervention might become uneconomical or even unfeasible.

¹ By semi-automatic design, we mean an approach in which a human designer is assisted by an optimization process similar to the one of automatic design, but can afford intervening in the process (on a per-mission basis) to guide it according to their insight.

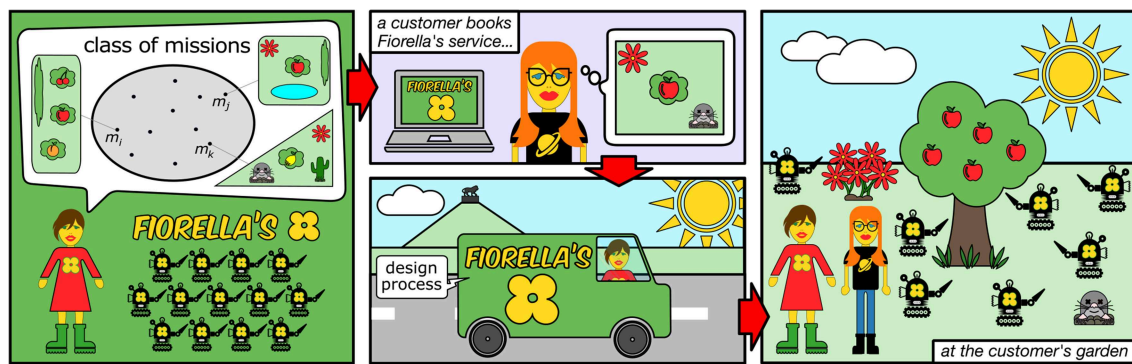


FIGURE 1 | Fiorella's swarm gardening. Fiorella's robots can perform a large class of gardening missions. Through her website, customers book Fiorella's services, specify the interventions to be performed, and provide a description of their garden. On the basis of this information, while Fiorella drives her robots to customers, her algorithms automatically design and fine-tune the behavior of the robots so as to offer a tailored service. When she arrives at a customer location, the gardening swarm is operational and immediately deployed.

Conceiving, implementing, and setting up an automatic design method is in itself an investment of time and resources, which pays off only if the design process is then repeated a sufficient number of times on multiple missions—those of the class for which the automatic design method is conceived. If one had to address a single mission, it would be more sensible to invest time and resources on that specific mission—by adopting an *ad-hoc* manual or semi-automatic approach—rather than on the development of an automatic design method that would be then used only once. For a schematic representation of the automatic off-line design process, see **Figure 2**.

Fiorella's example allows us to highlight a number of issues and research questions that are relevant to the automatic off-line design of robot swarms.

Can effective robot swarms be designed automatically via an off-line process? Can we conceive a method to automatically design a swarm for any mission within a given class? What is the class of missions for which a given method can design an appropriate swarm? How can a given method be generalized to solve a larger class of missions?

Given a class of missions, which is the most appropriate design method? What elements or characteristics of a design method influence its ability to handle missions of a given class? Vice versa, what elements/characteristics of a class of missions might suggest that a given method is appropriate to handle them? Which features of a specific mission make it challenging for a given design method? Are these challenging features equally challenging for all possible design methods? Is it possible to match challenging features of missions with characteristics of design methods?

To what extent a given design method is robust to the so called reality gap—that is, the difference between simulation models and reality? Is it possible to predict the performance drop that a swarm designed off-line will experience when deployed in the target environment? Are different design methods equally sensitive to the reality gap? What elements/characteristics of a design method

make it more or less robust to the reality gap? Can these characteristics be leveraged to engineer a design method that is inherently robust to the reality gap? How should models be devised to be effectively used within an off-line design process?

How efficient is a design method? In other terms, how many off-line simulation runs are required to produce an effective design? What elements/characteristics of a design method increase/decrease its efficiency? How well does a given design method behave for a large/small design budget—that is, when allowed to perform few/many off-line simulation runs? Does the efficiency of a design method depend on the specific mission or class of missions considered? What elements/characteristics of a mission determine the minimum size of the design budget needed to produce an effective design? When should a design process be stopped?

This list of questions encompasses many relevant issues but it is by no means exhaustive. For example, other relevant issues would concern the off-line definition of the swarm size (or its spatial density), its impact on performance, and the robustness/scalability of behaviors that are automatically designed.

A body of literature exists that is relevant to the automatic off-line design of robot swarms. The largest share of the design methods described in the relevant literature belong in the neuro-evolutionary domain (Nolfi and Floreano, 2000): robots are controlled by a neural network whose synaptic weights (and possibly the topology) are optimized by an evolutionary algorithm (Quinn et al., 2003; Christensen and Dorigo, 2006; Baldassarre et al., 2007; Trianni, 2008; Hauert et al., 2009; Trianni and Nolfi, 2009; Waibel et al., 2009; Ferrante et al., 2013, 2015; Gomes et al., 2013; Trianni and López-Ibáñez, 2015). For a review of the neuro-evolutionary approach (including also single-robot studies) see Floreano and Keller (2010), Bongard (2013), Bongard and Lipson (2014), Trianni (2014), Doncieux et al. (2015), and Silva et al. (2016). Other approaches depart from neuro-evolution as (i) robots are controlled by software architectures other than

Repeat *ad libitum*:

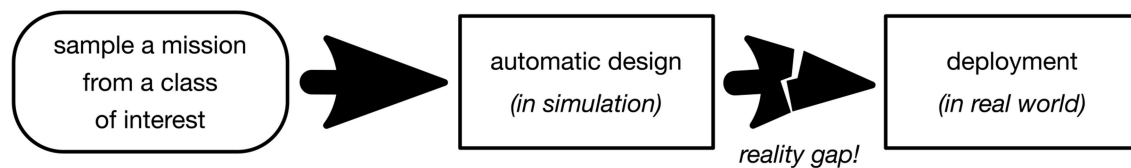


FIGURE 2 | Flowchart diagram of the automatic off-line design process. A mission is sampled from a class of interest. Using computer-based simulations, an automatic design method defines a robot swarm tailored to the mission sampled. Once the automatic design terminates, the swarm is deployed in the target environment and has to cross the so-called reality gap—the possibly subtle but inevitable difference between simulation and reality (Brooks, 1992; Jakobi et al., 1995)—which is among the most challenging issues in automatic off-line design. The process is repeated *ad libitum*. It should be noted that an automatic design method could generate a robot swarm from scratch for every mission sampled or could refine and adapt a solution previously generated for a similar one. It could also, for example, produce a robot swarm by combining and modifying solutions (or partial solutions) contained in a catalog of template solutions that were pre-defined by a human expert. The only condition that needs to be respected for the process to be qualified as automatic off-line design is that such initial (partial) solutions must be selected without any per-mission human intervention and without recourse to tests performed in the target environment.

neural networks (Hecker et al., 2012; Gauci et al., 2014a; Jones et al., 2016), or (ii) they adopt optimization algorithms other than an evolution algorithm (Pugh et al., 2005), or (iii) both (Francesca et al., 2014, 2015; Gauci et al., 2014b; Kuckling et al., 2018). Besides, a few studies exist that provide insight into the reality gap in the automatic design of robot swarms and/or define methods to handle it (Francesca et al., 2014; Birattari et al., 2016; Ligot and Birattari, 2018). Additionally, a number of methods have been proposed that, although described in single-robot applications, are relevant to the design of robot swarms (Jakobi et al., 1995; Miglino et al., 1995; Floreano and Mondada, 1996; Jakobi, 1997; Bongard and Lipson, 2004; Zagal et al., 2004; Boeing and Braunl, 2012; Koos et al., 2013).

It is our contention that, with only few exceptions, the aforementioned methods have been studied following protocols that were not conceived to directly address the core research questions sketched above. Although these protocols allowed studies which partially addressed those questions, they were conceived to target other questions that are mostly relevant to other domains including, for example, evolutionary biology and the semi-automatic design of robot swarms¹. In almost the totality of the studies, the focus is on a specific mission that must be performed by a swarm—or, equivalently, on a specific capability that the swarm should acquire and display. The design method is proposed only as a way to achieve the desired collective behavior and is not the protagonist of the study: the study is not structured to highlight its properties and assess its performance. The design method has so little importance that it is not customarily given an identifying name—contrary to what happens in related fields such as machine learning or heuristic optimization. Typically, the design method is tested on a single mission and it is not compared to any alternative. It is rare that a same design method is tested across multiple studies on multiple missions without undergoing any (manually-applied) mission-specific modification. In many studies, the control software produced by a design method is tested only in simulation and no assessment is provided on whether and to what extent it crosses the reality gap satisfactorily. Moreover, design methods survive only the time span of the

paper in which they are introduced and their implementation is not routinely made publicly available for further studies, to be possibly performed by a third party. Often, a design method is run iteratively on a single mission. It is run once, the behavior generated is inspected by the designer who then modifies the method itself or the objective function to be optimized—e.g., by adding/removing terms. These activities are then iterated at will until a satisfactory behavior is obtained. In most cases, this iterative process is not detailed in research articles: it is often unclear how many iterations have been performed, what has been measured at each iteration, what modifications have been implemented, what ideas have been tried and then abandoned. In these cases, the research articles present only the final setting that eventually generated the behavior discussed. The iterative process is repeated only once, as it would be difficult to produce independent trials of a process that features a human in the loop. As a result, the robustness and the repeatability of the process are not assessed.

An appropriate protocol to address the aforementioned issues should reflect the following tenets of the research in automatic off-line design: (i) automatic off-line design methods should not be mission-specific and should be able to address a whole class of missions without undergoing any modification; (ii) once a mission is specified, human intervention is not provided for in any phase of the design process. Indeed, research that is intended to be relevant to the automatic off-line design of robot swarms should exclude the case in which design methods are conceived for or are manually adapted to a specific mission—for example, by manually tuning parameters of the optimization algorithm and/or of the control architecture, or by pre-filtering sensor readings on the basis of insight that only a human designer can provide. It should also exclude the case in which, on a per-mission basis, human designers are allowed to inspect (via either simulation or robot experiments) the behavior of an automatically designed swarm and, on the basis of their observations, modify elements of the automatic design process and iterate it at will, until they obtain satisfactory results. In particular, human designers should not be allowed, on a

per-mission basis, to use any insight gained through inspection to modify the design method (optimization algorithm, architecture, sensor pre-filtering, etc.); to adapt simulation models; and to amend the objective function by adding/removing terms so as to steer the design process as wished. On the other hand, to effectively contribute to the development of the domain, researchers in the automatic off-line design of robot swarms should pay particular attention to a number of methodological issues. In particular, they should: (a) provide a clear and thorough description of the design methods they propose, including a list of the value of all parameters; (b) precisely characterize the platforms for which these methods can generate control software; (c) clearly identify and name methods for future reference; (d) publish implementations; (e) test methods on multiple missions; (f) identify—at least informally—the class of missions that a method is intended to address; (g) perform comparative studies in which methods under analysis are tested under the same conditions; and (h) run robot experiments to assess robustness to the reality gap. It is our contention that this minimal set of guidelines will allow the domain to grow healthy and thriving so as to eventually prove its practical relevance in real-world applications.

REFERENCES

- Baldassarre, G., Trianni, V., Bonani, M., Mondada, F., Dorigo, M., and Nolfi, S. (2007). Self-organized coordinated motion in groups of physically connected robots. *IEEE Trans. Syst. Man Cybern. B* 37, 224–239. doi: 10.1109/TSMCB.2006.881299
- Beal, J., Dulman, S., Usbeck, K., Viroli, M., and Correll, N. (2012). “Organizing the aggregate: languages for spatial computing,” in *Formal and Practical Aspects of Domain-Specific Languages: Recent Developments*, Chap. 16, ed M. Mernik (Hershey, PA: IGI Global), 436–501.
- Beni, G. (2004). “From swarm intelligence to swarm robotics,” in *Swarm Robotics*, SAB, volume 3342 of LNCS (Berlin: Springer), 1–9.
- Berman, S., Kumar, V., and Nagpal, R. (2011). “Design of control policies for spatially inhomogeneous robot swarms with application to commercial pollination,” in *IEEE International Conference on Robotics and Automation – ICRA* (Piscataway, NJ: IEEE), 378–385.
- Birattari, M., Delhaisse, B., Francesca, G., and Kerdoncuff, Y. (2016). “Observing the effects of overdesign in the automatic design of control software for robot swarms,” in *Swarm Intelligence – ANTS*, volume 9882 of LNCS (Cham: Springer), 45–57.
- Boeing, A., and Braunl, T. (2012). “Leveraging multiple simulators for crossing the reality gap,” in *Proceedings of the International Conference on Control, Automation, Robotics and Vision – ICARCV* (Piscataway NJ: IEEE), 1113–1119.
- Bongard, J., and Lipson, H. (2004). “Once more unto the breach: co-evolving a robot and its simulator,” in *Artificial Life IX: Proceedings of the Conference on the Simulation and Synthesis of Living Systems* (Cambridge, MA: MIT Press), 57–62.
- Bongard, J., and Lipson, H. (2014). Evolved machines shed light on robustness and resilience. *Proc. IEEE* 102, 899–914. doi: 10.1109/JPROC.2014.2312844
- Bongard, J. C. (2013). Evolutionary robotics. *Commun. ACM* 56, 74–83. doi: 10.1145/2492007.2493883
- Bozhinoski, D., Di Ruscio, D., Malavolta, I., Pelliccione, P., and Tivoli, M. (2015). “Flyaq: enabling non-expert users to specify and generate missions of autonomous multicopters,” in *IEEE/ACM International Conference on Automated Software Engineering – ASE* (Piscataway, NJ: IEEE), 801–806.
- Brambilla, M., Brutschy, A., Dorigo, M., and Birattari, M. (2015). Property-driven design for swarm robotics: a design method based on prescriptive modeling and model checking. *ACM Trans. Auton. Adapt. Syst.* 9, 17.1–28. doi: 10.1145/2700318
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* 7, 1–41. doi: 10.1007/s11721-012-0075-2
- Bredeche, N., Haasdijk, E., and Prieto, A. (2018). Embodied evolution in collective robotics: a review. *Front. Robot. AI* 5:12. doi: 10.3389/frobt.2018.00012
- Brooks, R. (1992). “Artificial life and real robots,” in *Towards a Practice of Autonomous Systems. Proceedings of the First European Conference on Artificial Life* (Cambridge, MA: MIT Press), 3–10.
- Brugali, D. (2007). *Software Engineering for Experimental Robotics*, Vol. 30. Berlin; Heidelberg: Springer.
- Christensen, A. L., and Dorigo, M. (2006). “Evolving an integrated phototaxis and hole-avoidance behavior for a swarm-bot,” in *Artificial Life – ALIFE* (Cambridge, MA: MIT Press), 248–254.
- Di Ruscio, D., Malavolta, I., and Pelliccione, P. (2014). “A family of domain-specific languages for specifying civilian missions of multi-robot systems,” in *Proceedings of the 1st International Workshop on Model-Driven Robot Software Engineering – MORSE* (York: MORSE workshop), 13–26. Available online at: <http://ceur-ws.org/Vol-1319/>
- Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. E. G. (2015). Evolutionary robotics: what, why, and where to. *Front. Robot. AI* 2:4. doi: 10.3389/frobt.2015.00004
- Dorigo, M., Birattari, M., and Brambilla, M. (2014). Swarm robotics. *Scholarpedia* 9:1463. doi: 10.4249/scholarpedia.1463
- Duarte, M., Oliveira, S., and Christensen, A. (2014). “Evolution of hierarchical controllers for multirobot systems,” in *Artificial Life – ALIFE*, eds H. Sayama, J. Rieffel, S. Risi, R. Doursat, and H. Lipson (Cambridge, MA: MIT Press), 657–664.
- Ferrante, E., Duéñez Guzmán, E., Turgut, A. E., and Wenseleers, T. (2013). “Geswarm: grammatical evolution for the automatic synthesis of collective behaviors in swarm robotics,” in *Genetic and Evolutionary Computation – GECCO* (New York, NY: ACM), 17–24.
- Ferrante, E., Turgut, A., Duéñez-Guzmán, E., Dorigo, M., and Wenseleers, T. (2015). Evolution of self-organized task specialization in robot swarms. *PLoS Comput. Biol.* 11:e1004273. doi: 10.1371/journal.pcbi.1004273

AUTHOR CONTRIBUTIONS

All authors contributed to the elaboration of the ideas presented in the paper, read the text, and provided comments. In particular, MBI started the discussion and lead it, he also drafted the paper and coordinated its revision. AL selected/reviewed the relevant literature and contributed to the formulation of the research questions. DB contributed to framing the complexity of the design problem in swarm robotics. The other authors equally contributed to this paper and are listed in alphabetical order.

FUNDING

The project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 681872). MBI, JK, and TS acknowledge support from the Belgian *Fonds de la Recherche Scientifique – FNRS*. DR acknowledges support from the Colombian *Administrative Department of Science, Technology and Innovation – COLCIENCIAS*. AR acknowledges the support of the *Chaire internationale* programme of Université libre de Bruxelles.

- Floreano, D., and Keller, L. (2010). Evolution of adaptive behaviour in robots by means of darwinian selection. *PLoS Biol.* 8:e1000292. doi: 10.1371/journal.pbio.1000292
- Floreano, D., and Mondada, F. (1996). "Evolution of plastic neurocontrollers for situated agents," in *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB)* (Cambridge, MA: MIT Press), 402–410.
- Francesca, G., and Birattari, M. (2016). Automatic design of robot swarms: achievements and challenges. *Front. Robot. AI* 3:29. doi: 10.3389/frobt.2016.00029
- Francesca, G., Brambilla, M., Brutschy, A., Garattoni, L., Miletitch, R., Podevijn, G., et al. (2015). AutoMoDe-Chocolate: automatic design of control software for robot swarms. *Swarm Intell.* 9, 125–152. doi: 10.1007/s11721-015-0107-9
- Francesca, G., Brambilla, M., Brutschy, A., Trianni, V., and Birattari, M. (2014). AutoMoDe: a novel approach to the automatic design of control software for robot swarms. *Swarm Intell.* 8, 89–112. doi: 10.1007/s11721-014-0092-4
- Garattoni, L., and Birattari, M. (2018). Autonomous task sequencing in a robot swarm. *Sci. Robot.* 3:eaa0430. doi: 10.1126/scirobotics.aat0430
- Gauci, M., Chen, J., Li, W., Dodd, T. J., and Groß, R. (2014a). "Clustering objects with robots that do not compute," in *Autonomous Agents and Multiagent Systems – AAMAS* (Richland, SC: IFAAMAS), 421–428.
- Gauci, M., Chen, J., Li, W., J Dodd, T., and Groß, R. (2014b). Self-organized aggregation without computation. *Int. J. Robot. Res.* 33, 1145–1161. doi: 10.1177/0278364914525244
- Gomes, J., Urbano, P., and Christensen, A. (2013). Evolution of swarm robotics systems with novelty search. *Swarm Intell.* 7, 115–144. doi: 10.1007/s11721-013-0081-z
- Hamann, H., and Wörn, H. (2008). A framework of space-time continuous models for algorithm design in swarm robotics. *Swarm Intell.* 2, 209–239. doi: 10.1007/s11721-008-0015-3
- Hauert, S., Zufferey, J.-C., and Floreano, D. (2009). Evolved swarming without positioning information: an application in aerial communication relay. *Auton. Robots* 26, 21–32. doi: 10.1007/s10514-008-9104-9
- Hecker, J. P., Letendre, K., Stolleis, K., Washington, D., and Moses, M. E. (2012). "Formica ex machina: ant swarm foraging from physical to virtual and back again," in *Swarm Intelligence – ANTS, volume 7461 of LNCS* (Berlin: Springer), 252–259.
- Jakobi, N. (1997). Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adapt. Behav.* 6, 325–368. doi: 10.1177/105971239700600205
- Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: the use of simulation in evolutionary robotics. *Lect. Notes Comput. Sci.* 929, 704–720. doi: 10.1007/3-540-59496-5_337
- Jones, S., Studley, M., Hauert, S., and Winfield, A. (2016). "Evolving behaviour trees for swarm robotics," in *Distributed Autonomous Robotic Systems* (Cham: Springer), 487–501.
- Kazadi, S. (2009). Model independence in swarm robotics. *Int. J. Intell. Comput. Cybern.* 2, 672–694. doi: 10.1108/17563780911005836
- Koos, S., Mouret, J.-B., and Doncieux, S. (2013). The transferability approach: crossing the reality gap in evolutionary robotics. *IEEE Trans. Evol. Comput.* 17, 122–145. doi: 10.1109/TEVC.2012.2185849
- Kuckling, J., Ligtot, A., Bozhinoski, D., and Birattari, M. (2018). "Behavior trees as a control architecture in the automatic modular design of robot swarms," in *Swarm Intelligence – ANTS, volume 11172 of LNCS* (Cham: Springer), 30–43.
- Li, S., Batra, R., Brown, D., Chang, H.-D., Ranganathan, N., Hoberman, C., et al. (2019). Particle robotics based on statistical mechanics of loosely coupled components. *Nature* 567, 361–365. doi: 10.1038/s41586-019-1022-9
- Ligtot, A., and Birattari, M. (2018). "On mimicking the effects of the reality gap with simulation-only experiments," in *Swarm Intelligence – ANTS, volume 11172 of LNCS* (Cham: Springer), 109–122.
- Lopes, Y. K., Trenkwalder, S. M., Leal, A. B., Dodd, T. J., and Groß, R. (2016). Supervisory control theory applied to swarm robotics. *Swarm Intell.* 10, 65–97. doi: 10.1007/s11721-016-0119-0
- Miglino, O., Lund, H., and Nolfi, S. (1995). Evolving mobile robots in simulated and real environments. *Artif. Life* 2, 417–434. doi: 10.1162/artl.1995.2.4.417
- Nolfi, S., and Floreano, D. (2000). *Evolutionary Robotics*. Cambridge, MA: MIT Press.
- Pincirol, C., and Beltrame, G. (2016). Buzz: a programming language for robot swarms. *IEEE Softw.* 33, 97–100. doi: 10.1109/MS.2016.95
- Pugh, J., Martinoli, A., and Zhang, Y. (2005). "Particle swarm optimization for unsupervised robotic learning," in *Swarm Intelligence Symposium – SI* (Piscataway, NJ: IEEE), 92–99.
- Quinn, M., Smith, L., Mayley, G., and Husbands, P. (2003). Evolving controllers for a homogeneous system of physical robots: structured cooperation with minimal sensors. *Philos. Trans. R. Soc. A* 361, 2321–2343. doi: 10.1098/rsta.2003.1258
- Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M., and Trianni, V. (2015). A design pattern for decentralised decision making. *PLoS ONE* 10:e0140950. doi: 10.1371/journal.pone.0140950
- Rubenstein, M., Cornejo, A., and Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science* 345, 795–799. doi: 10.1126/science.1254295
- Şahin, E. (2004). "Swarm robotics: from sources of inspiration to domains of application," in *Swarm Robotics, SAB, volume 3342 of LNCS* (Berlin: Springer), 10–20.
- Schlegel, C., Lotz, A., Lutz, M., Stampfer, D., Inglés-Romero, J. F., and Vicente-Chicote, C. (2015). Model-driven software systems engineering in robotics: covering the complete life-cycle of a robot. *Informat. Technol.* 57, 85–98. doi: 10.1515/itit-2014-1069
- Silva, F., Duarte, M., Correia, L., Oliveira, S. M., and Christensen, A. L. (2016). Open issues in evolutionary robotics. *Evol. Comput.* 24, 205–236. doi: 10.1162/EVCO_a_00172
- Slavkov, I., Carrillo-Zapata, D., Carranza, N., Diego, X., Jansson, F., Kaandorp, J., et al. (2018). Morphogenesis in robot swarms. *Sci. Robot.* 3:eaau9178. doi: 10.1126/scirobotics.aau9178
- Trianni, V. (2008). *Evolutionary Swarm Robotics*. Berlin: Springer.
- Trianni, V. (2014). Evolutionary robotics: model or design? *Front. Robot. AI* 1:13. doi: 10.3389/frobt.2014.00013
- Trianni, V., and López-Ibáñez, M. (2015). Advantages of task-specific multi-objective optimisation in evolutionary robotics. *PLoS ONE* 10:e0136406. doi: 10.1371/journal.pone.0136406
- Trianni, V., and Nolfi, S. (2009). Self-organizing sync in a robotic swarm: a dynamical system view. *IEEE Trans. Evol. Comput.* 13, 722–741. doi: 10.1109/TEVC.2009.2015577
- Waibel, M., Keller, L., and Floreano, D. (2009). Genetic team composition and level of selection in the evolution of multi-agent systems. *IEEE Trans. Evol. Comput.* 13, 648–660. doi: 10.1109/TEVC.2008.2011741
- Watson, R., Ficici, G. S., and Pollack, J. (2002). Embodied evolution: distributing an evolutionary algorithm in a population of robots. *Robot. Auton. Syst.* 39, 1–18. doi: 10.1016/S0921-8890(02)00170-7
- Werfel, J., Petersen, K., and Nagpal, R. (2014). Designing collective behavior in a termite-inspired robot construction team. *Science* 343, 754–758. doi: 10.1126/science.1245842
- Xie, H., Sun, M., Fan, X., Lin, Z., Chen, W., Wang, L., et al. (2019). Reconfigurable magnetic microrobot swarm: multimode transformation, locomotion, and manipulation. *Sci. Robot.* 4:eaav8006. doi: 10.1126/scirobotics.aav8006
- Yang, G.-Z., Bellingham, J., Dupont, E. P., Fischer, P., Floridi, L., Full, R., et al. (2018). The grand challenges of Science Robotics. *Sci. Robot.* 3:eaar7650. doi: 10.1126/scirobotics.aar7650
- Yu, J., Wang, B., Du, X., Wang, Q., and Zhang, L. (2018). Ultra-extensible ribbon-like magnetic microswarm. *Nat. Commun.* 9:3260. doi: 10.1038/s41467-018-05749-6
- Zagal, J. C., Ruiz-Del-Solar, J., and Vallejos, P. (2004). Back to reality: crossing the reality gap in evolutionary robotics. *IFAC Proc. Vol.* 37, 834–839. doi: 10.1016/S1474-6670(17)32084-0

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2019 Birattari, Ligtot, Bozhinoski, Brambilla, Francesca, Garattoni, Garzón Ramos, Hasselmann, Kegeleirs, Kuckling, Pagnozzi, Roli, Salman and Stützle. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



A Comparison of Individual Learning and Social Learning in Zebrafish Through an Ethorobotics Approach

Yanpeng Yang^{1,2}, Romain J. G. Clément², Stefano Ghirlanda^{3,4,5} and Maurizio Porfiri^{2,6*}

¹ Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, School of Mechanical Engineering, Tianjin University, Tianjin, China, ² Department of Mechanical and Aerospace Engineering, New York University, Tandon School of Engineering, Brooklyn, NY, United States, ³ Department of Psychology, Brooklyn College, Brooklyn, NY, United States, ⁴ Departments of Psychology and Biology, The Graduate Center of the City University of New York (CUNY), New York, NY, United States, ⁵ Centre for the Study of Cultural Evolution, Stockholm University, Stockholm, Sweden, ⁶ Department of Biomedical Engineering, New York University, Tandon School of Engineering, Brooklyn, NY, United States

OPEN ACCESS

Edited by:

Heiko Hamann,
Universität zu Lübeck, Germany

Reviewed by:

Pawel Romanczuk,
Humboldt-Universität zu Berlin,
Germany
David Bierbach,
Leibniz-Institute of Freshwater
Ecology and Inland Fisheries (IGB),
Germany

*Correspondence:

Maurizio Porfiri
mporfiri@nyu.edu

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 08 May 2019

Accepted: 19 July 2019

Published: 14 August 2019

Citation:

Yang Y, Clément RJG, Ghirlanda S and Porfiri M (2019) A Comparison of Individual Learning and Social Learning in Zebrafish Through an Ethorobotics Approach. *Front. Robot. AI* 6:71. doi: 10.3389/frobt.2019.00071

Social learning is ubiquitous across the animal kingdom, where animals learn from group members about predators, foraging strategies, and so on. Despite its prevalence and adaptive benefits, our understanding of social learning is far from complete. Here, we study observational learning in zebrafish, a popular animal model in neuroscience. Toward fine control of experimental variables and high consistency across trials, we developed a novel robotics-based experimental test paradigm, in which a robotic replica demonstrated to live subjects the correct door to join a group of conspecifics. We performed two experimental conditions. In the individual training condition, subjects learned the correct door without the replica. In the social training condition, subjects observed the replica approaching both the incorrect door, to no effect, and the correct door, which would open after spending enough time close to it. During these observations, subjects could not actively follow the replica. Zebrafish increased their preference for the correct door over the course of 20 training sessions, but we failed to identify evidence of social learning, whereby we did not register significant differences in performance between the individual and social training conditions. These results suggest that zebrafish may not be able to learn a route by observation, although more research comparing robots to live demonstrators is needed to substantiate this claim.

Keywords: behavior, biomimetics, ethorobotics, observational learning, robotics

1. INTRODUCTION

Social learning is widespread among animals, contributing significantly to behavioral adaptation in both individuals and groups (Zentall and Galef, 1988; Leadbeater and Chittka, 2007; van Schaik, 2010; Hoppitt and Laland, 2013). In addition to elucidating a crucial adaptive mechanism, studies of animal social learning can lead to improved understanding of human pathologies to which social learning contributes, such as anxiety and phobias (Blanchard et al., 2001; Delgado et al., 2006; Mineka and Zinbarg, 2006), or in which it is affected, such as in autism spectrum disorders (Schneider and Przewlocki, 2005; Markram et al., 2008).

A long-standing question is whether social learning can be explained by associative learning mechanisms or whether it requires more sophisticated learning abilities

(Heyes, 2012b; Lind et al., 2019). Within this debate, observational learning is of special interest. Observational learning refers to learning a behavior from simple observation, without the opportunity for practice. In standard associative learning theory, learning an action requires performing it (instrumental conditioning; Pearce, 2008; Bouton, 2016). Hence, evidence of observational learning of actions would indicate a learning mechanism that is more sophisticated than associative learning (Lind et al., 2019), or possibly a modified associative mechanism (Heyes, 2001, 2012a).

Here, we study observational action learning in zebrafish, *Danio rerio* (Engeszer et al., 2007). The use of zebrafish in developmental biology has produced in-depth knowledge and powerful tools for genetic experimentation (Vascotto et al., 1997), which is being leveraged in behavioral genetics and neuroscience (Norton and Bally-Cuif, 2010). Genetic similarities with mammals (Crollius and Weissenbach, 2005) have also established zebrafish as a prime model organism for translational clinical research (Stewart et al., 2012). However, the potential of zebrafish in behavioral science is not fully realized because of the relative paucity of behavioral screening tools (Sison and Gerlai, 2010), and this is especially true in the case of learning (Gerlai, 2011). Our study is simultaneously an investigation of social learning and a contribution to the wider landscape of behavioral methods in zebrafish.

Social learning is common in fish (Brown and Laland, 2003), but existing studies do not conclusively establish learning of actions by observation. For example, fish can learn a route by following conspecifics (Laland and Williams, 1997; Laland and Williams, 1998; Reebs, 2000), but this allows them to practice the route and could be based on innate following behavior (Brown and Laland, 2003) in combination with associative learning (Lind et al., 2019). Anthouard (1987) demonstrated that naïve *Dicentrarchus labrax* learned an action more quickly after observing experienced conspecifics, but the setup enabled naïve fish to make partial responses, such as approaching and snapping, which may have facilitated learning. Because these are likely genetically predisposed responses to the sight of foraging fish (Brown and Laland, 2003), the study does not unequivocally support observational learning of an action.

Further evidence of observational learning come from studies with guppies (*Poecilia reticulata*) and sailfin mollies (*Poecilia latipinna*), showing that females can learn preferences for males by observing other females (Dugatkin and Godin, 1992, 1993; Schlupp and Ryan, 1997; Witte and Ueding, 2003; Godin et al., 2005). These results may derive either from observational action learning (learning to swim toward a specific male) or from observational learning of a preference for a stimulus (a specific male) coupled with a pre-existing response (swimming toward males in general). Because these studies bear some conceptual similarity to ours, we will consider them in more detail in the Discussion. Similarly, males *Astatotilapia burtoni* have been shown to infer the fighting ability of conspecifics by observations (Grosenick et al., 2007), but evidence that fish are capable of observational learning of actions remains scarce.

Robotics often take inspiration from nature (Brambilla et al., 2013; Kim et al., 2013; Valentini et al., 2016), but robots are

also increasingly used to study animals. In order to advance our understanding of observational learning in fish, we established a novel ethorobotics-based experimental paradigm that could afford finer control of experimental conditions. Ethorobotics represent a promising interdisciplinary research area at the interface of ethology and robotics (Webb, 2000; Partan et al., 2009; Krause et al., 2011; Halloy et al., 2013; Frohnwieser et al., 2016; Porfiri, 2018; Romano et al., 2018), in which robots whose design is inspired by animals help understanding animal behavior by allowing fine-tuned interactions. Our paradigm uses a robotic zebrafish replica as a demonstrator in order to control precisely what information is displayed to the subject. The replica is built to mimic the morphology, size, coloration, and motion of live zebrafish. Its motion is controlled in two dimensions (2D) via a Cartesian plotter, which allows for the implementation of realistic swimming patterns, in terms of both movement trajectory and body undulations. In previous work, we showed that equivalent robotic replicas elicit approach responses in live fish, similar to social behavior that is generally exhibited toward conspecifics (Ruberto et al., 2016, 2017; Kim et al., 2018). For example, zebrafish show a similar preference for associating with a replica and a conspecific in binary choice tests (Ruberto et al., 2017).

Here, to ensure that learning could proceed only by observation, rather than by practicing the correct behavior, we confined subjects in a small area during demonstrations. The task consisted of learning to approach one of two doors in order to gain proximity to a shoal of conspecifics. Subjects in the social training condition observed the robotic replica approaching both the incorrect door, to no effect, and the correct door, whose opening is triggered automatically by a real-time video tracking system. Subjects in the individual training condition learned without the demonstrator and provided a control group. In this task, zebrafish learned a preference for swimming to the correct door, but we observed no effect of social vs. individual training: fish that observed the demonstrator did not learn more quickly, and did not spend more time in proximity of the correct door compared to fish who learned individually. We consider the implications of these results and further developments in the Discussion.

2. MATERIALS AND METHODS

This section is organized as follows. First, we detail the robotics-based experimental setup, focusing on both the hardware and software. Then, we present the experimental procedure, including the animals, the structure of the trials, and the experimental groups used in the study. Finally, we articulate our data analysis, consisting of a wide range of behavioral and learning measures, along with multivariate statistical models. All data and code for analysis are available as **Supplementary Information**.

2.1. Robotics-Based Experimental Setup

2.1.1. Hardware

The experiment was performed in a glass tank (74 × 30 × 30 cm; length, width, and depth) supported by a custom frame built with T-slot bars (McMaster, Robbinsville, NJ, USA), shown in

Figure 1A. The bottom of the tank was raised 29 cm above floor level to fit the Cartesian plotter used to maneuver the replica. To minimize extraneous visual stimuli, dark curtains were mounted around the tank. The bottom and side walls of the tank were covered by white contact paper (McMaster, Robbinsville, NJ, USA) to ease video tracking.

The tank was divided into three sections with lengths of 30, 34, and 10 cm using two partitions: a transparent partition with two doors and a one-way glass partition, see **Figure 1B**. The one-way glass partition, with thickness of 5.9 mm, was used to house a shoal of 10 zebrafish, preventing them to see the subject and interact with it. The lateral section delimited by the partition with the doors is the focal compartment where subject behavior was monitored. The middle section was intended to maintain some distance between the subject and the stimulus group, such that the subject would need to explore the partition with the doors to gain proximity to the group.

The doors were cut from a transparent acrylic sheet (McMaster, Robbinsville, NJ, USA), and they were held in place by acrylic guides glued to the main partition, so that they could only move along the vertical direction. Each door was 1.5 body lengths (BLs) wide to allow the subject and the replica to smoothly transit through them. The doors were located at 1/4 and 3/4 of the width of the partition, symmetrically with respect to the middle horizontal axis.

Each door was connected to a pulley via a transparent fishing line (Berkley Trilene XT Extra Tough, Pure Fishing, Inc., Columbia, SC, USA), shown in **Figure 1C**. The pulleys (external diameter of 13 cm and internal diameter of 12 cm) consisted of a 3D printed plastic plate and a servo motor (HS-5086 WP, Hitec RCD USA, Inc., Poway, CA, USA). The motors were activated by a microcontroller (Arduino Uno, Arduino Srl, Italy).

The replica was fabricated using a 3D-printed mold (Ultimaker 2+, Ultimaker B.V., Geldermalsen, The Netherlands), where we poured a flexible silicone mixture (Smooth-On, Inc., Macungie, PA, USA), see **Figure 2**. The use of silicone instead of rigid material allows a more naturalistic bending of the replica's body during its motion through the experimental tank, which could increase its biomimicry and acceptance by the live zebrafish (Romano et al., 2017, 2019a). The replica was then painted with silicone-based paint (Smooth-On, Inc., Macungie, PA, USA). The replica was attached to a transparent rod, clamped to a 3D-printed base, which, in turn, was magnetically connected to a Cartesian plotter (XY Plotter Robot Kit, Makeblock Co., Ltd, Shenzhen, China) to control its motion. The plotter was placed below the tank to minimize acoustic and visual confounds. As discussed in a separate, focused publication, this platform enables realistic swimming motion of the robotic replica with accurate positioning and fast reaction time (DeLellis et al., submitted).

Above the tank, we installed two cameras at a height of 137 cm from the floor, see **Figure 1C**. A Logitech C920 (Newark,

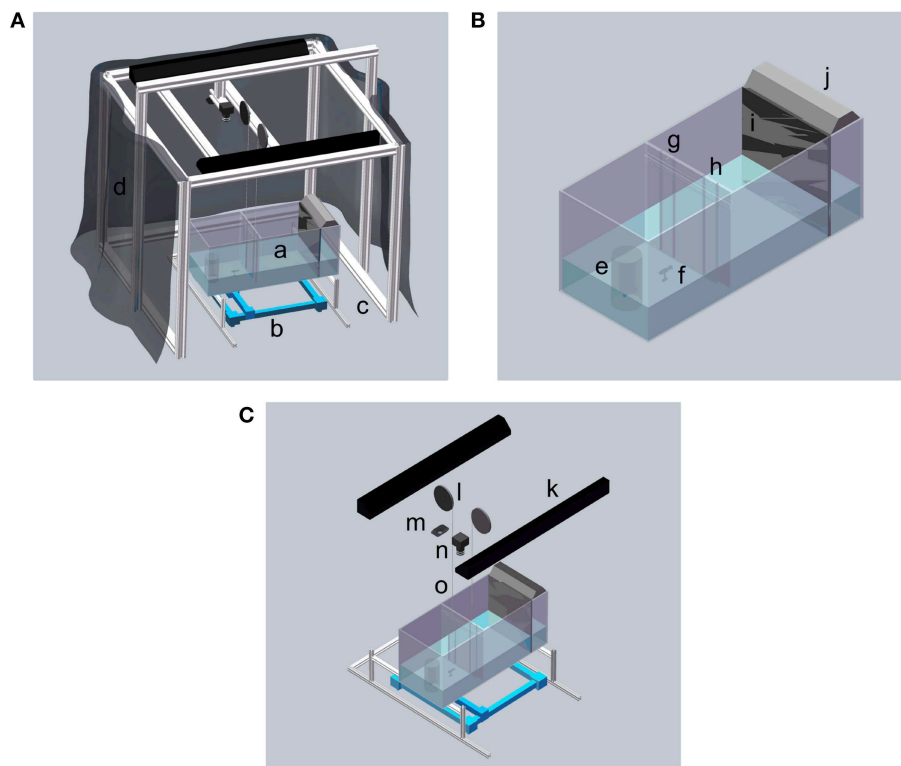
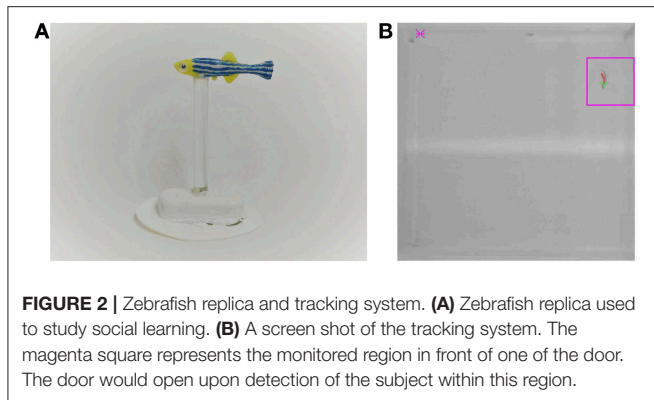


FIGURE 1 | Schematic of the experimental apparatus. **(A)** Overview of the experimental apparatus. **(B)** Detailed view of the experiment tank. **(C)** Detailed view of the hardware. a Experimental tank; b 2D Cartesian plotter; c Aluminum frame made by T-slot bars; d Curtain; e Transparent cylinder and live zebrafish; f Zebrafish replica; g and h Acrylic transparent doors; i One-way glass; j 12-inch light; k 36-inch light; l Pulley system; m Logitech webcam; n Flea3 camera; and o Fishing line.



CA, USA) webcam was used for tracking the position of the subject in the focal compartment with a resolution of 640×480 pixels. A Flea3 FL3-U3-13E4C USB camera (FLIR Integrated Imaging Solutions Inc., Richmond, BC, Canada), with a higher resolution of 1280×1024 pixels was used to capture the entire experimental tank and monitor the subject's interaction with the shoal, for reward timing. This camera was controlled by software FlyCapture SDK (FLIR Integrated Imaging Solutions Inc., Richmond, BC, Canada).

Uniform illumination was provided by two 36-inch, 30 W white fluorescent lights (All-Glass Aquarium Co., Inc., Franklin, WI, USA) mounted along the sides of the tank at a distance of 110 cm from the floor. A third light, a 12-inch fluorescent strip light with a power of 8 W (All-Glass Aquarium Co., Inc., Franklin, Wisconsin, USA), was used for additional illumination of the stimulus region so that the group could be seen clearly by the subject, see **Figure 1B**.

2.1.2. Software

The apparatus was operated from a PC using a custom software developed in Matlab 2018a (The MathWorks, Inc., Natick, MA, USA). Live tracking of the subject fish was based on Matlab computer vision toolbox, including detection of moving objects and localization of object centroids. At each tracking step, two gray-scale frames were acquired by the Logitech C920 Pro camera and clipped to a fixed region of interest containing the tank. Frames were captured at 20 Hz. The first frame was subtracted from the second, yielding an image with outlines of the fish and, if present, of the replica. This image was processed to remove noise, fill in the outlines of the targets, and estimate the targets' positions from the centroids of the filled outlines. The fish and replica were distinguished from each other by using the input to the Cartesian plotter. If this procedure failed to locate the fish, a Kalman filter was used to extrapolate from previous frames; in DeLellis et al. (submitted), details of the tracking system are presented.

The tracking system also monitored a square region of 2×2 BL in front of the correct door to detect the presence of the subject fish, see **Figure 2**. The latter could be opened and closed by sending appropriate commands from the PC to an Arduino Uno controller. The replica was controlled by programming sequences of 2D coordinates and sending them from the PC to another Arduino microcontroller. The sequence

was generated by implementing a stochastic mathematical model of zebrafish swimming, which we have established in our previous work (Mwaffo et al., 2015, 2017; Zienkiewicz et al., 2018). The model captures the typical burst-and-coast swimming style of zebrafish, where sudden tail beats are followed by longer coasting phases. Details of the implementation are presented in DeLellis et al. (submitted).

2.2. Experimental Procedure

2.2.1. Animals

Zebrafish were purchased from Carolina Biological Supply Co. (Burlington, NC, USA). We used a total of 56 fish: 36 fish were used as focal subjects, with a female/male ratio of 5:4 and an average BL of around 3 cm. An equal number of focal subjects (18) were used for each condition. The remaining 20 fish were used to form the stimulus shoals, with an equal sex ratio and similar average BL as the experimental subjects.

Animals were housed in 37.5 L (10 gallons) vivarium tanks (Pentair Aquatic Eco-systems Locations, Cary, NC, USA) with a density of no more than 10 fish per tank. Water temperature and acidity were kept at 26° and 7.2 pH. Housing lights were maintained for a period of 12 h light/12 h dark. The fish were fed commercial flake food (Nutrafin max; Hagen Corp., Mansfield, MA, USA) once per day around 7 PM.

After the fish habituated to the housing tank for at least 15 days, they were individually tagged with silicone-based visible implant elastomers (VIEs) (Northwest Marine Technology Inc., Shaw Island, WA, USA). Before tagging, the colored part and the curing agent of the VIE were mixed with a proportion of 10:1, and the fish was anesthetized to avoid unnecessary wounds. The VIE was injected bilaterally on two locations near the head. Tag colors were randomly selected and combined among white, purple, blue, and yellow. After tagging, the fish were given at least 14 days of recovery in their housing tank.

2.2.2. Trial Structure

The experiment investigated whether zebrafish would learn to open a door in order to join a shoal of 10 conspecifics, visible behind a one-way glass, see **Figure 1**. Each subject was trained for 20 trials either individually, where it would learn alone how to open the correct door, or socially, where it could observe a robotic zebrafish replica demonstrate door opening at the beginning of each trial.

For the first 10 min of each trial, experimental subjects were confined in the focal region via a transparent plastic cylinder (diameter 8 cm). During this time, subjects in the social training condition observed the robotic replica demonstrate door opening as sketched in **Figure 3**, while subjects in the individual training condition simply waited.

The demonstration by the replica entailed the following steps. At the beginning of each trial, the replica interacted with the focal subject for 30 s, following a trajectory generated via our stochastic model of zebrafish locomotion (Mwaffo et al., 2015, 2017; Zienkiewicz et al., 2018) with an attraction point at the center of the cylinder that housed the experimental subject. This resulted in the replica swimming in the focal region, while frequently approaching the subject and "wall kissing" the cylinder. The replica then swam in a straight line to the correct

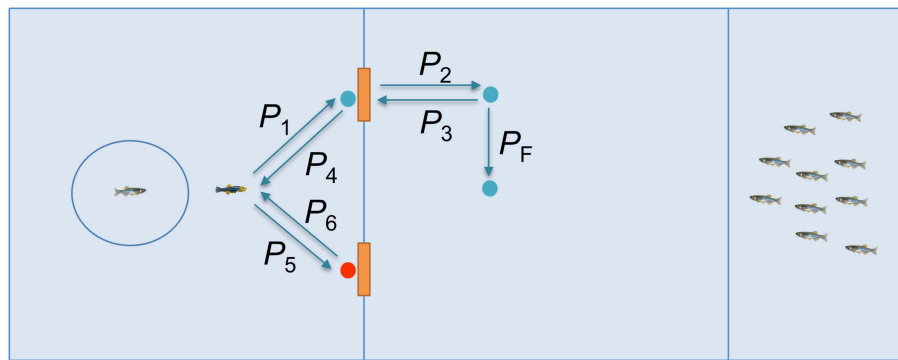


FIGURE 3 | Illustration of the demonstration by the replica in the social training condition.

door (P_1) and started tail beating for 3 s. As a result, the door would open and the replica swam through (P_2), before stopping for 5 s while beating its tail. After these 5 s, the replica would go back to the focal region (P_3 and P_4) to resume the interaction with the subject for another 30 s. Then, it approached the wrong door and station there, beating its tail for 3 s, but the door would not open. After this cycle was repeated 6 times over 10 min, the robotic replica transited through the correct door and move to the final position (P_F), facing the stimulus group until the subject finished the task.

At the end of the first 10 min of each trial, subjects were released from the cylinder and allowed to swim freely in the focal compartment until they opened the door, within a time limit of 30 min. The open door allowed subjects to access to the central compartment, bringing focal subjects closer to the shoal of conspecifics. The door would open if the focal subject stationed in a 6×6 cm, unmarked zone in front of the correct door (Figure 2) for at least 3 s out of any 5 s. The triggering process was controlled automatically through the tracking system described above. Once the door opened, the subject were rewarded by being allowed to swim for 2 min close to the conspecifics in the central compartment.

Learning was assessed by measuring changes in proximity to the two doors across learning trials, as well as during three 30 min tests conducted before the first trial, after trial 10, and after trial 20. During these three additional tests, both doors remained closed, and no robotic replica was present. The subject was confined in region A for 10 min and then released in region B for an additional 10 min.

Correct functioning of the apparatus was tested prior to the beginning the experiment, using several pilot fish not included in the experiment. Sample videos of individual and social training are provided as **Supplementary Information**. Some of these trials along with a preliminary description of the experiment have been presented in a recent meeting (Yang et al., 2019).

Upon inspecting the data, we discovered that performance was consistently better when subjects had to swim to one of the doors, and that this preferred door changed between the first two batches, that is, depending on the orientation of the apparatus. This pattern indicates the presence of an uncontrolled factor external to the apparatus, which biased exploration toward one

of the two sides. We have thus coded all data to indicate whether the correct door was, for each subject, on the overall preferred or non-preferred side.

We discovered this bias after completing the individual training condition, and we kept the same experimental layout for the social training condition to ensure that the data were comparable. We speculate that fish might have been attracted to the familiar sound of the housing tanks, which were ~ 2 m from the tank on the preferred side. In the future, we will orient the apparatus so that the housing tanks will lie behind the shoal of conspecifics, thus reinforcing their attractive effect rather than introducing a side bias.

2.2.3. Experimental Groups

The experiment ran from June to September 2018. In each trial, only one fish was trained. Each fish (a total number of 36) was tested twice per day, once in a morning session between 9 a.m. and 1 p.m., and once in an afternoon session between 2 and 6 p.m. Each condition (individual or social training) was performed on two batches of nine subjects each. The assignment of the correct door was fully counterbalanced across conditions, batches, and subjects. A consistent sex ratio of five females to four males was used in each batch. In between trials, subjects were housed in four tanks, keeping together individuals of the same sex that were assigned the same correct door. Twenty more fish were used as stimuli, split into two groups of 10 individuals each. In both conditions, the stimulus group used in the morning sessions of the first batch was used in the afternoon sessions of the second batch, and vice-versa.

Before each training session, the tank was filled with new tap water and a drop of coating (AcquaSafe Plus, Tetra, Blacksburg, VA, USA) to neutralize pollutants, such as chlorine, chloramines, and heavy metals, and strengthen bacterial beds. The water height was always 10 cm and the temperature was maintained at around 27°C .

2.3. Data Analysis

2.3.1. Behavioral and Learning Measures

The raw data collected in the experiments consisted of the subject's trajectory and the door triggering time acquired by the real-time tracking system. From these data, we computed the

parameters defined in **Table 1** to measure behavior and learning. The two main measures of learning are T , the time between the release of the subject from the cylinder and the door opening (right-censored at 30 min on unsuccessful trials) and preference index (PI), defined as the time spent in proximity of the correct door over the time spent in proximity of either door (that is, within the region used to trigger door opening, see **Figure 2**). If learning is successful, we expect T to decrease over trials and PI to increase from a value close to 0.5 (no preference) to a value above 0.5 (preference for the correct door).

To fully characterize zebrafish behavior, we also computed the following measures:

- H , the entropy of the trajectory. If the subject learns to approach and open the door efficiently, its swimming should become less random and thus the entropy of the trajectory should decrease.
- θ_C , correct heading, defined as the absolute value of the angle between the current heading of the fish and the vector from the fish to the center of the correct door. A value of 0° indicates swimming directly toward the door, while nonzero values indicate less precise swimming. This parameter

and the following ones were computed based on recorded swimming trajectories.

- RI, the reward index that quantifies the subject's preference for swimming close to conspecifics vs. far from them. We used this measure to evaluate whether the stimulus shoal was attractive to the subjects, as assumed in our experimental setup.
- Locomotor activities, in terms of freezing time, average speed, average angular speed, and average acceleration (Macrì et al., 2017). We used these indices to evaluate whether exposure to the replica altered the behavior of the animals.
- A, avoidance response for the door after it opened, scored by evaluating whether the distance between the focal fish and the center of the correct doors reached a value larger than two body lengths within the 15 s following door opening. This parameter was computed using the first and last training trial for each fish, for a total of 72 videos overall.
- PI_m , the modified preference index was used to measure the preference of focal fish for the replica, which was stationed between the two doors. We divided the tank into three rectangular regions of equal area along its width, and we computed this index as the fraction of time spent in the middle region of the tank.

TABLE 1 | Behavioral and learning measures.

Symbol	Description	Formula
T	Time between subject release and door opening	—
τ_C	Time spent in a 6×6 cm region in front of the correct door	—
τ_I	Time spent in a 6×6 cm region in front of the incorrect door	—
T_i	Time spent within 3 cm of wall i	—
PI	Preference index for the correct door	$\tau_C / (\tau_C + \tau_I)$
RI	Reward index for associating with conspecifics (reward value)	$T_1 / \sum_{i=1}^4 T_i$
H	Entropy of the trajectory	$-\sum_{i=1}^{100} P_i \log_2 P_i$
θ_C	Angle between the fish heading and the direction to the correct door	—
v_t	Speed	$\frac{\ \mathbf{x}_{t+1} - \mathbf{x}_t\ }{\Delta_t}$
a_t	Acceleration	$\frac{\ \mathbf{v}_{t+1} - \mathbf{v}_t\ }{\Delta_t}$
ω_t	Turn rate	$\frac{1}{\Delta_t} \cos^{-1} \left(\frac{\mathbf{v}_{t+1} \cdot \mathbf{v}_t}{\ \mathbf{v}_{t+1}\ \ \mathbf{v}_t\ } \right)$
F	Freezing time (time that the fish moved <4 cm over a rolling window of 2 s)	—
A	Avoidance response for the door after it opened	—
PI_m	Modified preference index for the replica, held in place at the middle of the tank width	—

In the Formula column, — indicates a primary variable derived directly from video tracking. The walls are ordered such that wall 1 is the transparent partition separating regions A and B. In the formula for H , we partitioned the 30×30 cm region B into a 10×10 square grid, so that the length of each square is ~ 1 body length; therein, P_i is the fraction of video frames in which the subject was in grid cell i . Δ_t is the interval between frames, that is, 0.05 s. For calculations, raw trajectory data were smoothed using a moving average with a span of 4 frames, such that \mathbf{x}_t is the smoothed 2D position in the tank.

2.3.2. Statistical Model

Using linear mixed effects modeling, with subject as a random factor, we related door triggering time (T), preference for the correct door (PI), and the other variables in **Table 1** to the independent variables “condition” (individual or social training), “correct door location,” and either “trial,” for data collected during training trials, or “test,” for data collected during test trials. These independent variables and their possible values are summarized in **Table 2**. The independent variable “correct door location” encapsulates the experimental bias that we observed in our data. Although the correct door was counterbalanced across subjects and the apparatus was rotated 180° in between the two batches of each condition, fish consistently displayed better performance when they had to swim to one of the two doors.

At first, a linear mixed full model with the global ID of the fish as random effect was built. Non-significant interaction terms were then discarded from the model. In order to correct for false positive due to multiple testing, we took into account that each independent variable entered two statistical tests relative to test data (preference index and heading), and three tests relative to training data (preference index, heading, and door triggering time). Conservatively, we applied an alpha level of $0.050/3 \simeq 0.017$.

TABLE 2 | Independent variables used in data analysis. See the text for details.

Variable	Type	Values
Condition	Unordered factor	Individual, Social
Test	Ordered factor	0, 10, 20
Training	Numeric	1, ..., 20
Correct door location	Unordered factor	Preferred side, Non-preferred side

We also used Levene's test to investigate differences in variability of the dependent variables across different combinations of independent variables.

Data analysis was conducted in Emacs Org-mode (Dominik, 2010; Schulte et al., 2012) and R version 3.5.0 (R Core Team, 2018) with packages *car* (Fox and Weisberg, 2011), *data.table* (Dowle and Srinivasan, 2018), *readxl* (Wickham and Bryan, 2018), *effects* (Fox and Weisberg, 2018), and *ascii* (Hajage, 2009).

3. RESULTS

3.1. Test Data

Test trials provide the best assessment of differences between social and individual training because they took place on days in which no training occurred and because, contrary to training trials, the replica was absent even in the social condition. Thus, any difference between conditions would be attributable to learning rather than to short-term influence of the replica, such as on emotional response.

Table 3 shows type II ANOVA results for the preference between correct and incorrect door (PI in **Table 1**). We found a

main effect of test, showing an improvement from no preference to about 62% preference for the correct door, and an interaction between test and location of the correct door, illustrating that the improvement over tests occurs primarily when the correct door is on the preferred side of the tank, see **Figure 4**. There was no effect of social vs. individual training, see **Figure 4**. There was also no significant difference between the variability of the preference across groups of subjects [Levene's test: $F_{(11,96)} = 1.13$, $P = 0.347$].

A type II ANOVA of heading direction toward the correct door (θ_C) yields similar results, see **Table 4** and **Figure 5**. In addition, we found a significant interaction between training condition and correct door location, indicating less accurate heading for the social training condition when the correct door was on the preferred side of the tank, but not when the door was on the non-preferred side.

Approaching conspecifics appeared to be an adequate motivation for the focal fish, as they spent considerable time close to the wall with the doors. Of all the time spent within one body length (3 cm) of the walls, an average of 87% was spent near this wall.

The modified preference index, assessing the preference of the fish toward the replica, tended to decrease with the number

TABLE 3 | Type II ANOVA table for the preference index (PI) during test trials, as a function of training condition, correct door location, and test.

	χ^2	Df	p
Condition	0.82	1	0.364
CorrectDoorLocation	1.44	1	0.230
Test	10.44	2	0.005
Condition:CorrectDoorLocation	0.91	1	0.339
Condition:Test	2.10	2	0.350
CorrectDoorLocation:Test	12.89	2	0.002
Condition:CorrectDoorLocation:Test	1.56	2	0.459

Here and in the remaining tables, we write "CorrectDoorLocation" to identify with a single word the corresponding independent variable. Bold values indicate statistically significant results.

TABLE 4 | Type II ANOVA table of heading direction during test trials, as a function of training condition, correct door location, and test.

	χ^2	Df	p
Condition	1.55	1	0.213
CorrectDoorLocation	0.39	1	0.535
Test	14.30	2	0.001
Condition:CorrectDoorLocation	8.02	1	0.005
Condition:Test	0.05	2	0.975
CorrectDoorLocation:Test	8.45	2	0.015
Condition:CorrectDoorLocation:Test	0.07	2	0.964

Bold values indicate statistically significant results.

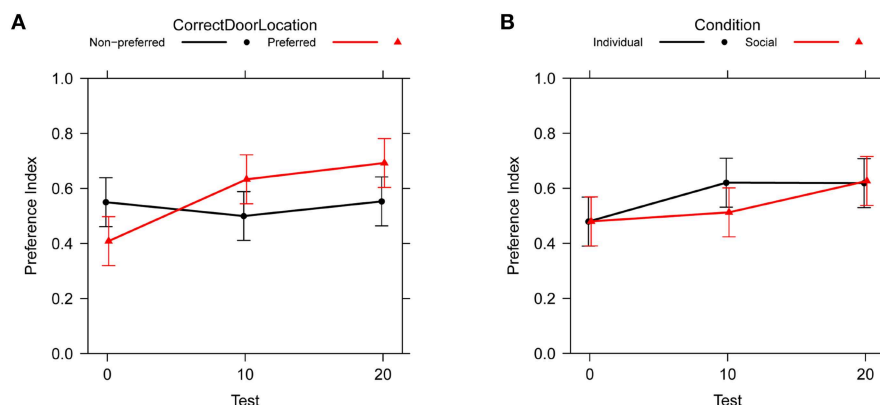


FIGURE 4 | Change in preference index across test trials (PI in **Table 1**). Bars are 95% confidence intervals. **(A)** Comparison between preferred and non-preferred correct door location. **(B)** Comparison between social and individual training.

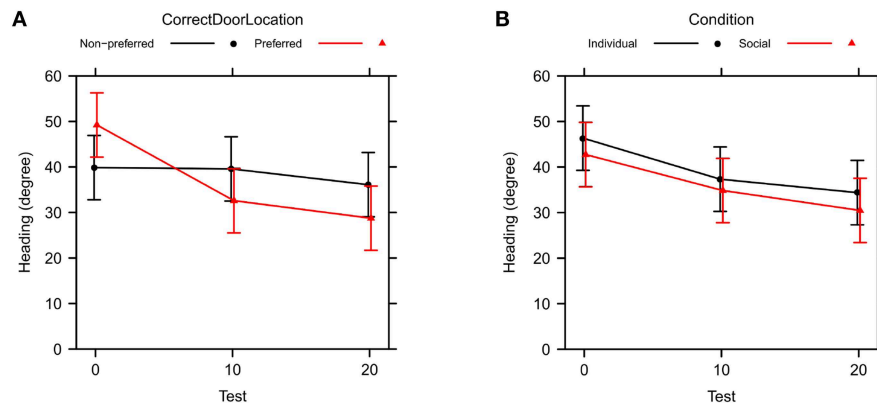


FIGURE 5 | Change in heading precision toward the correct door across test trials (θ_C in **Table 1**). Bars are 95% confidence intervals. **(A)** Comparison between preferred and non-preferred correct door location. **(B)** Comparison between social and individual training.

of tests and did not show a significant variation between social and individual training. Examining the effect of the number of training sessions, we found that the tendency to explore the doors increased after 10 training trials (**Supplementary Information**).

We further investigated whether other behavioral variables differed across tests and conditions. Type II ANOVAs using the dependent variables in **Table 1** and the independent variables in **Table 2** generally failed to show differences between social and individual training (**Supplementary Information**). We did observe some non-specific changes in swimming behavior over successive tests, consistent with decreased arousal as the fish become acquainted with the testing tank, such as a decrease in wall following and turn rate.

3.2. Training Data

We analyzed data from training trials similarly to data from test trials, with the difference that the test dependent variable is replaced by the trial variable in type II ANOVAs. Additionally, we analyzed the time subjects took to trigger the opening of the door (T). ANOVA of triggering time shows no significant effect of social vs. individual condition (**Table 5**). Thus, zebrafish did not learn to open the door faster, whether learning alone or with the replica. Fish, however, did spend more time close to the correct door as training progressed (**Table 6**), and showed increased precision in heading toward the correct door (**Table 7**). Both the preference for the correct door and the precision in heading were stronger when the correct door was on the preferred side of the tank.

Similarly to test data, we also found an interaction between training condition and correct door location, in that subjects trained with the replica did slightly worse than subjects trained individually when the correct door was on the preferred side, see **Figures 6, 7**. Overall, these results are consistent with the focal fish being attracted to locations where it saw the robotic replica, regardless of whether the replica successfully swam through a door.

With respect to potentially aversive effect of the door opening mechanism, we found that focal fish moved away in 80.6 %

of the trials when the door started opening (58 out of 72 trials, individual and social learning combined). As a result, we cannot exclude that the door opening might induce a short-term fear reaction on the subjects. The modified preference index, assessing the preference of the fish toward the replica, showed an interaction among the condition, trials, and correct door location. Similar to the analysis of the test data, we found that the tendency of the animals to explore the bottom and top third of the tank where the doors resided increased with the training trials.

The other variables in **Table 1** did not differ depending on the training condition, but sometimes we found an effect of the location of the correct door or an interaction between the condition and location of the correct door. For example, trajectory entropy was higher in the social training condition, when the correct door was on the preferred side (Condition \times Correct door location interaction: $\chi^2_{(1)}=43.59$, $P < 0.001$), indicating more erratic swimming, consistently with the analogous effect noted above for heading direction. Changes in swimming behavior during training were consistent with those observed during test trials, see above.

4. DISCUSSION

We established a novel experimental paradigm, which capitalizes on recent advances in robotics and automated video-tracking to afford fine control of experimental conditions in observational learning. The proposed paradigm features a biologically-inspired zebrafish replica that is controlled by a robotic platform along trajectories, which demonstrate to experimental subjects a route that would allow them to gain proximity to a group of conspecifics. The route consisted of transiting through one of two transparent doors, which automatically opened when the animal spent sufficient time in its proximity. The setup can also be used to investigate individual learning and, as we have done here, to compare individual and social learning.

In addition to its technical innovations, the proposed experimental paradigm appears highly motivating to zebrafish. During the trials, subjects spent considerable time near the

TABLE 5 | Type II ANOVA table for door triggering time (T) during training trials, as a function of training condition, trial, and correct door location.

	χ^2	Df	p
Condition	1.65	1	0.199
Trial	0.68	1	0.411
CorrectDoorLocation	8.88	1	0.003
Condition:Trial	0.06	1	0.799
Condition:CorrectDoorLocation	0.09	1	0.762
Trial:CorrectDoorLocation	2.82	1	0.093
Condition:Trial:CorrectDoorLocation	0.22	1	0.638

Bold values indicate statistically significant results.

transparent partition with the doors, and once they went through the door they swam up to the conspecifics and attempted to interact with them through the one-way glass. While motivating, the setup did not elicit undesired stress responses; experimental subjects swam normally and rarely froze during trials (**Supplementary Information**). Thus, the proposed robotics-based paradigm could constitute a promising avenue for investigating learning in zebrafish, and can be extended to other organisms.

In our experiment, zebrafish did not open the door faster over successive trials, but they learned to preferentially approach the area near the correct door, and they oriented toward this area more over the course of the experiment. However, fish exposed to the robotic demonstrator did not learn more quickly than fish trained individually, despite having 120 experiences in which the replica approached, opened, and swam through the correct door, and an equal number of experiences in which the incorrect door remained closed when the replica approached it. We thus failed to show observational learning of approach to the correct door location. The only effect of the replica on the subject we found was to reduce the experimental bias toward one of the door locations, which is consistent with the focal fish being attracted to the replica. This failure should not be attributed to a ceiling effect as the task proved difficult enough that social training could have produced a substantial improvement in performance, over the baseline provided by individual training.

Overall, our results suggest that zebrafish social learning may depend on following conspecifics, and thus on experiencing first-hand the relevant stimulus-response contingencies. This hypothesis is consistent with existing demonstrations of social learning in zebrafish (Lindeyer and Reader, 2010) and fish in general (Brown and Laland, 2003), where either following or approach responses were possible. More generally, the hypothesis that social learning requires trying out the behavior to be learned, rather than just observing it in others, is of great relevance to current theory of social learning (Heyes, 2012b; Lind et al., 2019). Previous work has shown that robots can be used to influence the response of animal in longitudinal studies with sequential exposure to robotic stimuli. For example, Locusts (*Locusta migratoria*) learned to escape preferentially on a side, following exposure to a robotic Gecko coming from the opposite side (Romano et al., 2019b). In our case, the robot is used to

TABLE 6 | Type II ANOVA table for the preference index (PI) during training trials, as a function of training condition, trial, and correct door location.

	Chisq	Df	p
Condition	3.22	1	0.073
Trial	6.09	1	0.014
CorrectDoorLocation	33.09	1	<0.001
Condition:Trial	0.68	1	0.408
Condition:CorrectDoorLocation	6.45	1	0.011
Trial:CorrectDoorLocation	0.46	1	0.496
Condition:Trial:CorrectDoorLocation	0.02	1	0.885

Bold values indicate statistically significant results.

TABLE 7 | Type II ANOVA table of heading direction toward the correct door (θ_C) during training trials, as a function of training condition, trial, and correct door location.

	Chisq	Df	p
Condition	0.13	1	0.723
Trial	9.52	1	0.002
CorrectDoorLocation	59.87	1	<0.001
Condition:Trial	3.19	1	0.074
Condition:CorrectDoorLocation	14.30	1	<0.001
Trial:CorrectDoorLocation	1.18	1	0.278
Condition:Trial:CorrectDoorLocation	1.16	1	0.281

Bold values indicate statistically significant results.

proxy a trained conspecific that acts as a demonstrator in a social learning task, while in the study by Romano et al. (2019b) a robotic predator served as aversive stimulus to condition the subjects spatially. Our experimental paradigm could serve as inspiration to design similar studies in other species.

The work that most closely resembles ours is, perhaps, that of Dugatkin and coworkers on mate choice copying in female guppies (Dugatkin and Godin, 1992, 1993; Godin et al., 2005). In these experiments, a female subject could observe another female approaching one of two males, which resulted in the subject subsequently preferring to approach the same male. This result is seemingly at odds with ours, for which several explanations are possible. First, the capacity for observational learning may be dependent on which behavior system is engaged. Because a single mate choice is likely more important for fitness than a single choice of swimming direction, mate choice decisions may have evolved to take into account social information to a larger extent [indeed, Dugatkin and Godin (1993) showed that it is mainly inexperienced females that copy the preferences of others]. Second, it is possible that the subjects of Dugatkin and coworkers learned a preference for a stimulus (a male) rather than an approach response, which then resulted in approach because of a hardwired predisposition to approach males. In our experiment, however, there was no conspicuous visual stimulus for which a preference could be learned. Lastly, the learning observed by Dugatkin and coworkers may have been driven by responses performed during the observation phase. Female

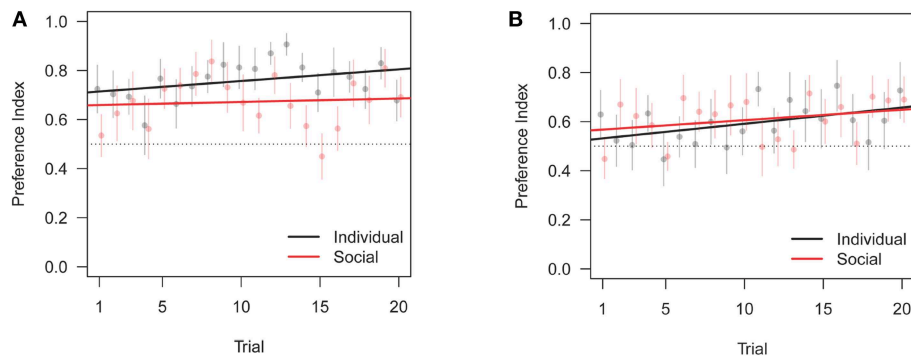


FIGURE 6 | Change in preference index across training trials (PI in **Table 1**). Bars are standard errors of the mean. **(A)** Preferred correct door location. **(B)** Non-preferred correct door location.

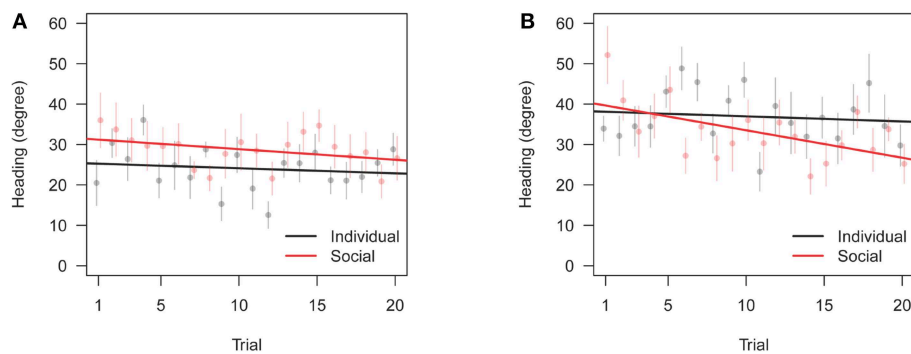


FIGURE 7 | Change in heading precision toward the correct door across training trials (θ_C in **Table 1**). Bars are standard errors of the mean. **(A)** Preferred correct door location. **(B)** Non-preferred correct door location.

subjects, in fact, had to choose between two males at the opposite ends of a tank. Observing the female demonstrator would thus have biased the subject to turn toward one end of the tank, which may have been instrumental in establishing the preference for swimming in that direction once this became possible. In our experimental setup, on the other hand, the two doors were both in front of the subject, and the scope for orienting differentially toward one or the other was much more limited.

Related work by Webster and Laland involved food, which could offer a more motivating stimulus than a shoal of conspecifics. In these studies, the demonstrator also displayed feeding behavior, which is likely to convey additional information, compared to swimming toward a particular location. Furthermore, with food patches, Webster and Laland (2017) demonstrated the ability of both social and non-social species to use social information in the determination of the better patch. Nine-spined Stickleback (*Pungitius pungitius*) were shown to be more likely to travel toward the location where they had previously observed other individuals feeding (Webster and Laland, 2015), while social learning was more likely observed when predation risk was higher in Minnows (*Phoxinus phoxinus*) (Webster and Laland, 2008). The difference in results between

our experiment and the above-mentioned studies suggests many opportunities for further investigation.

Additional caution in drawing conclusions about zebrafish social learning is advised given that our study is the first attempt to disentangle observational learning from following, and given the novelty of our experimental paradigm. For example, we cannot exclude that a live zebrafish would have been a more effective demonstrator, although in previous work we established that zebrafish associate with the replica and with live conspecifics to similar extents, when given the choice (Ruberto et al., 2016, 2017; Kim et al., 2018). The replica also demonstrated the door-opening behavior with much more precision and consistency than a live fish could have done. Our task, however, might have been more difficult than other tasks in the zebrafish learning literature, since it required experimental subjects to approach a small area and station there for 3 s out of any 5 s window. This behavior is more complex than behaviors investigated in other studies, in which subjects simply had to swim in one or another direction without a time requirement (Bilotta et al., 2005; Xu et al., 2007; Pather and Gerlai, 2009; Sison and Gerlai, 2010; Morin et al., 2013). Our task also lacked salient visual cues distinguishing the correct door from the incorrect one, although previous work suggests that zebrafish can improve

substantially in a spatial discrimination in fewer than 10 trials (Arthur and Levin, 2001). Finally, our task required the fish to remember which door the replica had swam through, although this memory had to be maintained only for a few seconds. To evaluate whether observational learning could be more effective in different circumstances, we will perform further experiments with visually marked doors, a reduced time to trigger door opening, and a shorter interval between the replica crossing the door and the subject being released. We will also evaluate whether allowing zebrafish to follow the robotic replica leads to better learning.

While this is the first robotics-based setup for zebrafish learning, a few previous efforts have explored other automation techniques. For example, Pather and Gerlai (2009) utilized computer-animated images of zebrafish as rewarding stimuli in a shuttle box task, while (Hicks et al., 2006) used real-time video tracking to deliver rewarding or punishing stimuli, in the form of a change in illumination and brief electric shock, Gerlai et al. (2009) showed that zebrafish react to computerized images of a predator, and Fangmeier et al. (2018) demonstrated the possibility to use automated video stimulus to quantify behavioral traits in zebrafish. Here, we took a significant step forward by combining engineered stimuli with real-time control, affording the possibility of maneuvering them in the entire experimental tank. For example, compared to the experimental modifications in Hicks et al. (2006), our approach offers an additional independent variable to explore social learning, by enabling, for the first time, high-precision demonstration through a biologically-inspired replica.

The potentially negative effects on learning of the door opening mechanism and the presence of the robot seem to be limited. Although focal fish might have initially displayed an aversive response toward the door as it started opening, they eventually went through the door to interact with the fish shoal. The short-term avoidance reaction is likely due to the mechanical noise from the door movement and the concurrent water motion, which did not last long enough to significantly affect their motivation to join the shoal. The modified preference index significantly decreased in both training and test trials, indicating that, over time, the focal fish increasingly preferred to spend time in the vicinity of the doors rather than close to the replica that was visible through the partition. Thus, the potential attraction toward the replica did not significantly reduce fish motivation to explore the doors.

In conclusion, we have presented a novel robotics-based experimental paradigm that enables us to study both social and individual learning in zebrafish, with many possible variations in experimental parameters. Beyond zebrafish, the setup can be adapted to investigate social learning in other animal species for which ethorobotics-based approaches have been previously explored, including, other fish species, such as guppies (Landgraf et al., 2016; Bierbach et al., 2018a) and mollies (Bierbach et al., 2018b), insects, such as bees (Landgraf et al., 2018) and cockroaches (Halloy et al., 2007), and mammals, such as tree

squirrels (Partan et al., 2009), dogs (Kubinyi et al., 2004), and rats (Takanishi et al., 1998), and even to invertebrates such as cephalopods, for example by using prey items as motivating stimulus rather than a shoal of conspecifics. The data presented above suggest that our paradigm has the potential to contribute new knowledge to the experimental analysis of learning in fish and other aquatic animals.

DATA AVAILABILITY

The supporting data and codes to reproduce the analysis and results of the paper has been uploaded as **Supplementary Information**.

ETHICS STATEMENT

All animal procedures were approved by the University Animal Welfare Committee of New York University under protocol number 13–1424.

AUTHOR CONTRIBUTIONS

SG and MP designed the research. YY designed and developed the experimental setup. YY and RC conducted the experiments. YY wrote a first draft of the manuscript and RC offered comments. SG and MP wrote the final draft. All the authors analyzed the data, discussed the results, and reviewed and approved the final draft.

FUNDING

This work was supported by the National Science Foundation under grant numbers CMMI-1433670 and CMMI-1505832, the National Institutes of Health, National Institute on Drug Abuse under grant number 1R21DA042558-01A1, the Office of Behavioral, and Social Sciences Research that co-funded the National Institute on Drug Abuse grant, by the Knut and Alice Wallenberg Foundation under grant number 2015–0005, and the China Scholarship Council.

ACKNOWLEDGMENTS

The authors are grateful to Rana El Khoury and Brandon LeMay for their help in designing the setup and conducting some of the trials.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2019.00071/full#supplementary-material>

Video S1 | Exemplary video of individual training.

Video S2 | Exemplary video of social training.

REFERENCES

- Anthouard, M. (1987). A study of social transmission in juvenile *Dicentrarchus labrax* (Pisces, Serranidae), in an operant conditioning situation. *Behaviour* 103, 266–275. doi: 10.1163/156853987X00206
- Arthur, D., and Levin, E. D. (2001). Spatial and non-spatial visual discrimination learning in zebrafish (*Danio rerio*). *Anim. Cogn.* 4, 125–131. doi: 10.1007/s100710100111
- Bierbach, D., Landgraf, T., Romanczuk, P., Lukas, J., Nguyen, H., Wolf, M., et al. (2018a). Using a robotic fish to investigate individual differences in social responsiveness in the guppy. *R. Soc. Open Sci.* 5:181026. doi: 10.1098/rsos.181026
- Bierbach, D., Lukas, J., Bergmann, A., Elsner, K., Höhne, L., Weber, C., et al. (2018b). Insights into the social behavior of surface and cave-dwelling fish (*Poecilia mexicana*) in light and darkness through the use of a biomimetic robot. *Front. Robot. AI* 5:3. doi: 10.3389/frobt.2018.00003
- Bilotta, J., Risner, M. L., Davis, E. C., and Haggblom, S. J. (2005). Assessing appetitive choice discrimination learning in zebrafish. *Zebrafish* 2, 259–268. doi: 10.1089/zeb.2005.2.259
- Blanchard, R. J., McKittrick, C. R., and Blanchard, D. C. (2001). Animal models of social stress: effects on behavior and brain neurochemical systems. *Physiol. Behav.* 73, 261–271. doi: 10.1016/S0031-9384(01)00449-8
- Bouton, M. E. (2016). *Learning and Behavior: A Modern Synthesis*, 2 Edn. Sunderland, MA: Sinauer.
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* 7, 1–41. doi: 10.1007/s11721-012-0075-2
- Brown, C., and Laland, K. N. (2003). Social learning in fishes: a review. *Fish Fish.* 4, 280–288. doi: 10.1046/j.1467-2979.2003.00122.x
- Crollius, H. R., and Weissenbach, J. (2005). Fish genomics and biology. *Genome Res.* 15, 1675–1682. doi: 10.1101/gr.3735805
- Delgado, M. R., Olsson, A., and Phelps, E. A. (2006). Extending animal models of fear conditioning to humans. *Biol. Psychol.* 73, 39–48. doi: 10.1016/j.biopsycho.2006.01.006
- Dominik, C. (2010). *The Org-Mode 7 Reference Manual: Organize Your Life With GNU Emacs*. Bristol: Network Theory.
- Dowle, M., and Srinivasan, A. (2018). *data.table: Extension of 'data.frame'*. R package version 1.11.4. Available online at: <https://github.com/Rdatatable/data.table/wiki>
- Dugatkin, L. A., and Godin, J.-G. J. (1992). Reversal of female mate choice by copying in the guppy (*Poecilia reticulata*). *Proc. R. Soc. B Biol. Sci.* 249, 179–184. doi: 10.1098/rspb.1992.0101
- Dugatkin, L. A., and Godin, J.-G. J. (1993). Female mate copying in the guppy (*Poecilia reticulata*): age-dependent effects. *Behav. Ecol.* 4, 289–292. doi: 10.1093/beheco/4.4.289
- Engeszer, R. E., Patterson, L. B., Rao, A. A., and Parichy, D. M. (2007). Zebrafish in the wild: a review of natural history and new notes from the field. *Zebrafish* 4, 21–40. doi: 10.1089/zeb.2006.9997
- Fangmeier, M. L., Noble, D. W., O'Dea, R. E., Usui, T., Lagisz, M., Hesselson, D., et al. (2018). Computer animation technology in behavioral sciences: a sequential, automatic, and high-throughput approach to quantifying personality in zebrafish (*Danio rerio*). *Zebrafish* 15, 206–210. doi: 10.1089/zeb.2017.1532
- Fox, J., and Weisberg, S. (2011). *An R Companion to Applied Regression*, 2nd Edn. Thousand Oaks, CA: Sage.
- Fox, J., and Weisberg, S. (2018). Visualizing fit and lack of fit in complex regression models with predictor effect plots and partial residuals. *J. Stat. Softw.* 87, 1–27. doi: 10.18637/jss.v087.i09
- Frohnwieser, A., Murray, J. C., Pike, T. W., and Wilkinson, A. (2016). Using robots to understand animal cognition. *J. Exp. Anal. Behav.* 105, 14–22. doi: 10.1002/jeab.193
- Gerlai, R. (2011). “Associative learning in zebrafish (*Danio rerio*),” in *Methods in Cell Biology*, Vol 101, eds H. W. Detrich III, M. Westerfield, L. I. Zon (Waltham, MA: Elsevier), 249–270. doi: 10.1016/B978-0-12-387036-0.00012-8
- Gerlai, R., Fernandes, Y., and Pereira, T. (2009). Zebrafish (*Danio rerio*) responds to the animated image of a predator: towards the development of an automated aversive task. *Behav. Brain Res.* 201, 318–324. doi: 10.1016/j.bbr.2009.03.003
- Godin, J.-G. J., Herdman, E. J. E., and Dugatkin, L. A. (2005). Social influences on female mate choice in the guppy, *Poecilia reticulata*: generalized and repeatable trait-copying behaviour. *Anim. Behav.* 69, 999–1005. doi: 10.1016/j.anbehav.2004.07.016
- Grosenick, L., Clement, T. S., and Fernald, R. D. (2007). Fish can infer social rank by observation alone. *Nature* 445:429. doi: 10.1038/nature05511
- Hajage, D. (2009). *ascii: Export R objects to several markup languages*. Available online at: <http://eusebe.github.com/ascii/>
- Halloy, J., Mondada, F., Kernbach, S., and Schmickl, T. (2013). “Towards bio-hybrid systems made of social animals and robots,” in *Conference on Biomimetic and Biohybrid Systems* (London: Springer), 384–386.
- Halloy, J., Sempo, G., Caprari, G., Rivault, C., Asadpour, M., Tâche, F., et al. (2007). Social integration of robots into groups of cockroaches to control self-organized choices. *Science* 318, 1155–1158. doi: 10.1126/science.1144259
- Heyes, C. (2001). Causes and consequences of imitation. *Trends Cogn. Sci.* 5, 253–261. doi: 10.1016/S1364-6613(00)01661-2
- Heyes, C. (2012a). Simple minds: a qualified defence of associative learning. *Philos. Trans. R. Soc. B Biol. Sci.* 367, 2695–2703. doi: 10.1098/rstb.2012.0217
- Heyes, C. (2012b). What's social about social learning? *J. Compar. Psychol.* 126, 193–202. doi: 10.1037/a0025180
- Hicks, C., Sorocco, D., and Levin, M. (2006). Automated analysis of behavior: a computer-controlled system for drug screening and the investigation of learning. *J. Neurobiol.* 66, 977–990. doi: 10.1002/neu.20290
- Hoppitt, W., and Laland, K. N. (2013). *Social Learning: An Introduction to Mechanisms, Methods, and Models*. Princeton, NJ: Princeton University Press. doi: 10.1515/9781400846504
- Kim, C., Ruberto, T., Phamduy, P., and Porfiri, M. (2018). Closed-loop control of zebrafish behaviour in three dimensions using a robotic stimulus. *Sci. Rep.* 8:657. doi: 10.1038/s41598-017-19083-2
- Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends Biotechnol.* 31, 287–294. doi: 10.1016/j.tibtech.2013.03.002
- Krause, J., Winfield, A. F., and Deneubourg, J.-L. (2011). Interactive robots in experimental biology. *Trends. Ecol. Evol.* 26, 369–375. doi: 10.1016/j.tree.2011.03.015
- Kubinyi, E., Miklósi, Á., Kaplan, F., Gácsi, M., Topál, J., and Csányi, V. (2004). Social behaviour of dogs encountering aibo, an animal-like robot in a neutral and in a feeding situation. *Behav. Process.* 65, 231–239. doi: 10.1016/j.beproc.2003.10.003
- Laland, K. N., and Williams, K. (1997). Shoaling generates social learning of foraging information in guppies. *Anim. Behav.* 53, 1161–1169.
- Laland, K. N., and Williams, K. (1998). Social transmission of maladaptive information in the guppy. *Behav. Ecol.* 9, 493–499.
- Landgraf, T., Bierbach, D., Kirbach, A., Cusing, R., Oertel, M., Lehmann, K., et al. (2018). Dancing honey bee robot elicits dance-following and recruits foragers. *arXiv arXiv:1803.07126*.
- Landgraf, T., Bierbach, D., Nguyen, H., Muggelberg, N., Romanczuk, P., and Krause, J. (2016). Robofish: increased acceptance of interactive robotic fish with realistic eyes and natural motion patterns by live trinidadian guppies. *Bioinspir. Biomimet.* 11:015001. doi: 10.1088/1748-3190/11/1/015001
- Leadbeater, E., and Chittka, L. (2007). Social learning in insects - from miniature brains to consensus building. *Curr. Biol. Philos.* 17, 703–713. doi: 10.1016/j.cub.2007.06.012
- Lind, J., Ghirlanda, S., and Enquist, M. (2019). Social learning through associative processes: a computational theory. *R. Soc. Open Sci.* 6:181777. doi: 10.1098/rsos.181777
- Lindeyer, C. M., and Reader, S. M. (2010). Social learning of escape routes in zebrafish and the stability of behavioural traditions. *Anim. Behav.* 79, 827–834. doi: 10.1016/j.anbehav.2009.12.024
- Macrì, S., Neri, D., Ruberto, T., Mwaffo, V., Butail, S., and Porfiri, M. (2017). Three-dimensional scoring of zebrafish behavior unveils biological phenomena hidden by two-dimensional analyses. *Sci. Rep.* 7:1962. doi: 10.1038/s41598-017-01990-z
- Markram, K., Rinaldi, T., La Mendola, D., Sandi, C., and Markram, H. (2008). Abnormal fear conditioning and amygdala processing in an animal model of autism. *Neuropsychopharmacology* 33:901. doi: 10.1038/sj.npp.1301453
- Mineka, S., and Zinbarg, R. (2006). A contemporary learning theory perspective on the etiology of anxiety disorders: it's not what you thought it was. *Am. Psychol.* 61:10. doi: 10.1037/0003-066X.61.1.10

- Morin, C., de Souza Silva, M. A., Müller, C. P., Hardigan, P., and Spieler, R. E. (2013). Active avoidance learning in zebrafish (*Danio rerio*)—the role of sensory modality and inter-stimulus interval. *Behav. Brain Res.* 248, 141–143. doi: 10.1016/j.bbr.2013.04.009
- Mwaffo, V., Anderson, R. P., Butail, S., and Porfiri, M. (2015). A jump persistent turning walker to model zebrafish locomotion. *J. R. Soc. Interf.* 12:20140884. doi: 10.1098/rsif.2014.0884
- Mwaffo, V., Butail, S., and Porfiri, M. (2017). In-silico experiments of zebrafish behaviour: modeling swimming in three dimensions. *Sci. Rep.* 7:39877. doi: 10.1038/srep39877
- Norton, W., and Bally-Cuif, L. (2010). Adult zebrafish as a model organism for behavioural genetics. *BMC Neurosci.* 11:90. doi: 10.1186/1471-2202-11-90
- Partan, S. R., Larco, C. P., and Owens, M. J. (2009). Wild tree squirrels respond with multisensory enhancement to conspecific robot alarm behaviour. *Anim. Behav.* 77, 1127–1135. doi: 10.1016/j.anbehav.2008.12.029
- Pather, S., and Gerlai, R. (2009). Shuttle box learning in zebrafish (*Danio rerio*). *Behav. Brain Res.* 196, 323–327. doi: 10.1016/j.bbr.2008.09.013
- Pearce, J. M. (2008). *Animal Learning and Cognition*, 3 Edn. Hove: Psychology Press.
- Porfiri, M. (2018). Inferring causal relationships in zebrafish-robot interactions through transfer entropy: a small lure to catch a big fish. *Anim. Behav. Cogn.* 5, 341–367. doi: 10.26451/abc.05.04.03.2018
- R Core Team (2018). *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.
- Reebs, S. G. (2000). Can a minority of informed leaders determine the foraging movements of a fish shoal? *Anim. Behav.* 59, 403–409. doi: 10.1006/anbe.1999.1314
- Romano, D., Benelli, G., Donati, E., Remorini, D., Canale, A., and Stefanini, C. (2017). Multiple cues produced by a robotic fish modulate aggressive behaviour in siamese fighting fishes. *Sci. Rep.* 7:4667. doi: 10.1038/s41598-017-04840-0
- Romano, D., Benelli, G., Hwang, J.-S., and Stefanini, C. (2019a). Fighting fish love robots: mate discrimination in males of a highly territorial fish by using female-mimicking robotic cues. *Hydrobiologia* 833, 185–196. doi: 10.1007/s10750-019-3899-6
- Romano, D., Benelli, G., and Stefanini, C. (2019b). Encoding lateralization of jump kinematics and eye use in a locust via bio-robotic artifacts. *J. Exp. Biol.* 222:jeb187427. doi: 10.1242/jeb.187427
- Romano, D., Donati, E., Benelli, G., and Stefanini, C. (2018). A review on animal-robot interaction: from bio-hybrid organisms to mixed societies. *Biol. Cybernet.* 113, 201–225. doi: 10.1007/s00422-018-0787-5
- Ruberto, T., Mwaffo, V., Singh, S., Neri, D., and Porfiri, M. (2016). Zebrafish response to a robotic replica in three dimensions. *R. Soc. Open Sci.* 3:160505. doi: 10.1098/rsos.160505
- Ruberto, T., Polverino, G., and Porfiri, M. (2017). How different is a 3d-printed replica from a conspecific in the eyes of a zebrafish? *J. Exp. Anal. Behav.* 107, 279–293. doi: 10.1002/jeab.247
- Schlupp, I., and Ryan, M. J. (1997). Male sailfin mollies (*Poecilia latipinna*) copy the mate choice of other males. *Behav. Ecol.* 8, 104–107.
- Schneider, T., and Przewlocki, R. (2005). Behavioral alterations in rats prenatally exposed to valproic acid: animal model of autism. *Neuropsychopharmacology* 30:80. doi: 10.1038/sj.npp.1300518
- Schulte, E., Davison, D., Dye, T., and Dominik, C. (2012). A multi-language computing environment for literate programming and reproducible research. *J. Stat. Softw.* 46, 1–24. doi: 10.18637/jss.v046.i03
- Sison, M., and Gerlai, R. (2010). Associative learning in zebrafish (*Danio rerio*) in the plus maze. *Behav. Brain Res.* 207, 99–104. doi: 10.1016/j.bbr.2009.09.043
- Stewart, A., Gaikwad, S., Kyzar, E., Green, J., Roth, A., and Kalueff, A. V. (2012). Modeling anxiety using adult zebrafish: a conceptual review. *Neuropharmacology* 62, 135–143. doi: 10.1016/j.neuropharm.2011.07.037
- Takanishi, A., Aoki, T., Ito, M., Ohkawa, Y., and Yamaguchi, J. (1998). “Interaction between creature and robot: development of an experiment system for rat and rat robot interaction,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (Victoria, BC: IEEE), 1975–1980. doi: 10.1109/IROS.1998.724896
- Valentini, G., Brambilla, D., Hamann, H., and Dorigo, M. (2016). “Collective perception of environmental features in a robot swarm,” in *Swarm Intelligence*, eds M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, et al. (Brussels: Springer International Publishing), Lecture Notes in Computer Science, 65–76. doi: 10.1007/978-3-319-44427-7-6
- van Schaik, C. (2010). “Social learning and culture in animals,” in *Animal Behaviour: Evolution and Mechanisms*, ed P. Kappeler (Berlin; Heidelberg: Springer), 623–653.
- Vascotto, S. G., Beckham, Y., and Kelly, G. M. (1997). The zebrafish's swim to fame as an experimental model in biology. *Biochem. Cell Biol.* 75, 479–485.
- Webb, B. (2000). What does robotics offer animal behaviour? *Anim. Behav.* 60, 545–558. doi: 10.1006/anbe.2000.1514
- Webster, M., and Laland, K. (2015). Public information use by foraging ninespine sticklebacks: Social learning or an unlearned social influence on travel direction? *Behaviour* 152, 1569–1584. doi: 10.1163/1568539X-00003293
- Webster, M. M., and Laland, K. N. (2008). Social learning strategies and predation risk: minnows copy only when using private information would be costly. *Proc. R. Soc. Lond. B Biol. Sci.* 275, 2869–2876. doi: 10.1098/rspb.2008.0817
- Webster, M. M., and Laland, K. N. (2017). Social information use and social learning in non-grouping fishes. *Behav. Ecol.* 28, 1547–1552. doi: 10.1093/beheco/arx121
- Wickham, H., and Bryan, J. (2018). *readxl: Read Excel Files*. R package version 1.1.0. Available online at: <https://readxl.tidyverse.org/>
- Witte, K., and Ueding, K. (2003). Sailfin molly females (*Poecilia latipinna*) copy the rejection of a male. *Behav. Ecol.* 14, 389–395. doi: 10.1093/beheco/14.3.389
- Xu, X., Scott-Scheiern, T., Kempker, L., and Simons, K. (2007). Active avoidance conditioning in zebrafish (*Danio rerio*). *Neurobiol. Learn. Memory* 87, 72–77. doi: 10.1016/j.nlm.2006.06.002
- Yang, Y., LeMay, B., El Khoury, R., Clément, R. J., Ghirlanda, S., and Porfiri, M. (2019). “Can robotic fish help zebrafish learn to open doors?” in *Bioinspiration, Biomimetics, and Bioreplication IX*, vol 10965, eds R. J. Martín-Palma, M. Knez, A. Lakhtakia (Denver, CO: International Society for Optics and Photonics), 109650B.
- Zentall, T. R., and Galef, B. G. J., editors (1988). *Social Learning: Psychological and Biological Perspectives*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Zienkiewicz, A. K., Ladu, F., Barton, D. A., Porfiri, M., and Di Bernardo, M. (2018). Data-driven modelling of social forces and collective behaviour in zebrafish. *J. Theor. Biol.* 443, 39–51. doi: 10.1016/j.jtbi.2018.01.011

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2019 Yang, Clément, Ghirlanda and Porfiri. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



A Survey on Swarming With Micro Air Vehicles: Fundamental Challenges and Constraints

Mario Coppola^{1,2*}, Kimberly N. McGuire¹, Christophe De Wagter¹ and Guido C. H. E. de Croon^{1*}

¹ Micro Air Vehicle Laboratory (MAVLab), Department of Control and Simulation, Faculty of Aerospace Engineering, Delft University of Technology, Delft, Netherlands, ² Department of Space Systems Engineering, Faculty of Aerospace Engineering, Delft University of Technology, Delft, Netherlands

OPEN ACCESS

Edited by:

Eliseo Ferrante,
Vrije Universiteit Amsterdam,
Netherlands

Reviewed by:

Titus Cieslewski,
University of Zurich, Switzerland
Tomas Krajnik,
Czech Technical University in Prague,
Czechia

*Correspondence:

Mario Coppola
m.coppola@tudelft.nl
Guido C. H. E. de Croon
g.c.h.e.decroon@tudelft.nl

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 18 November 2019

Accepted: 04 February 2020

Published: 25 February 2020

Citation:

Coppola M, McGuire KN, De Wagter C and de Croon GCH (2020) A Survey on Swarming With Micro Air Vehicles: Fundamental Challenges and Constraints. *Front. Robot. AI* 7:18. doi: 10.3389/frobt.2020.00018

This work presents a review and discussion of the challenges that must be solved in order to successfully develop swarms of Micro Air Vehicles (MAVs) for real world operations. From the discussion, we extract constraints and links that relate the local level MAV capabilities to the global operations of the swarm. These should be taken into account when designing swarm behaviors in order to maximize the utility of the group. At the lowest level, each MAV should operate safely. Robustness is often hailed as a pillar of swarm robotics, and a minimum level of local reliability is needed for it to propagate to the global level. An MAV must be capable of autonomous navigation within an environment with sufficient trustworthiness before the system can be scaled up. Once the operations of the single MAV are sufficiently secured for a task, the subsequent challenge is to allow the MAVs to sense one another within a neighborhood of interest. Relative localization of neighbors is a fundamental part of self-organizing robotic systems, enabling behaviors ranging from basic relative collision avoidance to higher level coordination. This ability, at times taken for granted, also must be sufficiently reliable. Moreover, herein lies a constraint: the design choice of the relative localization sensor has a direct link to the behaviors that the swarm can (and should) perform. Vision-based systems, for instance, force MAVs to fly within the field of view of their camera. Range or communication-based solutions, alternatively, provide omni-directional relative localization, yet can be victim to unobservable conditions under certain flight behaviors, such as parallel flight, and require constant relative excitation. At the swarm level, the final outcome is thus intrinsically influenced by the on-board abilities and sensors of the individual. The real-world behavior and operations of an MAV swarm intrinsically follow in a bottom-up fashion as a result of the local level limitations in cognition, relative knowledge, communication, power, and safety. Taking these local limitations into account when designing a global swarm behavior is key in order to take full advantage of the system, enabling local limitations to become true strengths of the swarm.

Keywords: swarm, challenges, review, robustness, autonomous, micro air vehicles, drones, MAV

1. INTRODUCTION

Micro Air Vehicles (MAVs), or “small drones,” are becoming commonplace in the modern world. The term refers to small, light-weight, flying robots. Several MAV designs exist, including multirotors (Kumar and Michael, 2012), flapping wing (Michelson and Reece, 1998; Wood et al., 2013; de Croon et al., 2016), fixed wing (Green and Oh, 2006), morphing designs (Falanga et al., 2019b), or “hybrid” vehicles (Itasse et al., 2011). Of these, quadrotors have enjoyed the spotlight due to their high maneuverability, their ability to take-off vertically (as opposed to most fixed wing MAVs, for instance), and their relative simplicity in design (Gupte et al., 2012; Kumar and Michael, 2012). MAVs can be used for surveillance and mapping (Mohr and Fitzpatrick, 2008; Scaramuzza et al., 2014; Saska et al., 2016b), infrastructure inspection (Sa and Corke, 2014), load transport and delivery (Palunko et al., 2012), or construction (Lindsey et al., 2012; Augugliaro et al., 2014). Such applications are particularly useful in areas that are not easily accessible by humans, like forests or disaster sites (Alexis et al., 2009; Achtelik et al., 2012). Smaller and lighter designs push the boundaries of their applications further. Aside from the asset of increased portability, smaller MAVs can also navigate through tighter spaces, such as narrow indoor environments with higher agility (Mohr and Fitzpatrick, 2008). They also cause less damage to their surroundings (including people) in the event of a collision, making them intrinsically safer tools (Kushleyev et al., 2013).

Unfortunately, smaller size comes at the expense of more limited capabilities. The interplay between limited flight time, limited sensing, and limited power hinder an MAV from performing grander tasks on its own. This has created a strong interest in developing MAV *swarms* (Yang et al., 2018). The paradigm of swarm robotics aims to transcend the limitations of a single robot by enabling cooperation in larger teams. This is inspired by the animal kingdom, where animals and insects have been observed to unite forces toward a common goal that is otherwise too complex or challenging for the lone individual (Garnier et al., 2007). Using several robots at once can bring several advantages and possibilities, such as: redundancy, faster task completion due to parallelization, or the execution of collaborative tasks (Martinoli and Easton, 2003; Trianni and Campo, 2015; Nedjah and Junior, 2019). The control of robotic swarms is envisioned to be fully distributed. The individual robots perceive and process their environment locally and then act accordingly without global awareness or direct awareness of the final goal of the swarm. Nevertheless, by means of collaboration, the robots can achieve an objective that they would not have been able to achieve by themselves. As they say: there is strength in numbers.

It is easy to imagine swarms of MAVs jointly carrying a load that is too heavy for a single one to lift, or persistently exploring an area without interruption. As is often the case, however, putting such visions into practice is another story altogether. Developing self-organizing swarms of MAVs in the real world is a multi-disciplinary challenge coarsely divided in two main aspects. One aspect is that of the individual MAV design, where the local abilities of a single MAV are defined. The second aspect is the swarm design, whereby we need to develop controllers with

which the global goal can be efficiently achieved, autonomously, by the swarm. To make matters more complicated, the two are not decoupled. As we shall explore in this paper, there exist fundamental links between the local limitations of an MAV and the behaviors that a swarm of MAVs could, or should, execute as a result. Vice versa, in order to realize certain swarm behaviors, there are local requirements that the individual MAVs must meet. This bond between the local and the global cannot be ignored if MAV swarms are to be brought to the real world. In this paper, we aim to reconcile these two aspects and present a discussion of the fundamental challenges and constraints linking local MAV properties and global swarm behaviors.

2. CO-DEPENDENCE OF SWARM DESIGN AND INDIVIDUAL MAV DESIGN

Let us begin from the primary challenge of swarm robotics: to design local controllers that successfully lead to global swarm behaviors (Şahin et al., 2008). Concerning MAVs, these global behaviors include, but are not limited to: collaborative transport, collaborative construction, distributed sensing, collaborative object manipulation, and parallelized exploration and mapping of environments. Albeit the individual MAV may be limited in its ability to successfully perform these tasks (for instance, as areas get larger or loads get heavier), they can be tackled by collaborating in a swarm. Generally, swarms of robots are expected to feature the following inherent advantages (Şahin et al., 2008; Brambilla et al., 2013):

- **Robustness:** The swarm is robust to the loss or failure of individual robots.
- **Flexibility:** The swarm can reconfigure to tackle different tasks.
- **Scalability:** The swarm can grow and shrink in size depending on the needs of the global task.

When designing a swarm of MAVs, we must then ask ourselves: how can we design a swarm that is robust, flexible, and scalable? It is true that these properties pertain to the swarm rather than the individual, but if the swarm is composed of individual units, then it follows that they must also be present (although perhaps not always apparent) at the local level. We cannot use individual robots that are not robust and merely expect the swarm as a whole to be immune or tolerant to individual failures (Bjerknes and Winfield, 2013). If there is a high probability of errors at the local level, such as erroneous observations, poorly executed commands, or failure of a unit, then this may have a repercussion on the swarm's performance; an effect that Bjerknes and Winfield (2013) have shown can worsen with the number of robots in a swarm. There is a point after which the individual robots are too unreliable and the swarm can fail to achieve its goal, or it can be shown to be outperformed by smaller teams with more reliable units (Stancliff et al., 2006) or even by a single reliable system (Engelen et al., 2014). The further complication with MAVs is that local failures do not remain local, but are likely to cause collisions and damages to other nearby MAVs and/or objects. For some tasks, such as collective transport, the impact may be even more severe as the MAVs are mechanically attached to the

load (Tagliabue et al., 2019). It thus follows that, to develop a robust swarm for real world deployment, we must also ensure robustness at the local level.

Of equal importance is to make sure that the robots have the satisfactory tools and sensors to carry out their individual components of a global task. The more capable the sensors are, the more likely it is that the swarm can be flexible and adjust to different tasks or unexpected changes. When performing pure swarm intelligence research, we can afford to abstract away from lower level issues (Brutschy et al., 2015). For instance, in a study on making a decision about selecting a new location for a swarm's nest, one can abstract away from actually evaluating the quality of a nest location, and instead focus the analysis on a particular aspect of the system, such as the decision making process. However, when dealing with real-world applications, this is not an option. If we want to develop nest selection capabilities for a swarm in the real world, each robot should be capable of: flying and operating safely, recognizing the existence of a site, evaluating the quality of a site with a certain reliability, exchanging this information with its neighbors, and more. All these lower level requirements need to be appropriately realized for the global level outcome to emerge, or otherwise need to be accepted as limitations of the system. The way in which they are implemented shape the final behavior of the swarm.

Last but not least, unless properly accounted for, there are scalability problems that may also occur as the swarm grows in size. Examples of issues are: a congested airspace whereby the MAVs are unable to adhere to safety distances, a cluttered visual environment as a result of several MAVs (thus obstructing the task), or poor connectivity as a result of low-range communication capabilities. To achieve scalability, the MAV design must be such that these properties are appropriately accommodated, from the appropriate hardware design all the way to the higher level controllers which make up the swarm behavior.

2.1. The Challenge of Local Sensing and Control

When flying several MAVs at once, the control architecture can be of two types: (1) centralized, or (2) decentralized. In the centralized case, all MAVs in a swarm are controlled by a single computer. This “omniscient” entity knows the relevant states of all MAVs and can (pre-)plan their actions accordingly. The planning can be done a-priori and/or online. In the decentralized case, the MAVs make their decisions locally.

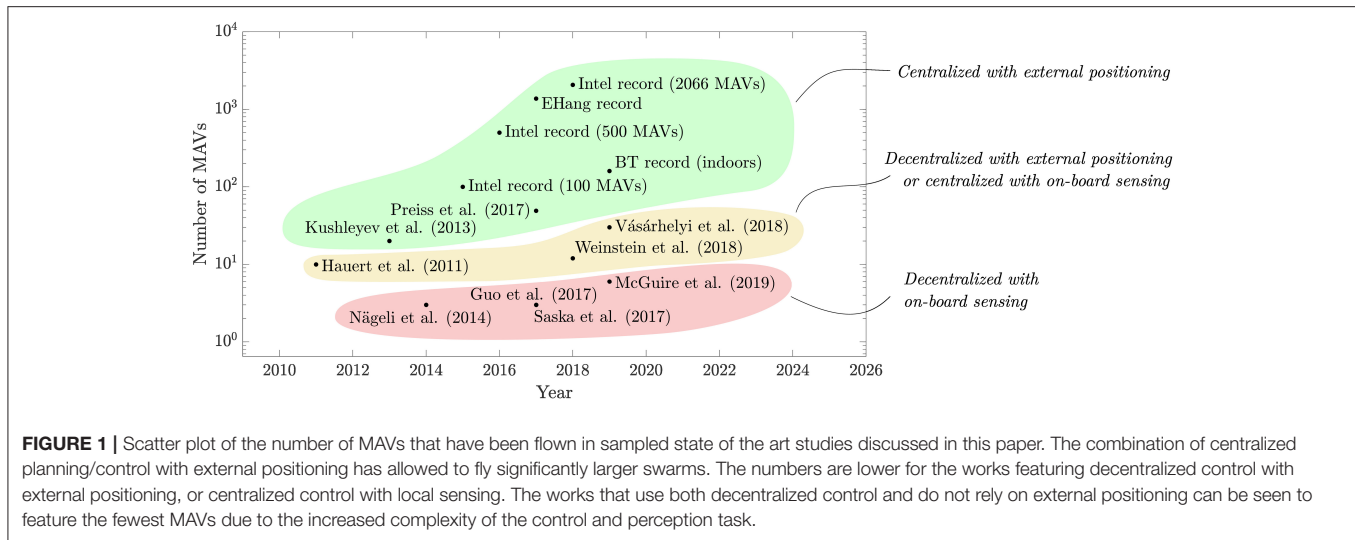
A second dichotomy can also be defined for how the MAVs sense their environment: (1) using external position sensing, or (2) locally. External positioning is typically achieved with a Global Navigation Satellite System (GNSS) or with a Motion Capture System (MCS), depending on whether the MAVs are flying outdoors or indoors, respectively. Alternatively, the latter only relies on the sensors that are on-board of the MAV.

Currently, the combination of centralized architecture and external positioning have achieved the highest stage of maturity, allowing for flights with several MAVs. Kushleyev et al. (2013)

showed a swarm of 20 micro quadrotors that could re-organize in several formations. Lindsey et al. (2012), Augugliaro et al. (2014), and Mirjan et al. (2016) developed impressive collaborative construction schemes using a team of MAVs. Preiss et al. (2017) showcased “Crazyswarm,” an indoor display of 49 small quadrotors flying together. The strategy of centralized planning and external positioning has also attracted large industry investments, leading to shows with record-breaking number of MAVs flying simultaneously. In 2015, Intel and Ars Electronica Futurelab first flew 100 MAVs, making a Guinness World Record (Swatman, 2016a). In 2016, Intel beat its own record by flying 500 MAVs simultaneously (Swatman, 2016b). In 2018, EHang claimed the record with 1,374 MAVs flying above the city of Xi'an, China (Cadell, 2018). In 2019, Intel reclaimed the title by flying 2,066 MAVs (Guinness World Records, 2018) outdoors. Meanwhile, the record for the most MAVs flying indoors (from a single computer) was recently broken by BT with 160 MAVs (Guinness World Records, 2019).

Without external positioning systems or centralized planning/control, the problem of flying several MAVs at once becomes more challenging. This is because: (1) the MAVs have to rely only on on-board perception, or (2) they have to make local decisions without the benefit of global planning, or (3) both. It is then not surprising that, as shown in **Figure 1**, the swarms that have been flown without external positioning and/or centralized control are significantly smaller. When the control is decentralized, but the MAVs benefit from an external positioning system, or vice versa, the largest swarms are in the dozens (Hauert et al., 2011; Vásárhelyi et al., 2018; Weinstein et al., 2018). For swarms featuring both local perception and distributed control, the highest numbers are currently in the single digits (Nägeli et al., 2014; Guo et al., 2017; Saska et al., 2017; McGuire et al., 2019). Despite the fact that these numbers have been increasing in the last few years, they are still lower, as the operations are shifted away from external system and toward on-board perception and control. If the past is any indication for the future, we expect that: (1) the numbers of drones will keep increasing for all cases, and (2) businesses will take over the records as the technologies for on-board decision making and perception become more mature.

Although we can fly a high number of MAVs when using centralized planning and external positioning, swarming is not just a numbers game. Flying with many MAVs does not automatically imply that we are achieving the benefits of swarm robotics (Hamann, 2018). A centralized system relies on a main computer to take all decisions. This means that a prompt online re-planning is needed in order to achieve robustness and flexibility. This re-planning grows in complexity with the size of the swarm, making the system unscalable. Moreover, the central computer represents a single point of failure. Instead, a swarm adopts a distributed strategy whereby each robot takes a decision independently. The fact that each MAV needs to take its own decisions, and additionally, if the MAVs do not rely on external infrastructure, introduces a new layer of difficulty. However, this is also what brings new advantages: redundancy, scalability, and



adaptability to changes (Şahin, 2005; Bonabeau and Théraulaz, 2008)¹.

When we analyze swarms of MAVs with local on-board sensing and control, we can observe two trends: (1) As the size of the swarm increases, the *relative* knowledge that each MAV will have of its global environment, which includes the remainder of the swarm, decreases (Bouffanais, 2016); (2) as the individual MAV's size and/or mass decreases, its capability to sense its own local environment decreases (Kumar and Michael, 2012). This creates an interesting challenge. On the one hand, we aim to design smaller, lighter, cheaper, and more efficient MAVs. On the other hand, as we make these MAVs smaller, the gap between the microscopic and macroscopic widens further. Designing the swarm becomes a more challenging task because each MAV has less information about its environment and is also less capable to act on it. This can be generalized to other robotic platforms as well, but MAVs feature the increased difficulty of having a tightly bound relationship between their on-board capabilities, their dynamics, their processing power, and their sensing (Chung et al., 2018). This is sometimes referred to as the SWaP (Size, Weight, and Power) trade-off (Mahony et al., 2012; Liu et al., 2018). The relationship is often non-linear. For instance, if we add a sensor that results in 5% more power usage, it does not only spend more energy per second, but it also affects the total energy that can be extracted from the battery as it will be operating in a different regime (de Croon et al., 2016). For many MAVs, grams and milliwatts matter. This makes the design of autonomous decentralized swarms of MAVs a more unique challenge.

2.2. Overview of Design Challenges Throughout the Design Chain

Throughout this paper, we shall review the state of the art in MAV technology from the swarm robotics perspective. To facilitate

our discussion, we will break down the challenges for the design and control of an MAV swarm in the following four levels, from “local” to “global.”

1. MAV design. This defines the processing power, flight time, dynamics, and capabilities of the single MAV. Most importantly from a swarm engineering perspective, it defines the sensory information available on each unit, from which it can establish its view of the world. This is discussed in section 3.
2. Local ego-state estimation and control. At the lowest level, an MAV must be capable of controlling its motion with sufficient accuracy. This lower level layer handles basic flight operations of the MAV. This includes attitude control, height control, and velocity estimation and control. Moreover, the MAV should be capable of safely navigating in its environment. Minimally, it should detect and avoid potential obstacles. The challenges and state of the art for these methods are discussed in section 4.
3. Intra-swarm relative sensing and avoidance. There are two key enabling technologies for swarming. The first is the knowledge on (the location of) nearby neighbors. This is particularly important for MAV swarms as it not only enables several higher level swarming behaviors, but it also ensures that MAVs do not collide with one another in mid-air. The second enabling technology is communication between MAVs, such that they can share information and thus expand their knowledge of the environment via their neighborhood. These are discussed in section 5.
4. Swarm behavior. This is the higher level control policy that the robots follow to generate the global swarm behavior. Examples of higher level controllers in swarms range from attraction and repulsion forces for flocking (Gazi and Passino, 2002; Vászárhelyi et al., 2018) to neural networks for aggregation, dispersion, or homing (Duarte et al., 2016). We discuss how MAV swarm behaviors can be designed in section 6.

Other similar taxonomies have been defined. Floreano and Wood (2015) describe three levels of robotic cognition:

¹Of course, flying several MAVs with a centralized controller has its own challenges, which we do not mean to undermine. We only mean that it appears that these methods are at a more mature stage with respect to self-organized approaches, which is the focus of this article.

sensory-motor autonomy, reactive autonomy, and cognitive autonomy. Meanwhile, de Croon et al. (2016) divide the control process for autonomous flight into four levels: attitude control, height control, collision avoidance, and navigation. Although the taxonomies above are conceptually similar (generally going from low level sensing and control to a higher level of cognition), the re-definition that we provide here is designed to better organize our discussion within the context of swarm robotics. Moreover, we also include the design of the MAV within the chain. As we will explain in this manuscript, this has a fundamental impact on the higher level layers.

The four stages that have been defined have an increasing level of abstraction. The lower levels enable the robustness, flexibility, and scalability properties expected at the higher level, while the higher levels dictate, accommodate, and make the most out of the capabilities set at the lower level. From a more systems perspective, the MAV design poses constraints on what the higher level controllers can expect to achieve, while the higher level controllers create requirements that the MAV must be able to fulfill. A simplified view of the flow of requirements and constraints is shown in **Figure 2**.

Throughout the remainder of this paper, as we discuss the state of the art at each level, we will highlight the major constraints that flow upwards and the requirements that flow downwards. Naturally, each sub-topic that we will treat features a plethora of solutions, challenges, and methods, each deserving of a review paper of its own. It is beyond the scope (and probably far beyond any acceptable word limit, too) to present an exhaustive review about each topic. Instead, we keep our focus to highlighting the main methodologies and how they can be used to design swarms of MAVs. Where possible, we will refer the reader to more in-depth reviews on a specific topic.

3. MAV DESIGN

The differentiating challenge faced by a flying robot, namely (and somewhat trivially) the fact that it has to carry its own mass around, creates a strong design driver toward minimalism. Despite battery mass consisting of up to 20–30% of the total system mass, the flight time of quadrotor MAVs still remains limited to the order of magnitude of minutes (Kumar and Michael, 2012; Mulgaonkar et al., 2014; Oleynikova et al., 2015). To increase the carrying capabilities of an MAV, enabling it to carry more/better sensors, processors, or actuators, while keeping flight time constant, means that the size of the battery should also increase. In turn, this leads to a new increase in mass, and so on. This type of spiral, often referred to as the “snowball effect,” is a well-known issue for the design of any flying vehicle, from MAVs to trans-Atlantic airliners (Obert, 2009; Lammering et al., 2012; Voskuil et al., 2018). It then becomes paramount for an MAV design to be as minimalist as possible relative to its task, such that it may fulfill the mission requirements with a minimum mass (or, at the very least, there is a trade-off to be considered). This design driver has been taken to the extreme and has led to the development of miniature MAV systems, popular examples of which include the Ladybird drone and the Crazyfly (Lehnert

and Corke, 2013; Remes et al., 2014; Giernacki et al., 2017). These MAVs have a mass of <50 g, making them attractive due to their low cost and the fact that they are safer to operate around people. This makes them appealing for swarming, especially in indoor environments (Preiss et al., 2017).

A substantial body of literature already exists on single MAV design, the specifics of which largely vary depending on the type of MAV in question. We refer the reader to the works of Mulgaonkar et al. (2014) and Floreano and Wood (2015) and the sources therein for more details. From the swarming perspective, it is important to understand that, independently of the type of MAV in question, the following constraints are intertwined during the design phase: (1) flight time, (2) on-board sensing, (3) on-board processing power, and (4) dynamics. This means that the choice of MAV directly constrains the application as well as the swarming behavior that can be achieved (or, vice versa, a desired swarming behavior requires a specific type of MAV). For example, fixed wing MAVs benefit from longer autonomy. This makes them ideal candidates for long term operations, and also give the operators more time to launch an entire fleet and replace members with low batteries (Chung et al., 2016). However, fixed wing MAVs also have limited agility in comparison to quadrotors or flapping wing MAVs. The latter, for instance, can have a very high agility (Karásek et al., 2018), but also comes with more limited endurance and payload constraints (Olejník et al., 2019). The MAV design impacts the number and type of sensors that can be taken on-board. It can also impact how these sensors are positioned and their eventual disturbances and noise. In turn, this affects the local sensing and control properties of the MAV and can also impact its ability to sense neighbors and operate in a team more effectively. We will return to this where relevant in the next chapters, whereby we discuss how an MAV can estimate and control its motion, sense its neighbors, and navigate in an environment together with the rest of the swarm.

A special note is made to designs that are *intended* for collaboration. Oung and D’Andrea (2011) introduced the Distributed Flight Array, a design whereby multiple single rotors can attach and detach from each other to form larger multi-rotors. More recently, Saldaña et al. (2018) introduced the ModQuad: a quadrotor with a magnetic frame designed for self-assembly with its neighbors. This design provides a solution for collaborative transport by creating a more powerful rigid structure with several drones. Gabrich et al. (2018) have shown how the ModQuad design can be used to form an aerial gripper. Because of the frame design, one of the difficulties of the ModQuad was in the disassembly back to individual quadrotors. This was tackled with a new frame design which enabled the quadrotors to disassemble by moving away from each other with a sufficiently high roll/pitch angle (Saldaña et al., 2019).

4. LOCAL EGO-STATE ESTIMATION AND CONTROL

The primary objective for a single MAV operating in a swarm is to remain in flight and perform higher level tasks with a

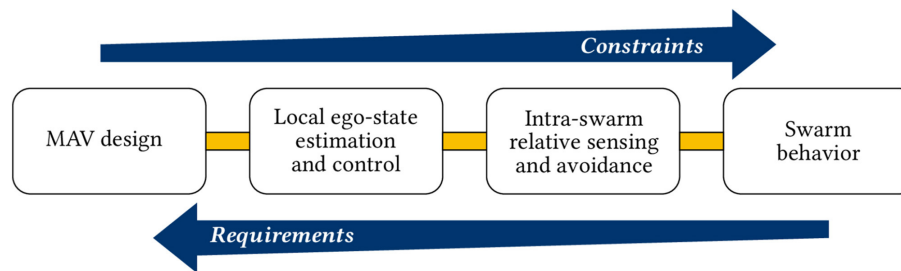


FIGURE 2 | Generalized depiction of the flow of requirements and constraints for the design of MAV swarms. The lower level design choices create constraints on the higher level properties of the swarm. Higher level design choices create requirements for the lower levels, down to the physical design of the MAV. Note that, for the specific case, the flow of requirements and constraints is likely to be more intricate than this picture makes it out to be. However, the general idea remains.

given accuracy. This requires a robust estimation of the on-board state as well as robust lower level control, preferably while minimizing the size, power, and processing required. The design choices made here dictate the accuracy (i.e., noise, bias, and disturbances) with which each MAV will know its own state, as well as which variables the state is actually comprised of. In turn, this affects the type of maneuvers and actions that an MAV can execute. For instance, aggressive flight maneuvers likely require relatively accurate real-time state estimation (Bry et al., 2015). Of equal importance are the considerations for the processing power that remains for higher level tasks. While it can be attractive to implement increasingly advanced algorithms to achieve a more reliable ego-state estimate, these can be too computationally expensive to run on-board even by modern standards (Ghadiok et al., 2012; Schauwecker and Zell, 2014). This limits the MAV, as processing power is diverted from tasks at a higher level of cognition. If not properly handled, it can lead to sub-optimal final performances by the MAVs and by the swarm².

4.1. Low-Level State Estimation and Control

This section outlines the main sensors and methods that can be used by MAVs to measure their on-board states, laying the foundations for our swarm-focused discussion in later sections. We organize the discussion by focusing on the following parameters: attitude (section 4.1.1), velocity and odometry (section 4.1.2), and height and altitude (section 4.1.3). Moreover, we restrict our overview to *on-board* sensing, as this is in line with the swarming philosophy and the relevant applications.

4.1.1. Attitude

It is essential for an MAV to estimate and control its own attitude in order to control its flight (Beard, 2007; Bouabdallah and Siegwart, 2007). Accelerations and angular rotation rates are typically measured through the on-board Inertial Measurement Unit (IMU) sensor (Bouabdallah et al., 2004; Gupte et al., 2012). The IMU measurements can be fused together to both

estimate and control the attitude of an MAV (Shen et al., 2011; Schauwecker et al., 2012; Macdonald et al., 2014; Mulgaonkar et al., 2015). Additionally to the IMU, MAVs equipped with cameras can also use it to infer the attitude with respect to certain reference features or planar surfaces, as in Schauwecker and Zell (2014). Thurrowgood et al. (2009), Dusha et al. (2011), de Croon et al. (2012), and Carrio et al. (2018) estimate the roll and pitch angles of an MAV based on the horizon line (outdoors). The measurements from the IMU and vision can then be filtered together to improve the estimate as well as filter out the accumulating bias from the IMU (Martinelli, 2011). Once known, attitude control can be achieved with a variety of controllers. For a recent survey that treats the topic of attitude control in more detail, we refer the reader to the review by Nascimento and Saska (2019). Of particular interest to swarming are controllers that can provide robustness to disturbances or mishaps. One interesting example is the scheme devised by Faessler et al. (2015), which can automatically re-initialize the leveled flight of an MAV in mid-air.

Measuring and controlling the heading (for instance, with respect to North) is not strictly needed for basic flight. However, it can be an enabler for collective motion by providing a common reference that can be measured locally by all MAVs (Flocchini et al., 2008). Heading with respect to North can be measured with a magnetometer, which is a common component for MAVs (Beard, 2007). A main limitation of this sensor is that it is highly sensitive to disturbances in the environment (Afzal et al., 2011). The disturbances can be corrected for with the use of other attitude sensors. For example, Pascoal et al. (2000) fused gyroscope measurements with the magnetometer in order to filter out disturbances from the magnetometer while also reducing the noise from the gyroscope. Another sensor that has been explored is the celestial compass, which extracts the orientation based on the Sun (Jung et al., 2013; Dupeyroux et al., 2019). Although this sensor is not subject to electro-magnetic disturbances, it is limited to outdoor scenarios and performs best under a clear sky, which may also not always be the case.

4.1.2. Velocity and Odometry

A tuned sensor fusion filter with an accurate prediction model can estimate velocity just based on the IMU readings (Leishman et al., 2014). However, the use of additional and dedicated velocity sensors is commonly used to achieve a more robust system

²When we relate this to nature, then low-level control and state-estimation seldom requires large “computational” efforts by the individual animal. Rather, they eventually become second nature (Rasmussen, 1983). The real focus is directed to higher level tasks.

without bias. Fixed wing MAVs can be equipped with a pitot tube in order to measure airspeed (Chung et al., 2016). For other designs, such as quadrotors, a popular solution is to measure the optic flow, i.e., the motion of features in the environment, from which an MAV can extract its own velocity (Santamaria-Navarro et al., 2015). To observe velocity, the flow needs to be scaled with the help of a distance measurement, such as height (albeit this assumes that the ground is flat, which may be untrue in cluttered/outdoor environments). Optic flow can be measured with a camera or with dedicated sensors, such as PX4FLOW (Honegger et al., 2013) or the PixArt sensor³. Using optical mouse sensors, Briod et al. (2013) were able to make a 46 g quadrotor fly based on only inertial and optical-flow sensors, even without the need to scale the flow by a distance measurement. This was achieved by only using the direction of the optic flow and disregarding its magnitude. In nature, optic flow has also been shown to be directly correlated with how insects control their velocity in an environment (Portelli et al., 2011; Lecoœur et al., 2019). Similar ideas have also been ported to the drone world, whereby the optic flow detection is directly correlated to a control input, without even necessarily extracting states from it (Zufferey et al., 2010). This can be an attractive property in order to create a natural correlation between a sensor and its control properties. State estimates improve when optic flow is fused with other sensors, such as IMU readings or pressure sensors (Kendoul et al., 2009a,b; Santamaria-Navarro et al., 2015), or with the control input of the drone (Ho et al., 2017). As opposed to optic flow sensors, a camera has the advantage that it can observe both optic flow as well as other features in the environment, thus enabling an MAV to get more out of a single sensor. Although this is more computationally expensive, it also provides versatility.

The use of vision also enables the tracking of features in the environment, which a robot can use to estimate its odometry. Using Visual Odometry (VO), a robot integrates vision-based measurements during flight in order to estimate its motion. The inertial variant of VO, known as Visual Inertial Odometry (VIO), further fuses visual tracking together with IMU measurements. This makes it possible for an MAV to move accurately relative to an initial position (Scaramuzza and Zhang, 2019). VIO has been exploited for swarm-like behaviors, such as in the work by Weinstein et al. (2018), whereby twelve MAVs form patterns by flying pre-planned trajectories and use VIO to track their motion. A step beyond VO and its variants is to use Simultaneous Localization And Mapping (SLAM). The advantage of SLAM is that it can mitigate the integration drift of VO-based methods. When solving the full SLAM problem, a robot estimates its odometry in the environment and then corrects it by recognizing previously visited places and optimizing the result accordingly, so as to make a consistent map (Cadena et al., 2016; Cieslewski and Scaramuzza, 2017). Yousif et al. (2015) and Cadena et al. (2016) provide more in-depth reviews of VO and SLAM algorithms. Within the swarming context, a map can also be shared so as to make use of places and features that have been seen by other members of the swarm. One common drawback of VO

and SLAM methods is that they are computationally intensive and thus reserved for larger MAVs (Ghadiok et al., 2012; Schauwecker and Zell, 2014). However, recent developments have also seen the introduction of more light-weight solutions, such as Navion (Suleiman et al., 2018).

Odometry and SLAM are not limited to the use of vision. A viable alternative sensor is the LIDAR (Light Detection and Ranging) scanner, more commonly referred to as “laser scanner.” LIDAR-based SLAM feature the same philosophy as the vision counterparts, but instead of a camera it uses LIDAR to measure depth information and build a map (Bachrach et al., 2011; Opromolla et al., 2016; Doer et al., 2017; Tripicchio et al., 2018). A LIDAR is generally less dependent on lighting conditions and needs less computations, but it is also heavier, more expensive, and consumes more on-board power (Opromolla et al., 2016). Vision and LIDAR can also be used together to further enhance the final estimates (López et al., 2016; Shi et al., 2016).

4.1.3. Height and Altitude

In an abstract sense, the ground represents an obstacle that the MAV must avoid, much like walls, objects, or other MAVs. It does not need to be explicitly known in order to control an MAV, as shown in the work of Beyeler et al. (2009). Unlike other obstacles, however, gravity continuously pulls the MAV toward the ground, meaning that measuring and controlling height and altitude often requires special attention.

Note that we differentiate here between height and altitude. Height is the distance to the ground surface, which can vary when there is a high building, a canyon, or a table. The height of an MAV can be measured with an ultrasonic range finder (or “sonar”). Sonar can provide more accurate data at the cost of power, mass, size, and a limited range. Its accuracy, however, made it a part of several designs (Krajník et al., 2011; Ghadiok et al., 2012; Abeywardena et al., 2013). Infra-red or laser range finders have also been used as an alternative (Grzonka et al., 2009; Gupte et al., 2012). The advantage of an infra-red sensor is that it can be very power efficient, albeit it is only reliable up to a limited range of a few meters, and on favorable light conditions (Laković et al., 2019)⁴. Altitude is the distance to a fixed reference point, such as sea level or a take-off position. A pressure sensor is a common sensor to obtain this measurement (Beard, 2007), but it can be subject to large noise and disturbances in the short term, which can be reduced via low pass filters (Sabatini and Genovese, 2013; Shilov, 2014). If flying outdoors, a Global Navigation Satellite System (GNSS) can also be used to obtain altitude.

The choice of height/altitude sensor has an impact on the swarm behaviors that can be programmed. GNSS and pressure sensors provide a measurement of the altitude of the MAV with respect to a certain position. This is an attractive property, although, as previously discussed, GNSS is limited to outdoor environments, while pressure sensors can be noisy. Moreover, all pressure sensors of all MAVs in the swarm should be equally calibrated. Unlike pressure sensors, ultrasonic sensors or laser range finders do not require this calibration step, since the

³ “PMW3901MB Product Datasheet” by PixArt Imaging Inc., June 2017.

⁴ See www.st.com/en/imaging-and-photonics-solutions/vl53l0x.html

measurement is made from the MAV to the nearest surface. However, one must then assume that the MAVs all fly on a flat plane with no objects (or other MAVs below them), which may turn out to not be a valid assumption. SLAM and VIO methods, previously discussed in section 4.1.2, can also estimate altitude/height as part of the odometry/mapping procedure provided that a downwards facing camera is available.

Just as for the use of a common heading like North, the measurements of height and/or altitude can provide a common reference plane for a swarm of MAVs. If the vertical distance between the MAVs is sufficient, it can provide a relatively simple solution for intra-swarm collision avoidance (albeit with constraints—we return to this in section 5.2). It can also enable self-organized behaviors, such as in the work of Chung et al. (2016), where the MAVs are made to follow the one with the highest altitude within their sub-swarm. In this way, the leader is automatically elected in a self-organized manner by the swarm. For example, should a current leader MAV need to land as a result of a malfunction, a new leader can be automatically re-elected so that the rest of the swarm can keep operating.

4.2. Achieving Safe Navigation

It is important that each MAV remains safe and that it does not collide with its surroundings, or that damages remain limited in case this happens. This safety requirement can be satisfied in two ways. The first, which is more “passive” and brings us back to MAV design, is to develop MAVs that are mechanically collision resilient. This allows the MAV to hit obstacles without risking significant damage to itself or its environment. With this rationale, Briod et al. (2012), Mulgaonkar et al. (2015, 2018), and Kornatowski et al. (2017) placed protective cages around an MAV. However, the additional mass of a cage can negatively impact flight time and the cage can also introduce drag and controllability issues (Floreano et al., 2017). Instead, Mintchev et al. (2017) developed a flexible design for miniature quadrotors in order to be more collision resilient upon impact with walls. The use of airships has also been proposed as a more collision resilient solution (Melhuish and Welsby, 2002; Troub et al., 2017). The limitations of airships, however, are in their lower agility and restricted payload capacity. More recently, Chen et al. (2019) demonstrated insect scale designs that use soft artificial muscles for flapping flight. The soft actuators, combined with the small scale of the MAV, are such that the MAVs can be physically robust to collisions with obstacles and with each other. Collision resistant designs can even be exploited to improve on-board state estimation, such as in the recent work by Lew et al. (2019), whereby collisions are used as pseudo velocity measurement under the assumption that the velocity perpendicular to an obstacle, at the time of impact, is null. The alternative, or complementary, solution to passive collision resistance is “active” obstacle sensing and avoidance, whereby an MAV uses its on-board sensors to identify and avoid obstacles in the environment.

Collision-free flight can be achieved via two main navigation philosophies: (1) map-based navigation, and (2) reactive navigation. With the former, a map of the environment can be used to create a collision-free trajectory (Shen et al., 2011; Weiss

et al., 2011; Ghadiok et al., 2012). The map can be generated during flight (using SLAM) and/or, for known environments, it can be provided a-priori. The advantage of a map-based approach is that obstacle avoidance can be directly integrated with higher level swarming behaviors (Saska et al., 2016b). Instead, a reactive control strategy uses a different philosophy whereby the MAV only reacts to obstacles in real-time as they are measured, regardless of its absolute position within the environment. In this case, if an MAV detects an obstacle, it reacts with an avoidance maneuver without taking its higher level goal into account. The trajectories pursued with a reactive controller may be less optimal, but the advantage of a reactive control strategy is that it naturally accounts for dynamic obstacles and it is not limited to a static map. The two can also operate in a hierarchical manner, such that the reactive controller takes over if there is a need to avoid an obstacle, and the MAV is otherwise controlled at a higher level by a path planning behavior. Regardless of the navigation philosophy in use, if the MAV needs to sense and avoid obstacles during flight, it will require sensors that can provide it with the right information in a timely manner.

Of all sensors, vision provides a vast amount of information from which an MAV can interpret its direct environment. By using a stereo-camera, the disparity between two images gives depth information (Heng et al., 2011; Matthies et al., 2014; Oleynikova et al., 2015; McGuire et al., 2017). Alternatively, a single camera can also be used. For example, the work of de Croon et al. (2012) exploited the decrease in the variance of features when approaching obstacles. Ross et al. (2013) used a learning routine to map monocular camera images to a pilot command in order to teach obstacle avoidance by imitating a human pilot. Kong et al. (2014) proposed edge detection to detect the boundary of potential obstacles in an image. Saha et al. (2014) and Aguilar et al. (2017) used feature detection techniques in order to extract potential obstacles from images. Alvarez et al. (2016) used consecutive images to extract a depth map (a technique known as “motion parallax”), albeit the accuracy of this method is dependent on the ego-motion estimation of the quadrotor. Learning approaches have also been investigated in order to overcome the limitations of monocular vision. By exploiting the collision resistant design of a Parrot AR Drone, Gandhi et al. (2017) collected data from 11,500 crashes and used a self-supervised learning approach to teach the drone how to avoid obstacles from only a monocular camera. Self-supervised learning of distance from monocular images can also be accomplished without the need to crash, but with the aid of an additional sensor. Lamers et al. (2016) did this by exploiting an infrared range sensor, and van Hecke et al. (2018) applied this to see distances with one single camera by learning a behavior that used a stereo-camera. This is useful if the stereo-camera were to malfunction and suddenly become monocular. Alternative camera technologies have also been developed, providing new possibilities. RGB-D sensors are cameras that also provide a per-pixel depth map, a mainstream example of which is the Microsoft Kinect camera (Newcombe et al., 2011). This particular sensor augments one RGB camera with an IR camera and an IR projector, which together are capable of measuring depth (Smisek et al., 2013). RGB-D sensors have been used on MAVs

to navigate in an environment and avoid obstacles (Shen et al., 2014; Stegagno et al., 2014; Odelga et al., 2016; Huang et al., 2017). One of the disadvantages of these RGB-D sensors over a stereo-camera set-up (whereby depth is inferred from the disparity) is that RGB-D sensors can be more sensitive to natural light, and may thus perform less well in outdoor environments (Stegagno et al., 2014). Finally, in recent years, the introduction of Dynamic Vision Sensor (DVS) cameras has also enabled new possibilities for reactive obstacle sensing. A DVS camera only measures changes in the brightness, and can thus provide a higher data throughput. This enables a robot to quickly react to sudden changes in the environment, such as the appearance of a fast moving obstacle (Mueggler et al., 2015; Falanga et al., 2019a).

The capabilities of a vision algorithm will depend on the resolution of the on-board cameras, the number of the on-board cameras, as well as the processing power on-board. On very lightweight MAVs, such as flapping wings, even carrying a small stereo-camera can be challenging (Olejnik et al., 2019). A further known disadvantage of vision is the limited Field of View (FOV) of cameras. Omni-directional sensing can only be achieved with multiple sets of cameras (Floreano et al., 2013; Moore et al., 2014) at the cost of additional mass, the impact of which is dependent on the design of the MAV.

Although vision is a rich sensor, in that it can provide different types of information, other sensors also can be used for reactive collision avoidance. LIDAR, for instance, has the advantage that it is less dependent on lighting conditions and can provide more accurate data for localization and navigation (Bachrach et al., 2011; Tripicchio et al., 2018). Alternatively, time-of-flight laser ranging sensors have also been proposed for reactive obstacle avoidance algorithms on small drones (Laković et al., 2019). These uni-directional sensors can sense whether an object appears along their line of sight (typically up to a few meters). Due to their small size and low power requirements, they can be used on tiny MAVs (Bitcraze, 2019)⁵.

5. INTRA-SWARM RELATIVE SENSING AND COLLISION AVOIDANCE

Once we have an MAV design that can perform basic safe flight, we begin to expand its capabilities toward collaboration in a swarm. Two fundamental challenges need to be considered in this domain. The first is relative localization. This is not only required to ensure intra-swarm collision avoidance, which is a basic safety requirement, but also to enable several swarm behaviors (Bouffanais, 2016). The design choice used for intra-swarm relative localization defines and constrains the motion of the MAVs relative to one another, which affects the swarming behavior that can be implemented. The second challenge is intra-swarm communication. Much like knowing the position of neighbors, the exchange of information between MAVs can help the swarm to coordinate (Valentini, 2017; Hamann, 2018). In this section, we explore the state of the art for relative localization (section 5.1), reactive collision avoidance maneuvers

(section 5.2), and we discuss intra-swarm communication technologies (section 5.3).

5.1. Relative Localization

In outdoor environments, relative position can be obtained via a combination of GNSS and intra-swarm communication. Global position information obtained via GNSS is communicated between MAVs and then used to extract relative position information. This has enabled connected swarms that can operate in formations or flocks (Chung et al., 2016; Yuan et al., 2017). An impressive recent display of this in the real world was put into practice by Vászárhelyi et al. (2018), who programmed a swarm of 30 MAVs to flock. The same concept can be applied to indoor environments if pre-fitted with, for example: external markers (Pestana et al., 2014), motion-tracking cameras (Kushleyev et al., 2013), antenna beacons (Ledergerber et al., 2015; Guo et al., 2016), or ultra sound beacons (Vedder et al., 2015). However, this dependency on external infrastructure limits the swarm to being operable only in areas that have been properly fitted to the task. Several tasks, especially the ones that involve exploration, cannot rely on these methods. In order to remove the dependency on external infrastructure, there is a need for technologies that allows the MAVs themselves to obtain a direct MAV-to-MAV relative location estimate. This is still an open challenge, with several technologies and sensors currently being developed.

One of the earlier solutions for direct relative localization on flying robots proposed the use of infrared sensors (Roberts et al., 2012). However, since infrared sensors are uni-directional, this used an array of sensors (both emitting and receiving) placed around the MAV in order to approach omni-directionality, making for a relatively heavy system. Alternatively, vision-based algorithms have once again been extensively explored. However, the robust visual detection of neighboring MAVs is not a simple task. The object needs to be recognized at different angles, positions, speeds, and sizes. Moreover, the image can be subject to blur or poor lighting conditions. One way to address this challenge is with the use of visual aids mounted on the MAVs, such as visual markers (Faigl et al., 2013; Krajník et al., 2014; Nägeli et al., 2014), colored balls (Roelofsen et al., 2015; Epstein and Feldman, 2018), or active markers, such as infrared markers (Faessler et al., 2014; Teixeira et al., 2018)⁶ or Ultra Violet (UV) markers (Walter et al., 2018, 2019). Visual aids simplify the task and improve the detection accuracy and reliability. However, they are not as easily feasible on all designs, such as flapping wing MAVs or smaller quadrotors. Marker-less detection of other MAVs is very challenging, since other MAVs have to be detected against cluttered, possibly dynamic backgrounds while the detecting MAV is moving by itself as well. A successful current approach is to rely on stereo vision, where other drones can be detected because they “float” in the air unlike other objects like trees or buildings. Carrio et al. (2018) explored a deep learning algorithm for the detection of other MAVs in stereo-based disparity images. An alternative is to detect other MAVs in monocular still images. Like the detection in

⁵See www.bitcraze.io/multi-ranger-deck/

⁶The solution by Teixeira et al. (2018) additionally uses communication between the MAVs.

stereo disparity images, this removes the difficulty of interpreting complex motion fields between frames, but it introduces the difficulty of detecting other, potentially (seemingly) small MAVs against background clutter. To solve the challenge, Opromolla et al. (2019) used a machine learning framework that exploited the knowledge that the MAVs were supposed to fly in formation. Their scheme used the knowledge of the formation in order to predict the expected position of a neighboring MAV and focus the vision-based detection on the expected region, thus simplifying the task. Employing a more end-to-end learning technique, Schilling et al. (2019) used imitation learning to autonomously learn a flocking behavior from camera images. Following the attribution method by Selvaraju et al. (2017), Schilling et al. studied the influence that each pixel of an input image had on the predicted velocity. It was shown that the parts of the image whereby neighboring MAVs could be seen were more influential, demonstrating that the network had implicitly learned to localize its neighbors. Despite the promising preliminary results, it is yet to be seen how it can handle other MAVs sizes or more cluttered backgrounds. Finally, it is possible to use the optic flow field for detecting other MAVs. This approach could have the benefit of generality, but it would require the calculation and interpretation of a complex, dense optic flow field. To our knowledge, this method has not yet been investigated.

From a swarming perspective, it may also be desirable to know the ID of a neighbor. However, IDs may be difficult to detect using vision without the aid of markers. This issue was explored by Stegagno et al. (2011), Cognetti et al. (2012), and Franchi et al. (2013) with fusion filters that infer IDs over time with the aid of communication. Moreover, cameras have a limited FOV. This limits the behaviors that can be achieved by the swarm. For instance, it may be limiting for surveillance tasks where quadrotors may need to look away from each other but can't or else they may collide or disperse. It can be addressed by placing several cameras around the MAVs (Schilling et al., 2019), but at the cost of additional mass, size, and power, which in turn creates new repercussions.

The use of vision is not only limited to directly recognizing other drones in the environment. With the aid of communication, two or more MAVs can also estimate their relative location indirectly by matching mutually observed features in the environment. The MAVs can compare their respective views and infer their relative location. In the most complete case, each MAV uses a SLAM algorithm to construct a map of its environment, which is then compared in full (as discussed in section 4, this can also be accomplished using other sensors, such as LIDAR, so this approach is not only reserved for vision). Although SLAM is a computationally expensive task, more easily handled centrally (Achtelik et al., 2012; Forster et al., 2013), it can also be run in a distributed manner, making for an infrastructure free system (Cunningham et al., 2013; Cieslewski et al., 2018; Lajoie et al., 2019). For a survey of collaborative visual SLAM, we refer the reader to the paper by Zou et al. (2019) and the sources therein. An additional benefit of collective map generation is that the MAVs benefit from the observations of their team-mates and can thus achieve a better collective map.

However, if the desired objective is only to achieve relative localization, the computations can be simplified. Instead of computing and matching an entire map, the MAVs need only to concern themselves with the comparison of mutually observed features in order to extract their relative geometric pose (Achtelik et al., 2011; Montijano et al., 2016). This requires that the images compared by the MAVs have sufficient overlap and can be uniquely identified.

An alternative stream of research leverages only communication between MAVs to achieve relative localization, while also using the antennas as relative range sensors. Here, we will refer to these methods as *communication-based ranging*. The advantage of this method is that it offers omni-directional information at a relatively low mass, power, and processing penalty, leveraging a technology that is likely available on even the smallest of MAVs. Szabo (2015) first proposed the use of signal strength to detect the presence of nearby MAVs and engage in avoidance maneuvers. Also for the purposes of collision avoidance, Coppola et al. (2018) implemented a beacon-less relative localization approach based on the signal strength between antennas, using the Bluetooth Low Energy connectivity already available on even the smaller drones. Guo et al. (2017) proposed a similar solution using UltraWide Band (UWB) antennas for relative ranging, which offer a higher resolution even at larger distances. However, this work used one of the drones as a reference beacon for the others. One commonality between the solutions by Guo et al. (2017) and Coppola et al. (2018) was that the MAVs were required to have a knowledge of North, which enabled them to compare each other's velocities along the same global axis. However, in practice this is a significant limitation due to the difficulties of reliably measuring North, especially if indoors, as already discussed in section 4.1.1. To tackle this, van der Helm et al. (2019) showed that, if using a high accuracy ranging antenna, such as UWB, then it is not necessary for the MAVs to measure a common North. However, selecting this option creates fundamental constraints on the high-level behaviors of the swarm. This issue is there for the case where North is known and when it is not, albeit the requirement when North is not known are more stringent. If North is known, at least one of the MAVs must be moving relative to the other for the relative localization to remain theoretically observable. If North is not known, all MAVs must be moving. The MAVs remain bound to trajectories that excite the filter (van der Helm et al., 2019). For the case where North is known, Nguyen et al. (2019) proposed that a portion of MAVs in the swarm should act as "observers" and perform trajectories that persistently excite the system.

Another solution is to use sound. Early research in this domain was performed by Tijs et al. (2010), who used a microphone to hear nearby MAVs. This was explored in more depth by Basiri (2015) using full microphone arrays for relative localization. A primary issue encountered was that the sound emitted by the listening quadrotor would mask the sound of the neighboring MAVs, which were also similar. This was addressed with the use of a "chirp" sound, which can then be easily heard by neighbors, in order to overcome this issue (Basiri et al., 2014, 2016). In recent work, Cabrera-Ponce et al. (2019) proposed the

use of a Convolutional Neural Network to detect the presence of nearby MAVs. This is done using a large scale microphone array (Ruiz-Espitia et al., 2018) featuring eight microphones based on the ManyEars framework (Grondin et al., 2013). Specific to sound sensors, the accuracy of the detection depends on how similar the sounds of other MAVs are. Moreover, the localization accuracy depends on the microphone setup. Most works use a microphone array, where the localization accuracy depends on the length of the baseline between microphones, which is inherently limited on small MAVs.

As it can be seen, several different techniques exist. Minimally, these technologies should enable neighboring MAVs to avoid collisions with one another. However, the particular choice of relative localization technology creates a fundamental constraint on the swarm behavior that can be achieved. For example, communication-based ranging methods have unobservable conditions depending on the MAVs' motion, and sound-based localization with microphone arrays will be less accurate when used on smaller MAVs. Similarly, certain swarm behaviors (e.g., one that requires known IDs, or long range distances) may place certain requirements on which technology is best to be used. In **Table 1**, we outline the major relative localization approaches with their advantages and disadvantages.

5.2. Intra-Swarm Collision Avoidance

Collision detection and avoidance of objects in the environment has already been discussed in section 4.2. As MAVs operate in teams, relative intra-swarm collision avoidance also becomes a safety-critical behavior that should be implemented. The complexity of this task is that it requires a collaborative maneuver between two or more MAVs.

MAVs operate in 3D space, and thus relative collision avoidance could be tackled by vertical separation. However, particularly in indoor environments where vertical space is limited, vertical avoidance maneuvers may cause undesirable aerodynamic interactions with other MAVs as well as other parts of the environment. For quadrotors, while aerodynamic influence is negligible when flying side-by-side, flying above another will create a disturbance for the lower one (Michael et al., 2010; Powers et al., 2013). Furthermore, emergency vertical maneuvers could also cause a quadrotor to fly too close to the ground, which creates a ground effect and pushes it upwards, or, if indoors, to fly too close to the ceiling, which creates a pulling effect toward the ceiling (Powers et al., 2013). Vertical avoidance may also corrupt the sensor readings of the MAV. For instance, height may be compromised if another MAV obstructs a sonar sensors. Overall, horizontal avoidance maneuvers are desired.

A popular algorithm for obstacle avoidance, provided that the robots know their relative position and velocity, is the Velocity Obstacle (VO) method (Fiorini and Shiller, 1998). The core idea is for a robot to determine a set of all velocities that will lead to collisions with the obstacle (a collision cone), and then choose a velocity outside of that set, usually the one that requires minimum change from the current velocity. VO has stemmed a number of variants specifically designed to deal with multi-agent avoidance, such as Reciprocal Velocity Obstacle (RVO) (van den Berg et al., 2008; van den Berg et al., 2011), Hybrid

Reciprocal Velocity Obstacle (HRVO) (Snape et al., 2009), and Optimal Reciprocal Collision Avoidance (ORCA) (Snape et al., 2011). These variants alter the set of forbidden velocities in order to address reciprocity, which may otherwise lead to oscillations in the behavior. These methods have been successfully applied on MAVs, both in a decentralized way as well as via centralized re-planners. They accounted for uncertainties by artificially increasing the perceived radii of the robots. Alonso-Mora et al. (2015) showed the successful use of RVO on a team of MAVs such that they may adjust their trajectory with respect to a reference. This was done using an external MCS for (relative) positioning. Coppola et al. (2018) showed a collision cone scheme with on-board relative localization, introducing a method to adjust the cone angle in order to better account for uncertainties in the relative localization estimates. A disadvantage of VO methods and its derivatives is scalability. If the flying area is limited and the airspace becomes too crowded, then it may become difficult for MAVs to find safe directions to fly toward (Coppola et al., 2018). Another avoidance algorithm, called Human-Like (HL), presents the advantage that the heading selection is decoupled from speed selection (Guzzi et al., 2013a; Guzzi et al., 2014), such that the MAVs only engage in a change in heading. HL has been found to be successful even when operating at relatively lower rates (Guzzi et al., 2013b). Although it has not been tested on MAVs, their tests also demonstrated generally better scalability properties.

Alternatively, attraction and repulsion forces between obstacles are also a valid algorithm for collision avoidance. This is a common technique which has been extensively studied in swarm research (Reynolds, 1987; Gazi and Passino, 2002; Gazi and Passino, 2004). If one wishes for the MAVs to flock, these attraction and repulsion forces can also be directly merged with the swarm controller (Vásárhelyi et al., 2018). One potential short-coming of this approach is that it can lead to equilibrium states whereby the swarm remains in a fixed final formation, although this can also be seen as a positive property that can be exploited (Gazi and Passino, 2011).

In summary, multiple methods exist for intra-swarm collision avoidance. Given sufficiently accurate relative locations, these methods are very successful. The main challenges here are: (1) how to deal with uncertainties and unobservable conditions deriving from the localization mechanism used by the drones, and (2) how to keep guaranteeing successful collision avoidance when the swarm scales up to very large numbers.

5.3. Intra-Swarm Communication

Direct sharing of information between neighboring robots is an enabler for swarm behaviors as well as relative sensing (Valentini, 2017; Hamann, 2018; Pitonakova et al., 2018). To achieve the desired effect, it needs to be implemented with scalability, robustness, and flexibility in mind. Common problems that can otherwise arise are: (1) the messaging rate between robots is too low (low scalability); (2) high packet loss (low robustness); (3) communication range is too low (low scalability and flexibility); (4) inability to adapt to a switching network topology (low flexibility) (Chamanbaz et al., 2017).

Solutions to the above depend on the application. With respect to hardware, the three main technologies in the state

TABLE 1 | Current technologies in the state of the art for relative localization between MAVs, with their main advantages and disadvantages.

Technology	Sample references	Advantages	Disadvantages
Vision (direct, passive)	Faigl et al. (2013) Krajník et al. (2014) Nägeli et al. (2014) Roelofsen et al. (2015) Carrio et al. (2018)	<ul style="list-style-type: none"> • Rich information • Passive sensor • Possible to extract ID (if using markers or having otherwise visually distinct MAVs) • Lightweight (depending on model) • Scalable (provided environment is not cluttered) 	<ul style="list-style-type: none"> • Computationally expensive • Limited FOV • Dependent on lighting conditions • Dependent on visual clutter in environment • No IDs (if marker-less) • Needs visual line of sight
Observation matching	Achtelik et al. (2011) Montijano et al. (2016)	<ul style="list-style-type: none"> • No direct line of sight is needed • Thrives in visually cluttered environments • Includes IDs (via communication) 	<ul style="list-style-type: none"> • Active sensor, requires communication • Requires sufficient visual overlap • Computationally expensive • Dependent on lighting conditions
Communication-based ranging	Szabo (2015) Guo et al. (2017) Coppola et al. (2018) van der Helm et al. (2019) Nguyen et al. (2019)	<ul style="list-style-type: none"> • Low mass • Omni-directional • Possible to extract IDs • Can work in visually cluttered environments • Enables communication of additional data • Scalable (provided that the sound environment is, or does not become, not cluttered) • Passive sensor • Omni-directional • Can work in visually cluttered environments 	<ul style="list-style-type: none"> • Active sensor, requires communication • Needs relative excitation maneuvers • Noisy (depending on sensor) • Difficult to scale due to communication interference • Self-propeller noise • Limited range • Noise pollution (if using “chirps”) • No IDs (if chirp-less) • Limited angular accuracy due to limited baseline between microphones in the array
Sound	Basiri et al. (2014, 2016) Cabrera-Ponce et al. (2019)	<ul style="list-style-type: none"> • Accurate • Computationally simple • Lower dependence on lighting conditions^a • Can work in visually cluttered environments 	<ul style="list-style-type: none"> • Heavy • Needs visual line of sight • Many active sensors result in high energy expense
Infra-red (sensor array)	Roberts et al. (2012)	<ul style="list-style-type: none"> • Accurate • Computationally simple • Lower dependence on lighting conditions^a • Can work in visually cluttered environments 	<ul style="list-style-type: none"> • Heavy • Needs visual line of sight • Many active sensors result in high energy expense
Vision (direct, with active markers)	Faessler et al. (2014) Teixeira et al. (2018) Walter et al. (2018, 2019)	<ul style="list-style-type: none"> • Accurate • Possible to extract IDs • Lower dependence on lighting conditions • Can work in visually cluttered environments 	<ul style="list-style-type: none"> • Needs visual line of sight • Many active sensors result in high energy expense

^aRoberts et al. (2012) tested the sensor for 0, 500, and 10,000 lux and found <1% relative error between these lighting conditions. However, the sensor was not tested outdoors.

of the art are: Bluetooth, WiFi, and ZigBee (Bensky, 2019). All three operate in the 2.4 GHz band⁷. Bluetooth is energy efficient, but features a low maximum communication distances of $\approx 10\text{--}20$ m (indoors, depending on the environment and version). This makes it more important to establish a network that can adapt to a switching topology, as it is very likely to change during operations. The latest version of the Bluetooth standard, Bluetooth 5, features a higher range and a higher data-rate despite keeping a low power consumption. It also has longer advertising messages, such that, without pairing, asynchronous network nodes can exchange messages of 255 bytes instead of 31 (Collotta et al., 2018). Bluetooth antennas were used in the previously discussed work of (Coppola et al.,

2018) on a swarm of 3 MAVs to exchange data indoors and to measure their relative range. In comparison to Bluetooth, WiFi is known to be less energy efficient, but works more reliably at longer ranges and has a higher data throughput. Chung et al. (2016) used WiFi to enable a swarm of 50 MAVs to form an *ad-hoc* network. WiFi was also used by Vászrhelyi et al. (2018) in combination with an XBee module⁸ using a proprietary communication protocol. ZigBee's primary benefits are scalability (it can keep up to, theoretically, 64,000 nodes) and low power, although it has a low data communication rate (Bensky, 2019)⁹. Depending on the application, this may or may not be an issue depending on what the intra-swarm communication requirements are. Allred et al. (2007) used a

⁷ WiFi also operates at other frequency bands. The 5 GHz band, for instance, is typically known to feature a lower interference (Verma et al., 2013). ZigBee can also operate at the 868 and 915 MHz frequency bands (Collotta et al., 2018).

⁸Not to be confused with ZigBee (Faludi, 2010).

⁹Note that Bluetooth Low Energy, a sub-version of the Bluetooth standard, also requires very little power. Tests by Collotta et al. (2018) return that Bluetooth 4.2 and 5.0 have a lower power consumption than ZigBee.

ZigBee module to enable communication on a flock of fixed wing MAVs due to its combination of low energy consumption and long range (offering “a range of over 1 mile at 60 mW”). For comparisons of technical details of these technologies we refer the reader to the detailed book by Bensky (2019), the MAV-focused review by Zufferey et al. (2013), as well as the earlier comparisons by Lee et al. (2007).

In addition to the technologies discussed above, there is also the possibility of enabling indirect communication via cellular networks. In the near future, 5G networks are expected to make it possible to have a reliable and high data throughput between several MAVs (Campion et al., 2018). Finally, the use of UWB can also gain more relevance in the future, especially because its additional capability to accurately measure the range between MAVs, as discussed in section 5.1, can be very helpful for swarms. One technological challenge is that communication needs power, and while this may be near-negligible for the bigger MAVs, it is not so for the smaller designs (Petricca et al., 2011). From this perspective, the communication-based relative localization discussed in section 5.1, which can also double as a communication device for MAVs, is an interesting solution if one desires a system that can achieve both goals simultaneously. However, using any relative localization approach that relies on communication means that having a stable connection among MAVs is an important requirement, and possibly a safety critical one. Moreover, high messaging rates also become important in order to have a high update rate.

6. SWARM-LEVEL CONTROL

We finally arrive at the “swarm” part of this paper. Once we have reliable MAVs that can safely fly in an environment, localize one another, and perhaps even communicate, we can begin to exploit them as a swarm. The complexity of this task stems from the fact that, due to the decentralized nature of the swarm, the local actions that a robot takes can have any number of repercussions at the global level. These cannot be known unless the system is fully observed and optimized for, which the individual robot cannot do.

This section discusses possible approaches to design MAV swarm behaviors. Prominent examples of behaviors are: flocking, formation flight, distributed sensing (e.g., mapping/surveillance), and collaborative transport and object manipulation¹⁰. Of these, formation flight receives significant attention. It can be useful for several applications, such as surveillance, mapping, or cinematography so as to collaboratively observe a scene (Mademlis et al., 2019). Additionally, it can also be used for collaborative transport (de Marina and Smeur, 2019), and it has even been shown that certain formations lead to energy efficient flight for groups (Weimerskirch et al., 2001). Flocking behaviors bear similar properties to formation flight, but with more “fluid” inter-agent behaviors that allow the swarm to re-organize according to their current neighborhood and the environment.

¹⁰Note that this list not exhaustive. Additionally, we will see that there may also be overlaps between these behaviors. For example, as explored in section 6.1, flocking behaviors may achieve fixed formations under certain equilibria.

Distributed sensing behaviors may require the swarm to travel in a formation or flock, but may also include behaviors in which the swarm distributes over pre-specified areas (Bähmann et al., 2017) or disperses (McGuire et al., 2019). Collaborative transport and object manipulations take two forms. The first is that of MAVs individually foraging for different objects and bringing them to base (Bähmann et al., 2017), the second is that of jointly carrying a load that is too heavy for the individual MAV to carry (Tagliabue et al., 2019). In order to achieve the behaviors above, and others, the MAVs can also engage in a number of more general swarm behaviors, such as distributed task allocation or collective decision making. For all cases, the challenge is to endow the MAVs with a controller that achieves the desired swarm behavior while also avoiding undesired results (Winfield et al., 2005, 2006).

Similarly to the review by Brambilla et al. (2013) (which the reader is referred to for a general overview of swarm robotics and engineering), we divide the design methods in two categories. The first, which we call “manual design methods,” refers to hand-crafted controllers that instigate a particular behavior in the swarm. These are discussed in section 6.1, where we provide an overview of the state of the art for different swarm behaviors. The second, which we refer to as “automatic design methods,” uses machine learning techniques in order to design and/or optimize the controller for an arbitrary goal. This is discussed in section 6.2. We discuss the advantages and disadvantages between the two, from the perspective of designing swarms of MAVs, in section 6.3.

6.1. Manual Design Methods

This is the “classical” strategy to control, whereby a swarm designer develops the controllers so as to achieve a desired global behavior. For swarm robotics, we differentiate between two approaches. One approach is to design local behaviors, analyze them, and then manually iterate until the swarm behaves as desired. Another approach is to make mathematical models of the robots and their interactions and then design a suitable controller that comes with a certain proof of convergence. The latter approach has some obvious advantages if one succeeds, but it makes the designer face the full complexity of swarm systems. Hence, such methods typically have limited applicability. For example, in the work of Izzo and Pettazzi (2007), the behavior is limited to only symmetrical formations of limited numbers of agents. The preferred approach is dependent on the swarm behavior that the designer wishes to achieve, under the constraints of the local properties of each MAV.

A large portion of methods focuses on formation control algorithms, whereby the goal is for the MAVs to form and/or keep a tight formation during flight. To hold a formation, the MAVs must hold a relative position or distance between given neighbors, such that they can move as one unit through space. See, for instance, the works of Quintero et al. (2013), Schiano et al. (2016), de Marina et al. (2017), Yuan et al. (2017), and de Marina and Smeur (2019). One advantage of flying in formation for MAV swarms is their predictability during operations. Several methods provide robust controllers with mathematical proofs that the formation can be achieved and maintained during flight.

A review dedicated to formation control algorithms for MAVs is provided by Oh et al. (2015). Chung et al. (2018) also discuss different methods.

There are applications for which a rigid formation is sub-optimal, undesired, or unnecessary, and it is better for the MAVs to move through space in a flock. Flocking behaviors were originally synthesized from the motion of animals in nature (Aoki, 1982), and were most famously formalized by Reynolds (1987) with the intent of simulating swarms in computer animations. The behavior is typically characterized by a combination of simple local rules: attraction forces, repulsion forces, heading alignment with neighbors, speed agreement with neighbors. This behavior naturally incorporates collision avoidance via the repulsion rule, and it has also been explored as a means to collectively navigate in an environment with obstacles, whereby the obstacles provide additional repulsion forces (Saska et al., 2014; Saska, 2015). Alternatively, the local rules can also be exploited to achieve formations by making use of equilibrium points between attraction and repulsion forces (Gazi, 2005). Depending on the way in which the rules are used, they can be incorporated into an iterative approach, or they can be made part of a mathematical regime combined with the model of the robot. An early real-world demonstration of distributed flocking was achieved by Hauert et al. (2011) with a swarm of ten fixed wing MAVs. The more recent work by Vászárhelyi et al. (2018) demonstrated outdoor flocking for a swarm of 30 quadrotors.

Concerning behaviors, such as distributed sensing, exploration, or mapping, there are several different types of solutions that have been developed specifically for MAVs. Typically, these are found to vary depending on the nature of the task, requiring the designer to make careful choices on the best algorithm to be used. Bähmann et al. (2017) and Spurný et al. (2019), aided by GNSS for positioning, divided a search area into multiple regions so that a team of three MAVs could efficiently explore it with a pre-planned trajectory. The recent work of McGuire et al. (2019) demonstrated a swarm of six Crazyflie MAVs performing an autonomous exploration task in an unknown indoor environment. Each MAV acted entirely locally based on a manually designed bug algorithm which enabled exploration as well as homing to a reference beacon.

6.2. Automatic Methods for Behavior Design and Optimization

In the last few decades, the increasing power of machine learning methods cannot be denied, with multiple examples in robotics, autonomous driving, smart-homes, and more. Machine learning techniques offer a way to automatically extract the local controller that can fulfill a task, relieving us from the need to design it ourselves. However, the problem shifts to devising algorithms that can efficiently and effectively discover the controllers for us. In this section, we discuss the possibilities based on two primary machine learning approaches in swarm intelligence research: Evolutionary Robotics (ER) and Reinforcement Learning (RL).

6.2.1. Evolutionary Robotics

ER uses the concept of *survival of the fittest* in order to efficiently search through the design space for an effective controller (Nolfi,

2002)¹¹. It has been widely adopted in swarm robotics literature in order to evolve local robot controllers that optimize the performance of the swarm with respect to a global, swarm-level objective (Trianni, 2008). ER bypasses the analysis of the relation between the local controllers and the global behavior of the swarm. Instead, it optimizes the controllers “blindly” by means of several evaluations in an evolutionary process, which most often happens in simulation, but can also be performed in the real world (Eiben, 2014). Evolved solutions often exploit the robots’ bodies and environment, including the behaviors of other swarm members. Moreover, thanks to the blind optimization, not only the controller can be evolved, but also other factors, such as the communication between robots (Ampatzis et al., 2008). Likewise, ER offers a generic approach to generate swarm controllers of different types, including, but not limited to: neural networks (Trianni et al., 2003; Silva et al., 2015), grammar rules (Ferrante et al., 2013), behavior trees (Scheper et al., 2016; Jones et al., 2018, 2019), and state machines (Francesca et al., 2014). Although neural network architectures can be very powerful, the advantage of the latter methods is that they can be better understood by a designer, which makes it easier to cross the “reality gap” between simulation and the real world when deploying the controllers on the real robots (Jones et al., 2019). Crossing the reality gap is a major challenge in the field of ER and many different approaches have been investigated, also for neural networks. See Scheper (2019) for a more extensive discussion on these methods.

A major challenge for the effective use of ER, especially for swarm robotics, is the design of the fitness functions to be optimized (Francesca and Birattari, 2016). This is usually left to the designer’s ability to explicitly define the key elements that indicate the success of a behavior in a measurable and quantitative manner. It is not uncommon to see empirically defined parameters that represent certain desired elements, such as “safety” in the example of Duarte et al. (2016). As task complexity increases, so does the challenge of designing a fitness function. In the worst case, it may become *uninformative* or even *deceptive*, leading the algorithm to not finding the desired behavior (Silva et al., 2016). Different approaches have been proposed to tackle this issue, such as behavioral decomposition or incremental learning (Nelson et al., 2009). The risk with these strategies, however, is that the designer shapes the learning of the task too much, which may lead to sub-optimal performances. As an alternative strategy for learning complex tasks, Lehman and Stanley (2011) proposed novelty search, whereby the fitness is not defined by how well the task is performed, but by how “novel” a behavior is. This can lead to finding more unorthodox solutions, also for swarm robotics (Gomes et al., 2013). Potential drawbacks of this approach are that the search becomes less directed, and that the shaping shifts from defining a fitness function to defining what constitutes a “behavior.”

¹¹ Looking at the complexity achieved by natural swarming systems, it also seems intuitive that such complexity could be achieved automatically by mimicking an evolutionary process (Bouffanais, 2016). It is no surprise that a closely related discipline to ER is that of Artificial Life (AL), dedicated to artificially representing life-like processes, albeit with generally more open ended exploratory goals (Bedau, 2003; Trianni, 2014).

To conclude, the ER approach applied to swarming has the large advantage that it deals with complexity by actually bypassing it. However, this currently comes at the cost of needing many evaluations involving the simulation of not one but multiple robots, which leads to longer lasting evolutions. An additional problem of simulating a specific number of robots to evolve a swarm behavior is that the evolution may overfit the behaviors not only to the (simulation) environment, but also to the exact number of robots that were used during the evolution. A naive solution is to simulate different swarm sizes over the evolution, but this will take even more simulation time, and in any case, the number of robots will be limited, meaning that scalability is not guaranteed. Recent developments in this domain have seen the introduction of size-agnostic techniques (Coppola et al., 2019). Finally, although there are studies on online evolutionary learning for swarm robotics (Bredeche et al., 2018), online evolutionary strategies have yet to be explored (in practice) for MAVs.

6.2.2. Reinforcement Learning

With RL, a robot is made to learn by trial-and-error from interacting with its environment under a certain reward scheme. This approach teaches the robot an optimal mapping between a state and the action that it should take so as to maximize its final reward (Sutton and Barto, 2018). RL has been widely used in robotics, and it has thus also found its way to swarm robotics (Brambilla et al., 2013). The advantage of RL is that the robots can explore the environment and continuously adapt their behavior. Several techniques have been proposed over the years for multi-agent RL (Busoniu et al., 2008). However, within swarm robotics literature, it has generally received less attention than ER (Brambilla et al., 2013). A main difficulty with this approach is that, from the perspective of the individual robot, being in a swarm is a non-Markovian task, and each robot only has a partial observation of the full global state. A potential issue, for instance, is “state aliasing,” which refers to when multiple states appear to be the same from the perspective of the agent, even though they are not (McCallum, 1997). It has been demonstrated that ER can achieve better solutions for non-Markovian tasks (de Croon et al., 2005).

The solution to use RL with non-Markovian task leads to a Partially Observable Markov Decision Problem (POMDP). In this case, a robot keeps a history of its observations and thus extracts the most likely global state from them. RL can be applied to POMDPs (Ishii et al., 2005), yet features scalability issues (the so called “state explosion”), especially when ported to the swarm domain because the global state of the swarm, which it tries to estimate, can take exponentially many forms (Parsons and Wooldridge, 2002). In recent work, Hüttenrauch et al. (2019) proposed to use mean feature embeddings which encode a mean distribution of the agents. This compression is then invariant to the number of agents in the swarm. Another known difficulty of RL with respect to ER is the credit assignment problem. This refers to the challenge of decomposing the global rewards into local rewards for each robot, as the individual contribution of a single robot to a global task may not always be clearly determined (Brambilla et al., 2013). The credit assignment problem is also

manifested over time, as it is difficult to judge which prior action was most conducive.

In short, until now ER appears to be a more appropriate choice for learning control in swarms, as it allows robots to exploit non-Markovian properties of the problem (e.g., the states and behaviors of other robots). However, because of the reality gap, online learning methods may turn out very useful in the future, including RL methods.

6.3. Manual vs. Automatic Methods for MAV Swarms

A primary advantage of manual design methods for MAV swarms is that the solutions are generally better understood, given that they have to be designed and programmed manually. The algorithms that are developed can be analyzed, and in certain cases it can even be assessed whether the system will converge to the desired properties and even be resilient to faults (Saldaña et al., 2017; Saulnier et al., 2017). This is a particularly attractive property for MAV applications, where safety and predictability are a primary concern. A second advantage is that they carry a clearer breakdown of the requirements. For these reasons, it is not surprising that, to the best of our knowledge and as confirmed by Chung et al. (2018), most real-world implementations of MAV swarms to date have relied on primarily manually designed swarming algorithms. These advantages have also been acknowledged by the automatic design community, which has brought a general interest in using automatic approach to develop explicit controllers, such as state machines (Francesca et al., 2014, 2015) or behavior trees (Kuckling et al., 2018; Jones et al., 2019). In future work, the use of these methods could lead to a compromise between extracting an understandable controller and exploiting the power of automatic methods.

A challenge of designing an algorithm manually is in the need to ensure that it can work within the limitations of the system. For instance, if using a communication-based ranging relative localization system, the relative location estimate is only observable when both MAVs are moving in such a way that the system is excited (Nguyen et al., 2019). Alternatively, cameras can be limited by the FOV and be forced to keep a reference neighbor in the center (Nägeli et al., 2014). This may be undesirable for the final application of the swarm (e.g., surveillance), since the camera is kept pointing to other MAVs as opposed to interesting features in the environment. Examples, such as these serve to show how a manually designed algorithm can either fail to regard certain elements, or may not exploit the environment optimally so as to best deal with the limitations. An automatic method, on the other hand, could extract a controller that best deals with the limitations, possibly finding solutions that cannot be easily designed manually. For instance, ER studies show that evolved robot controllers can find behaviors that tightly exploit the sensory and motor capabilities of the given robot (Nolfi, 2002)—this is called sensory-motor coordination.

Despite their power, the application of automatic design methods to MAV swarms are relatively few. One of the first steps was done by Hauert et al. (2009) for the purposes of developing a flying communication network. In this case, the authors proposed to reverse engineer the behavior of an evolved neural network and subsequently program a similar behavior manually.

This approach provided original and “creative” insights that enabled them to design a viable and flexible behavior. In later work, Szabo (2015) applied evolutionary behavior trees to a team of MAVs for the purposes of collision avoidance, exploiting the increased readability of behavior trees. The MAVs only knew each other’s relative distance (not position) as measured by noisy Bluetooth signal strength, yet the evolved behavior was capable of reducing the number of collisions in a cluttered space. The automatically evolved behavior tree was not only simpler (fewer nodes/branches), but also performed better when compared to a manually designed one. Scheper and de Croon (2017) trained a neural network to form a triangle with a team of three MAVs, inspired by a similar task by Izzo et al. (2014). Although not aimed at MAVs, Izzo et al. (2014) had previously shown that an automatic method was able to extract a behavior with which homogeneous agents could self-organize into asymmetric patterns, whereas the previously developed manual approaches for the same system were limited to symmetric patterns (Izzo and Pettazzi, 2007). Scheper and de Croon (2017) additionally showed that evolving a controller at a higher level of abstraction does not necessarily compromise the ability of automatic methods to exploit an environment and sensory-motor relationships, yet helps to reduce the reality gap. The more recent work of Schilling et al. (2019) showed that it’s possible to learn a flocking behavior directly from camera images using imitation learning. This was demonstrated in a real world environment with two MAVs. This automatic approach was able to find a viable, collision-free behavior that could also localize neighbors.

The limited amount of works show that this field is still young. The extra challenge comes from the several constraints that flow from the lower levels as well as the additional cost and difficulty of real-world experimentation. Nevertheless, there are arguments to show that automatic methods may eventually provide a way to make the most out of the swarms (Francesca and Birattari, 2016). We expect that in the future, once both MAVs as well as automatic swarming design technologies become more mature, we will begin to see an increase of (experimental) works in this domain.

7. FURTHER CHALLENGES AND FUTURE DEVELOPMENTS TO BE MADE

7.1. Battery Recharging and Scheduling

As already discussed, flight time is a fundamental constraint for MAVs. Swarming can help to expand the flight time of the whole system, such that a portion of MAVs can recharge while others are still in operation. This is subject to two main challenges. The first is the design of the combined MAV + re-charging ecosystem, and the second is the distributed scheduling between drones. Research has already begun on this front, albeit to the best of our knowledge an automated and distributed recharging method for a swarm of MAVs has yet to be demonstrated outside of a controlled environment. Toksoz et al. (2011) and Lee et al. (2015) designed a battery swapping station to quickly exchange batteries on a quadrotor. The advantage of such a system is that the battery can be changed quickly. However, it also requires

an intricate design as well as highly accurate landing to ensure that the battery is properly replaced. Instead, a contact-based re-charging station, such as the one proposed by Leonard et al. (2014) offers a simpler system, albeit at the cost of a slower turn-over. The authors investigated its use for a multi-UAV system, whereby the MAVs queued their use of the charging stations via a prioritization function. Using a similar charging system, Mulgaonkar and Kumar (2014) demonstrated a system where three quadrotors take turns to surveil a target region, such that one operates while the other two recharge. Vasile and Belta (2014) and Leahy et al. (2016) proposed formal strategies based on temporal logic constraints to ensure that the MAVs would correctly queue for recharging. However, the experimental efforts focused on the case where only one MAV operates at a given time. Nowadays, commercial charging stations are also available (Brommer et al., 2018). This will likely accelerate the research progress. Wireless charging, albeit slower, is also an attractive choice as it softens the requirement on precision landing (Choi et al., 2016; Junaid et al., 2017).

Flight time can also be increased at the MAV design level by designing MAVs with on-board recharging or longer endurance. The capability for long endurance would allow the swarm to be more flexible and take on a more diverse set of missions. One possible method to increase the flight time is to use solar cells. These have mostly been applied to fixed wing designs, such as the “Skysailor” MAV (Noth and Siegwart, 2010), benefiting from efficient flight conditions and large wing areas. It can in fact be shown that the benefit of solar cells begins to have little effect on smaller platforms, due to the reduced surface area available (Bronz et al., 2009). This trend is even more prominent on quadrotors, which have higher energy requirements. As a solution, D’Sa et al. (2016) proposed an MAV design that can alternate between fixed wing and quadrotor mode, such that *“surplus energy collected and stored while in a fixed wing configuration is utilized while in a quadrotor configuration.”* Recently, Goh et al. (2019) demonstrated a fully solar-powered quadrotor. To meet the energy requirements, an area of 4 m² was required together with a reliance on ground-effects, meaning that the MAV was bound to low altitudes. A different solution is to use combustion engines (Zufferey et al., 2013; Nex and Remondino, 2014; Ross, 2014). They benefit from the high-energy density of fuel and can help to provide long endurance flight, although they are typically applied to larger drones in outdoor environments. Alternatively, fuel cells have also been explored as a power source for long endurance flight, with increasingly promising results in the recent years (Gong and Verstraete, 2017; De Wagter et al., 2019; Pan et al., 2019).

7.2. Swarm-Level Active Fault Detection

Active and decentralized fault detection should also play a fundamental role for the realization of MAV swarms¹². If not catered to, then there is a risk that the erroneous actions of one MAV hinder the entire swarm (Bjerknes and Winfield, 2013).

¹² We differentiate between fault *detection* and fault *tolerance*. Fault tolerance refers to the ability of the system to be robust to faults. Fault detection refers to the ability of the robots in the swarm to detect issues, and thus possibly also cope with them.

Winfield and Nembrini (2006) applied the Failure Mode and Effect Analysis (FMEA) methodology to evaluate the reliability of an entire swarm based on its possible failure points. From such studies it can be evaluated whether, and to what extent, local failures can incapacitate the swarm. The question is how such faults can be detected and dealt with during operations. Doing so would create a system that is more robust to failures.

Li and Parker (2007) developed the Sensor Analysis based Fault Detection (SAFDetection). In this approach, a clustering algorithm is used to learn a model of the robots' expected behavior. This model is then used to determine whether the behavior of a robot in the swarm can be considered "normal" (i.e., falls within the learned model), or "abnormal," in which case a likely fault has been detected. A distributed version of the algorithm has also been developed (Li and Parker, 2009), in which case each robot learns its own behavior model locally and then shares it. This strategy scales better with the size of the swarm, as it parallelizes the clustering computations. The works by Tarapore et al. (2013, 2015a,b) also propose a strategy for normal/abnormal behavior classification by synthesizing the behavior of neighbors within a binary feature vector. In more recent work, Tarapore et al. (2017) proposed the use of a consensus algorithm so that the robots can collectively reach a decision on whether the behavior of a team-member can be considered normal or abnormal. This was also tested on a real robotic system (Tarapore et al., 2019). Qin et al. (2014) provide a review on this active area of research. Bringing these solutions to MAV swarms can largely improve the operational safety of the full system, which is paramount for deployment in the real world.

7.3. Controlling and Supervising Swarms of MAVs

A control interface should enable an operator to provide commands to the swarm, such as take-off and landing, the commencement of mission objectives, or the engagement of swarm-wide emergency procedures. All should be done in a direct and intuitive way to minimize the effort by the operator (Fuchs et al., 2014; Dousse et al., 2016). To this end, Nagi et al. (2014) explored the use of a gesture vocabulary which allows a human operator to instruct a team of MAVs. The human operator and the gestures are detected directly by the MAVs using their on-board camera. Thanks to their multiple viewpoints, they are able to discern the operator's commands in a distributed fashion. Tsykunov et al. (2018) explored how to use a haptic glove to control a team of drones as if they were all connected via a spring-damper system. Research has also focused on the development of gesture languages, as in the works of Soto-Gerrero and Ramirez-Torres (2016) and Couture et al. (2018). Virtual reality is also becoming an increasingly popular technology, and is beginning to be applied to the control of MAVs (Tsykunov and Tsetserukou, 2019; Vempati et al., 2019). Besides the above, a less technical, yet highly significant, challenge to overcome on this front is the (understandably) stringent legislation surrounding MAV flight, particularly in outdoor scenarios, often requiring at least one pilot per drone (the specifics vary based on the location) (Vincenzi et al., 2015). We refer the interested reader to Hocraffer

and Nam (2017) and the sources therein for a more thorough overview of the challenges and the current technologies for human control of aerial swarms.

8. DISCUSSION: HOW FAR ARE WE FROM MAV SWARMS?

Following the many topics discussed in this paper comes the inevitable question: how far are we from large scale aerial swarms that can cooperatively explore areas, carry heavier objects, and autonomously complete complex tasks without low level human-in-the-loop control? Despite the large amount of research and development that has been done to tackle the topics within this grander scheme, the field of robotics and the field of swarm intelligence are both still relatively young, and there remain advances to be made. In this paper, we discussed how the swarm behavior depends on the constraints set by lower level properties, and vice versa. This interdependency and iterative nature of design means that, if we wish to bring full-fledged MAV swarms to the real world, there must be a mutual understanding between the design levels as to what is required and what can be achieved in reality.

One of the main technologies required to make the leap from flying a single MAV to flying a decentralized swarm is an accurate and reliable intra-swarm relative localization technology. Even for those applications where cooperation is limited and each member in the swarm acts mostly independently, relative localization is still needed to ensure relative collision avoidance, which is a safety-critical requirement. As we have shown throughout this paper, several technologies are currently under exploration and it is still unclear which will prove most reliable and advantageous in the long run. As the choice of these systems very directly shapes the behavior of the swarm, the challenge of designing the swarm behavior needs to be tightly coupled to it, additionally to the way it is coupled to the design of the individual robots. As such, automatic design algorithms of swarm behaviors can provide a way to make the most out of the individual MAVs and their limitations, albeit at the potential cost of relying on less well-understood controllers.

Additionally, the on-going standardization of tools is expected to help the field to reach a new level of maturity (Nedjah and Junior, 2019). Systems, such as ROS (Quigley et al., 2009), Paparazzi (Mueller and Drouin, 2007; Brisset and Hattenberger, 2008), or PX4 (Meier et al., 2015) have now accelerated the process of prototyping and testing on real-world MAVs, and have also made it easier to share hardware/software advancements. Low cost programmable MAVs, such as the Crazyflie are also available, making it more feasible to experiment with large numbers of MAVs. Additionally, dedicated standards, such as MAVLink, which provides communication between software modules, are becoming increasingly popular (Dietrich et al., 2016), and full-stack frameworks have been developed to handle the entire pipeline (Sanchez-Lopez et al., 2016; Millan-Romera et al., 2019). The combination of these systems together with simulators, such as the well-known Gazebo (Koenig and Howard, 2004), ARGoS (Pinciroli et al., 2012), or AirSim (Shah

et al., 2018), further help to quickly prototype software in a realistic simulation environment. Combined with models and frameworks, such as hector-quadrotor (Meyer et al., 2012) or RotorS (Furrer et al., 2016), simulation environments can significantly accelerate the development time (Johnson and Mishra, 2002). Mairaj et al. (2019) provides an extensive review of several simulators for this purpose. Dedicated swarm languages, such as Buzz (Pinciroli and Beltrame, 2016) also provide a simpler prototyping framework dedicated to swarm robotics, which can also be applied to MAVs.

Finally, the prominent rise in popularity of MAVs in the last decade has brought about several technology accelerators. MAV focused robotics competitions, such as the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) or the International Micro Air Vehicle (IMAV) competition have now also begun to integrate swarming or multi-robot elements (Pestana et al., 2014; Saska et al., 2016a; Bähmann et al., 2017; Nieuwenhuisen et al., 2017; Spurný et al., 2019). This pushes researchers to take a technology out of the lab and into unknown environments, thereby increasing their robustness.

REFERENCES

- Abeywardena, D., Kodagoda, S., Dissanayake, G., and Munasinghe, R. (2013). Improved state estimation in quadrotor MAVs: a novel drift-free velocity estimator. *IEEE Robot. Autom. Mag.* 20, 32–39. doi: 10.1109/MRA.2012.2225472
- Achtelik, M., Achtelik, M., Brunet, Y., Chli, M., Chatzichristofis, S., Decotignie, J., et al. (2012). “SFly: swarm of micro flying robots,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vilamoura), 2649–2650. doi: 10.1109/IROS.2012.6386281
- Achtelik, M. W., Weiss, S., Chli, M., Dellaert, F., and Siegwart, R. (2011). “Collaborative stereo,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (San Francisco, CA), 2242–2248. doi: 10.1109/IROS.2011.6094866
- Afzal, M. H., Renaudin, V., and Lachapelle, G. (2011). “Magnetic field based heading estimation for pedestrian navigation environments,” in *2011 International Conference on Indoor Positioning and Indoor Navigation* (Guimaraes), 1–10. doi: 10.1109/IPIN.2011.6071947
- Aguilar, W. G., Casaliglla, V. P., and Pólit, J. L. (2017). “Obstacle avoidance for low-cost UAVs,” in *2017 IEEE 11th International Conference on Semantic Computing* (ICSC) (San Diego, CA), 503–508. doi: 10.1109/ICSC.2017.96
- Alexis, K., Nikolakopoulos, G., Tzes, A., and Dritsas, L. (2009). “Coordination of helicopter UAVs for aerial forest-fire surveillance,” in *Applications of Intelligent Control to Engineering Systems*, ed K. P. Valavanis (Dordrecht: Springer Netherlands), 169–193.
- Allred, J., Hasan, A. B., Panichsakul, S., Pisano, W., Gray, P., Huang, J., et al. (2007). “Sensorflock: an airborne wireless sensor network of micro-air vehicles,” in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems, SenSys '07* (New York, NY: Association for Computing Machinery), 117–129.
- Alonso-Mora, J., Nägeli, T., Siegwart, R., and Beardsley, P. (2015). Collision avoidance for aerial vehicles in multi-agent scenarios. *Auton. Robots* 39, 101–121. doi: 10.1007/s10514-015-9429-0
- Alvarez, H., Paz, L. M., Sturm, J., and Cremers, D. (2016). *Collision Avoidance for Quadrotors with a Monocular Camera*. Cham: Springer International Publishing, 195–209.
- Ampatzis, C., Tuci, E., Trianni, V., and Dorigo, M. (2008). Evolution of signaling in a multi-robot system: categorization and communication. *Adapt. Behav.* 16, 5–26. doi: 10.1177/1059712307087282
- Aoki, I. (1982). A simulation study on the schooling mechanism in fish. *Nippon Suisan Gakkaishi* 48, 1081–1088.
- Augugliaro, F., Lupashin, S., Hamer, M., Male, C., Hehn, M., Mueller, M. W., et al. (2014). The flight assembled architecture installation: cooperative construction with flying machines. *IEEE Control Syst. Mag.* 34, 46–64. doi: 10.1109/MCS.2014.2320359
- Bachrach, A., Prentice, S., He, R., and Roy, N. (2011). RANGE-robust autonomous navigation in GPS-denied environments. *J. Field Robot.* 28, 644–666. doi: 10.1002/rob.20400
- Bähmann, R., Schindler, D., Kamel, M., Siegwart, R., and Nieto, J. (2017). “A decentralized multi-agent unmanned aerial system to search, pick up, and relocate objects,” in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)* (Shanghai), 123–128. doi: 10.1109/SSRR.2017.8088150
- Basiri, M. (2015). *Audio-based positioning and target localization for swarms of micro aerial vehicles* (Ph.D. thesis), École polytechnique fédérale de Lausanne, Lausanne. doi: 10.5075/epfl-thesis-6508
- Basiri, M., Schill, F., Floreano, D., and Lima, P. U. (2014). “Audio-based localization for swarms of micro air vehicles,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 4729–4734. doi: 10.1109/ICRA.2014.6907551
- Basiri, M., Schill, F., Lima, P., and Floreano, D. (2016). On-board relative bearing estimation for teams of drones using sound. *IEEE Robot. Autom. Lett.* 1, 820–827. doi: 10.1109/LRA.2016.2527833
- Beard, R. W. (2007). “State estimation for micro air vehicles,” in *Innovations in Intelligent Machines-1*, eds J. S. Chahl, L. C. Jain, A. Mizutani, and M. Sato-Ilic (Berlin; Heidelberg: Springer Berlin Heidelberg), 173–199.
- Bedau, M. A. (2003). Artificial life: organization, adaptation and complexity from the bottom up. *Trends Cogn. Sci.* 7, 505–512. doi: 10.1016/j.tics.2003.09.012
- Bensky, A. (2019). *Short-Range Wireless Communication*. Cambridge, MA: Elsevier. doi: 10.1016/c2017-0-02356-x
- Beyeler, A., Zufferey, J.-C., and Floreano, D. (2009). Vision-based control of near-obstacle flight. *Auton. Robots* 27:201. doi: 10.1007/s10514-009-9139-6
- Bitcraze, A. B. (2019). *Multi-Ranger Deck*. Available online at: <https://www.bitcraze.io/multi-ranger-deck/>
- Bjerknes, J. D., and Winfield, A. F. T. (2013). *On Fault Tolerance and Scalability of Swarm Robotic Systems*. Berlin; Heidelberg: Springer Berlin Heidelberg, 431–444.
- Bonabeau, E., and Théraulaz, G. (2008). Swarm smarts. *Sci. Am.* 18, 40–47. doi: 10.1038/scientificamerican0208-40sp

9. CONCLUSION

The challenges to solve before we can expect to see swarms of autonomous MAVs are many. They begin at the lowest level, forcing us to think of how the MAV design will impact the swarm behavior, and they end at the highest level, where we must design collective behaviors that best exploit our lower level designs, controllers, and sensors. In the last decade, the field of swarm robotics and MAV design have started to merge more and more, leading to increasingly impressive achievements. To go further, the tight and complex relationship between the low level and the high level needs to be appreciated in order to break into a new era of truly autonomous and distributed swarms of MAVs.

AUTHOR CONTRIBUTIONS

This article has been set up based on discussions between all authors. It has been primarily written by MC with the additional help and advice of KM, CD, and GC.

- Bouabdallah, S., Murrieri, P., and Siegwart, R. (2004). "Design and control of an indoor micro quadrotor," in *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 5 (New Orleans, LA), 4393–4398. doi: 10.1109/ROBOT.2004.1302409
- Bouabdallah, S., and Siegwart, R. (2007). "Full control of a quadrotor," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (San Diego, CA), 153–158. doi: 10.1109/IROS.2007.4399042
- Bouffanais, R. (2016). *Design and Control of Swarm Dynamics*. Singapore: Springer Singapore.
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* 7, 1–41. doi: 10.1007/s11721-012-0075-2
- Bredeche, N., Haasdijk, E., and Prieto, A. (2018). Embodied evolution in collective robotics: a review. *Front. Robot. AI* 5:12. doi: 10.3389/frobt.2018.00012
- Briod, A., Klapotcz, A., Zufferey, J., and Floreano, D. (2012). "The AirBurr: a flying robot that can exploit collisions," in *2012 ICME International Conference on Complex Medical Engineering (CME)* (Kobe), 569–574. doi: 10.1109/ICME.2012.6275674
- Briod, A., Zufferey, J.-C., and Floreano, D. (2013). "Optic-flow based control of a 46g quadrotor," in *Workshop on Vision-Based Closed-Loop Control and Navigation of Micro Helicopters in GPS-Denied Environments, IROS 2013* (Tokyo).
- Brisset, P., and Hattenberger, G. (2008). "Multi-UAV control with the paparazzi system," in *Conference on Humans Operating Unmanned Systems (HUMOUS)* (Brest).
- Brommer, C., Malyuta, D., Hentzen, D., and Brockers, R. (2018). "Long-duration autonomy for small rotorcraft uas including recharging," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid), 7252–7258. doi: 10.1109/IROS.2018.8594111
- Bronz, M., Moschetta, J. M., Brisset, P., and Gorraz, M. (2009). Towards a long endurance MAV. *Int. J. Micro Air Vehicles* 1, 241–254. doi: 10.1260/175682909790291483
- Brutschy, A., Garattoni, L., Brambilla, M., Francesca, G., Pini, G., Dorigo, M., et al. (2015). The TAM: abstracting complex tasks in swarm robotics research. *Swarm Intell.* 9, 1–22. doi: 10.1007/s11721-014-0102-6
- Bry, A., Richter, C., Bachrach, A., and Roy, N. (2015). Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments. *Int. J. Robot. Res.* 34, 969–1002. doi: 10.1177/0278364914558129
- Busoni, L., Babuska, R., and De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst. Man Cybernet. C Appl. Rev.* 38, 156–172. doi: 10.1109/TSMCC.2007.913919
- Cabrera-Ponce, A. A., Martinez-Carranza, J., and Rascon, C. (2019). "Detection of nearby UAVs using CNN and spectrograms," in *International Micro Air Vehicle Conference and Competition (IMAV)* (Madrid).
- Cadell, C. (2018). *Flight of Imagination: Chinese Firm Breaks Record With 1,374 Dancing Drones*. Reuters. Available online at: www.reuters.com/article/us-china-drones/flight-of-imagination-chinese-firm-breaks-record-with-1374-dancing-drones-idUSKBN113189
- Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., et al. (2016). Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. *IEEE Trans. Robot.* 32, 1309–1332. doi: 10.1109/TRO.2016.2624754
- Campion, M., Ranganathan, P., and Faruque, S. (2018). "A review and future directions of UAV swarm communication architectures," in *2018 IEEE International Conference on Electro/Information Technology (EIT)* (Rochester, MI), 0903–0908. doi: 10.1109/EIT.2018.8500274
- Carrio, A., Bavle, H., and Campoy, P. (2018). Attitude estimation using horizon detection in thermal images. *Int. J. Micro Air Vehicles* 10, 352–361. doi: 10.1177/1756829318804761
- Carrio, A., Vemprala, S., Ripoll, A., Saripalli, S., and Campoy, P. (2018). "Drone detection using depth maps," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid), 1034–1037. doi: 10.1109/IROS.2018.8593405
- Chamanbaz, M., Mateo, D., Zoss, B. M., Tokić, G., Wilhelm, E., Bouffanais, R., et al. (2017). Swarm-enabling technology for multi-robot systems. *Front. Robot. AI* 4:12. doi: 10.3389/frobt.2017.00012
- Chen, Y., Zhao, H., Mao, J., Chirarattananon, P., Helbling, E. F., Hyun, N.-S. P., et al. (2019). Controlled flight of a microrobot powered by soft artificial muscles. *Nature* 575, 324–329. doi: 10.1038/s41586-019-1737-7
- Choi, C. H., Jang, H. J., Lim, S. G., Lim, H. C., Cho, S. H., and Gaponov, I. (2016). "Automatic wireless drone charging station creating essential environment for continuous drone operation," in *2016 International Conference on Control, Automation and Information Sciences (ICCAIS)* (Ansan), 132–136. doi: 10.1109/ICCAIS.2016.7822448
- Chung, S., Paranjape, A. A., Dames, P., Shen, S., and Kumar, V. (2018). A survey on aerial swarm robotics. *IEEE Trans. Robot.* 34, 837–855. doi: 10.1109/TRO.2018.2857475
- Chung, T. H., Clement, M. R., Day, M. A., Jones, K. D., Davis, D., and Jones, M. (2016). "Live-fly, large-scale field experimentation for large numbers of fixed-wing UAVs," in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (Stockholm), 1255–1262. doi: 10.1109/ICRA.2016.7487257
- Cieslewski, T., Choudhary, S., and Scaramuzza, D. (2018). "Data-efficient decentralized visual slam," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD), 2466–2473. doi: 10.1109/ICRA.2018.8461155
- Cieslewski, T., and Scaramuzza, D. (2017). "Efficient decentralized visual place recognition from full-image descriptors," in *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)* (Los Angeles, CA), 78–82. doi: 10.1109/MRS.2017.8250934
- Cognetti, M., Stegagno, P., Franchi, A., Oriolo, G., and Bühlhoff, H. H. (2012). "3-D mutual localization with anonymous bearing measurements," in *2012 IEEE International Conference on Robotics and Automation* (Saint Paul, MN), 791–798. doi: 10.1109/ICRA.2012.6225288
- Collotta, M., Pau, G., Talty, T., and Tonguz, O. K. (2018). Bluetooth 5: a concrete step forward toward the IoT. *IEEE Commun. Mag.* 56, 125–131. doi: 10.1109/MCOM.2018.1700053
- Coppola, M., Guo, J., Gill, E., and de Croon, G. C. H. E. (2019). The pagerank algorithm as a method to optimize swarm behavior through local analysis. *Swarm Intell.* 13, 277–319. doi: 10.1007/s11721-019-00172-z
- Coppola, M., McGuire, K. N., Scheper, K. Y. W., and de Croon, G. C. H. E. (2018). On-board communication-based relative localization for collision avoidance in micro air vehicle teams. *Auton. Robots* 42, 1787–1805. doi: 10.1007/s10514-018-9760-3
- Couture, N., Bottechia, S., Chaumette, S., Cecconello, M., Rekalde, J., and Desainte-Catherine, M. (2018). "Using the soundpainting language to fly a swarm of drones," in *Advances in Human Factors in Robots and Unmanned Systems*, ed J. Chen (Cham: Springer International Publishing), 39–51.
- Cunningham, A., Indelman, V., and Dellaert, F. (2013). "Ddf-sam 2.0: consistent distributed smoothing and mapping," in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe), 5220–5227. doi: 10.1109/ICRA.2013.6631323
- de Croon, G. C. H. E., de Weerd, E., De Wagter, C., Remes, B. D. W., and Ruijsink, R. (2012). The appearance variation cue for obstacle avoidance. *IEEE Trans. Robot.* 28, 529–534. doi: 10.1109/TRO.2011.2170754
- de Croon, G. C. H. E., Perçin, M., Remes, B. D. W., Ruijsink, R., and Wagter, C. D. (2016). *The DelFly*. Dordrecht: Springer Netherlands. doi: 10.1007/978-94-017-9208-0
- de Croon, G. C. H. E., Van Dartel, M. F., and Postma, E. O. (2005). "Evolutionary learning outperforms reinforcement learning on non-markovian tasks," in *Workshop on Memory and Learning Mechanisms in Autonomous Robots, 8th European Conference on Artificial Life* (Canterbury).
- de Croon, G. C. H. E., Wagter, C. D., Remes, B. D. W., and Ruijsink, R. (2012). Sub-sampling: real-time vision for micro air vehicles. *Robot. Auton. Syst.* 60, 167–181. doi: 10.1016/j.robot.2011.10.001
- de Marina, H. G., and Smeur, E. (2019). "Flexible collaborative transportation by a team of rotorcraft," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC), 1074–1080. doi: 10.1109/ICRA.2019.8794316
- de Marina, H. G., Sun, Z., Bronz, M., and Hattenberger, G. (2017). "Circular formation control of fixed-wing UAVs with constant speeds," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, BC), 5298–5303. doi: 10.1109/IROS.2017.8206422

- De Wagter, C., Remes, B., Ruisink, R., van Tienen, F., and van der Horst, E. (2019). "Design and testing of a vertical take-off and landing uav optimized for carrying a hydrogen fuel-cell with pressure tank," in *International Micro Air Vehicle Conference and Competition (IMAV)* (Madrid).
- Dietrich, T., Andryeyev, O., Zimmermann, A., and Mitschele-Thiel, A. (2016). "Towards a unified decentralized swarm management and maintenance coordination based on MAVLink," in *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (Braganca), 124–129. doi: 10.1109/ICARSC.2016.64
- Doer, C., Scholz, G., and Trommer, G. F. (2017). Indoor laser-based slam for micro aerial vehicles. *Gyros. Navig.* 8, 181–189. doi: 10.1134/S2075108717030038
- Dousse, N., Heitz, G., and Floreano, D. (2016). Extension of a ground control interface for swarms of small drones. *Artif. Life Robot.* 21, 308–316. doi: 10.1007/s10015-016-0302-9
- D'Sa, R., Jenson, D., Henderson, T., Kilian, J., Schulz, B., Calvert, M., et al. (2016). "SUAV:Q—an improved design for a transformable solar-powered UAV," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 1609–1615. doi: 10.1109/IROS.2016.7759260
- Duarte, M., Costa, V., Gomes, J., Rodrigues, T., Silva, F., Oliveira, S. M., et al. (2016). Evolution of collective behaviors for a real swarm of aquatic surface robots. *PLoS ONE* 11:e151834. doi: 10.1371/journal.pone.0151834
- Dupeyroux, J., Serres, J. R., and Viollet, S. (2019). AntBot: a six-legged walking robot able to home like desert ants in outdoor environments. *Sci. Robot.* 4:eau0307. doi: 10.1126/scirobotics.aau0307
- Dusha, D., Mejias, L., and Walker, R. (2011). Fixed-wing attitude estimation using temporal tracking of the horizon and optical flow. *J. Field Robot.* 28, 355–372. doi: 10.1002/rob.20387
- Eiben, A. E. (2014). Grand challenges for evolutionary robotics. *Front. Robot. AI* 1:4. doi: 10.3389/frobt.2014.00004
- Engelen, S., Gill, E., and Verhoeven, C. (2014). On the reliability, availability, and throughput of satellite swarms. *IEEE Trans. Aerospace Electron. Syst.* 50, 1027–1037. doi: 10.1109/TAES.2014.120711
- Epstein, D., and Feldman, D. (2018). Quadcopter tracks quadcopter via real-time shape fitting. *IEEE Robot. Autom. Lett.* 3, 544–550. doi: 10.1109/LRA.2017.2773668
- Faessler, M., Fontana, F., Forster, C., and Scaramuzza, D. (2015). "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA), 1722–1729. doi: 10.1109/ICRA.2015.7139420
- Faessler, M., Mueggler, E., Schwabe, K., and Scaramuzza, D. (2014). "A monocular pose estimation system based on infrared leds," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 907–913. doi: 10.1109/ICRA.2014.6906962
- Faigl, J., Krajník, T., Chudoba, J., Pfeučil, L., and Saska, M. (2013). "Low-cost embedded system for relative localization in robotic swarms," in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe), 993–998. doi: 10.1109/ICRA.2013.6630694
- Falanga, D., Kim, S., and Scaramuzza, D. (2019a). How fast is too fast? The role of perception latency in high-speed sense and avoid. *IEEE Robot. Autom. Lett.* 4, 1884–1891. doi: 10.1109/LRA.2019.2898117
- Falanga, D., Kleber, K., Mintchev, S., Floreano, D., and Scaramuzza, D. (2019b). The foldable drone: a morphing quadrotor that can squeeze and fly. *IEEE Robot. Autom. Lett.* 4, 209–216. doi: 10.1109/LRA.2018.2885575
- Faludi, R. (2010). *Building Wireless Sensor Networks: With ZigBee, XBee, Arduino, and Processing*. Sebastopol, CA: O'Reilly Media, Inc.
- Ferrante, E., Duéñez Guzmán, E., Turgut, A. E., and Wenseleers, T. (2013). "GESwarm: grammatical evolution for the automatic synthesis of collective behaviors in swarm robotics," in *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13* (New York, NY: ACM), 17–24.
- Fiorini, P., and Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *Int. J. Robot. Res.* 17, 760–772.
- Flocchini, P., Prencipe, G., Santoro, N., and Widmayer, P. (2008). Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. *Theor. Comput. Sci.* 407, 412–447. doi: 10.1016/j.tcs.2008.07.026
- Floreano, D., Mintchev, S., and Shintake, J. (2017). "Foldable drones: from biology to technology," in *Bioinspiration, Biomimetics, and Bioreplication 2017*, Vol. 10162, eds M. Knez, A. Lakhtakia, and R. J. Martin-Palma (Portland, OR: International Society for Optics and Photonics, SPIE), 1–6. doi: 10.1117/12.2259931
- Floreano, D., Pericet-Camara, R., Viollet, S., Ruffier, F., Brückner, A., Leitl, R., et al. (2013). Miniature curved artificial compound eyes. *Proc. Natl. Acad. Sci. U.S.A.* 110, 9267–9272. doi: 10.1073/pnas.1219068110
- Floreano, D., and Wood, R. J. (2015). Science, technology and the future of small autonomous drones. *Nature* 521:460. doi: 10.1038/nature14542
- Forster, C., Lynen, S., Kneip, L., and Scaramuzza, D. (2013). "Collaborative monocular slam with multiple micro aerial vehicles," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Tokyo), 3962–3970. doi: 10.1109/IROS.2013.6696923
- Francesca, G., and Birattari, M. (2016). Automatic design of robot swarms: achievements and challenges. *Front. Robot. AI* 3:29. doi: 10.3389/frobt.2016.00029
- Francesca, G., Brambilla, M., Brutschy, A., Garattoni, L., Miletitch, R., Podevijn, G., et al. (2015). Automode-chocolate: automatic design of control software for robot swarms. *Swarm Intell.* 9, 125–152. doi: 10.1007/s11721-015-0107-9
- Francesca, G., Brambilla, M., Brutschy, A., Trianni, V., and Birattari, M. (2014). Automode: a novel approach to the automatic design of control software for robot swarms. *Swarm Intell.* 8, 89–112. doi: 10.1007/s11721-014-0092-4
- Franchi, A., Oriolo, G., and Stegagno, P. (2013). Mutual localization in multi-robot systems using anonymous relative measurements. *Int. J. Robot. Res.* 32, 1302–1322. doi: 10.1177/0278364913495425
- Fuchs, C., Borst, C., de Croon, G. C. H. E., van Paassen, M. M. R., and Mulder, M. (2014). An ecological approach to the supervisory control of UAV swarms. *Int. J. Micro Air Vehicles* 6, 211–229. doi: 10.1260/1756-8293.6.4.211
- Furrer, F., Burri, M., Achtelik, M., and Siegwart, R. (2016). *RotorS—A Modular Gazebo MAV Simulator Framework*. Cham: Springer International Publishing, 595–625.
- Gabrich, B., Saldaña, D., Kumar, V., and Yim, M. (2018). "A flying gripper based on cuboid modular robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD), 7024–7030. doi: 10.1109/ICRA.2018.8460682
- Gandhi, D., Pinto, L., and Gupta, A. (2017). "Learning to fly by crashing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, BC), 3948–3955. doi: 10.1109/IROS.2017.8206247
- Garnier, S., Gautrais, J., and Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm Intell.* 1, 3–31. doi: 10.1007/s11721-007-0004-y
- Gazi, V. (2005). Swarm aggregations using artificial potentials and sliding-mode control. *IEEE Trans. Robot.* 21, 1208–1214. doi: 10.1109/TRO.2005.853487
- Gazi, V., and Passino, K. M. (2002). "A class of attraction/repulsion functions for stable swarm aggregations," in *Proceedings of the 41st IEEE Conference on Decision and Control* Vol. 3 (Las Vegas, NV), 2842–2847. doi: 10.1109/CDC.2002.1184277
- Gazi, V., and Passino, K. M. (2004). Stability analysis of social foraging swarms. *IEEE Trans. Syst. Man Cybernet. B Cybernet.* 34, 539–557. doi: 10.1109/TSMCB.2003.817077
- Gazi, V., and Passino, K. M. (2011). *Swarm Stability and Optimization*. Berlin; Heidelberg: Springer. doi: 10.1007/978-3-642-18041-5
- Ghadiok, V., Goldin, J., and Ren, W. (2012). On the design and development of attitude stabilization, vision-based navigation, and aerial gripping for a low-cost quadrotor. *Auton. Robots* 33, 41–68. doi: 10.1007/s10514-012-9286-z
- Giernacki, W., Skwarczyński, M., Witwicki, W., Wroński, P., and Kozierski, P. (2017). "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *22nd International Conference on Methods and Models in Automation and Robotics (MMAR)* (Miedzyzdroje), 37–42. doi: 10.1109/MMAR.2017.8046794
- Goh, C. S., Kuan, J. R., Yeo, J. H., Teo, B. S., and Danner, A. (2019). A fully solar-powered quadcopter able to achieve controlled flight out of the ground effect. *Prog. Photovolt. Res. Appl.* 27, 869–878. doi: 10.1002/ppp.3169
- Gomes, J., Urbano, P., and Christensen, A. L. (2013). Evolution of swarm robotics systems with novelty search. *Swarm Intell.* 7, 115–144. doi: 10.1007/s11721-013-0081-z

- Gong, A., and Verstraete, D. (2017). Fuel cell propulsion in small fixed-wing unmanned aerial vehicles: Current status and research needs. *Int. J. Hydrogen Energy* 42, 21311–21333. doi: 10.1016/j.ijhydene.2017.06.148
- Green, W. E., and Oh, P. Y. (2006). “Autonomous hovering of a fixed-wing micro air vehicle,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA)* (Orlando, FL), 2164–2169. doi: 10.1109/ROBOT.2006.1642024
- Grondin, F., Létourneau, D., Ferland, F., Rousseau, V., and Michaud, F. (2013). The ManyEars open framework. *Auton. Robots* 34, 217–232. doi: 10.1007/s10514-012-9316-x
- Grzonka, S., Grisetti, G., and Burgard, W. (2009). “Towards a navigation system for autonomous indoor flying,” in *2009 IEEE International Conference on Robotics and Automation* (Kobe), 2878–2883. doi: 10.1109/ROBOT.2009.5152446
- Guinness World Records (2018). *Most Unmanned Aerial Vehicles (UAVs) Airborne Simultaneously*. Available online at: www.guinnessworldrecords.com/world-records/373319-most-C2%A0unmanned-aerial-C2%A0vehicles-C2%A0uavs-airborne-C2%A0simultaneously-C2%A05-kg-or-less
- Guinness World Records (2019). *Most Unmanned Aerial Vehicles (UAVs) Airborne Simultaneously From a Single Computer (Indoors)*. Available online at: www.guinnessworldrecords.com/world-records/507534-most-uavs-controlled-from-a-single-computer
- Guo, K., Qiu, Z., Meng, W., Xie, L., and Teo, R. (2017). Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in GPS denied environments. *Int. J. Micro Air Vehicles* 9, 169–186. doi: 10.1177/1756829317695564
- Guo, K., Qiu, Z., Miao, C., Zaini, A. H., Chen, C.-L., Meng, W., et al. (2016). Ultra-wideband-based localization for quadcopter navigation. *Unmanned Syst.* 4, 23–34. doi: 10.1142/S2301385016400033
- Gupte, S., Mohandas, P. I. T., and Conrad, J. M. (2012). “A survey of quadrotor unmanned aerial vehicles,” in *Proceedings of IEEE Southeastcon* (Orlando, FL: IEEE), 1–6. doi: 10.1109/secon.2012.6196930
- Guzzi, J., Giusti, A., Gambardella, L. M., and Di Caro, G. A. (2013a). “Local reactive robot navigation: a comparison between reciprocal velocity obstacle variants and human-like behavior,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Tokyo), 2622–2629. doi: 10.1109/IROS.2013.6696726
- Guzzi, J., Giusti, A., Gambardella, L. M., and Di Caro, G. A. (2014). “Bioinspired obstacle avoidance algorithms for robot swarms,” in *Bio-Inspired Models of Network, Information, and Computing Systems*, eds G. A. Di Caro and G. Theraulaz (Cham: Springer International Publishing), 120–134.
- Guzzi, J., Giusti, A., Gambardella, L. M., Theraulaz, G., and Di Caro, G. A. (2013b). “Human-friendly robot navigation in dynamic environments,” in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe), 423–430. doi: 10.1109/ICRA.2013.6630610
- Hamann, H. (2018). *Swarm Robotics: A Formal Approach*. Cham: Springer International Publishing. doi: 10.1007/978-3-319-74528-2
- Hauert, S., Leven, S., Varga, M., Ruini, F., Cangelosi, A., Zufferey, J., et al. (2011). “Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (San Francisco, CA), 5015–5020. doi: 10.1109/IROS.2011.6095129
- Hauert, S., Zufferey, J.-C., and Floreano, D. (2009). Evolved swarming without positioning information: an application in aerial communication relay. *Auton. Robots* 26, 21–32. doi: 10.1007/s10514-008-9104-9
- Heng, L., Meier, L., Tanskanen, P., Fraundorfer, F., and Pollefeys, M. (2011). “Autonomous obstacle avoidance and maneuvering on a vision-guided map using on-board processing,” in *2011 IEEE International Conference on Robotics and Automation* (Shanghai), 2472–2477. doi: 10.1109/ICRA.2011.5980095
- Ho, H. W., de Croon, G. C. H. E., and Chu, Q. (2017). Distance and velocity estimation using optical flow from a monocular camera. *Int. J. Micro Air Vehicles* 9, 198–208. doi: 10.1177/1756829317695566
- Hocraffer, A., and Nam, C. S. (2017). A meta-analysis of human-system interfaces in unmanned aerial vehicle (UAV) swarm management. *Appl. Ergonom.* 58, 66–80. doi: 10.1016/j.apergo.2016.05.011
- Honegger, D., Meier, L., Tanskanen, P., and Pollefeys, M. (2013). “An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications,” in *2013 IEEE International Conference on Robotics and Automation (ICRA)* (Karlsruhe), 1736–1741. doi: 10.1109/ICRA.2013.6630805
- Huang, A. S., Bachrach, A., Henry, P., Krainin, M., Maturana, D., Fox, D., et al. (2017). *Visual Odometry and Mapping for Autonomous Flight Using an RGB-D Camera*. Cham: Springer International Publishing, 235–252.
- Hüttenrauch, M., Šošić, A., and Neumann, G. (2019). Deep reinforcement learning for swarm systems. *J. Mach. Learn. Res.* 20, 1–31.
- Ishii, S., Fujita, H., Mitsutake, M., Yamazaki, T., Matsuda, J., and Matsuno, Y. (2005). A reinforcement learning scheme for a partially-observable multi-agent game. *Mach. Learn.* 59, 31–54. doi: 10.1007/s10994-005-0461-8
- Itasse, M., Moschetta, J.-M., Ameho, Y., and Carr, R. (2011). Equilibrium transition study for a hybrid MAV. *Int. J. Micro Air Vehicles* 3, 229–245. doi: 10.1260/1756-8293.3.4.229
- Izzo, D., and Pettazzi, L. (2007). Autonomous and distributed motion planning for satellite swarm. *J. Guidance Control Dyn.* 30, 449–459. doi: 10.2514/1.22736
- Izzo, D., Simões, L. F., and de Croon, G. C. H. E. (2014). An evolutionary robotics approach for the distributed control of satellite formations. *Evol. Intell.* 7, 107–118. doi: 10.1007/s12065-014-0111-9
- Johnson, E., and Mishra, S. (2002). “Flight simulation for the development of an experimental UAV,” in *AIAA Modeling and Simulation Technologies Conference and Exhibit* (Monterey, CA). doi: 10.2514/6.2002-4975
- Jones, S., Studley, M., Hauert, S., and Winfield, A. (2018). *Evolving Behaviour Trees for Swarm Robotics*. Cham: Springer International Publishing, 487–501.
- Jones, S., Winfield, A. F., Hauert, S., and Studley, M. (2019). Onboard evolution of understandable swarm behaviors. *Adv. Intell. Syst.* 1:1900031. doi: 10.1002/aisy.201900031
- Junaid, A. B., Konoiko, A., Zweiri, Y., Sahinkaya, M. N., and Seneviratne, L. (2017). Autonomous wireless self-charging for multi-rotor unmanned aerial vehicles. *Energies* 10:803. doi: 10.3390/en10060803
- Jung, S., Liu, C., and Ariyur, K. B. (2013). “Absolute orientation for a uav using celestial objects,” in *AIAA Infotech@Aerospace (I@A) Conference* (Boston, MA).
- Karásek, M., Muijres, F. T., De Wagter, C., Remes, B. D. W., and de Croon, G. C. H. E. (2018). A tailless aerial robotic flapper reveals that flies use torque coupling in rapid banked turns. *Science* 361, 1089–1094. doi: 10.1126/science.aat0350
- Kendoul, F., Fantoni, I., and Nonami, K. (2009a). Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles. *Robot. Auton. Syst.* 57, 591–602. doi: 10.1016/j.robot.2009.02.001
- Kendoul, F., Nonami, K., Fantoni, I., and Lozano, R. (2009b). An adaptive vision-based autopilot for mini flying machines guidance, navigation and control. *Auton. Robots* 27, 165–188. doi: 10.1007/s10514-009-9135-x
- Koenig, N., and Howard, A. (2004). “Design and use paradigms for Gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE Cat. No.04CH37566), Vol. 3 (Sendai), 2149–2154. doi: 10.1109/IROS.2004.1389727
- Kong, L., Sheng, J., and Teredesai, A. (2014). “Basic micro-aerial vehicles (MAVs) obstacles avoidance using monocular computer vision,” in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)* (Singapore), 1051–1056. doi: 10.1109/ICARCV.2014.7064451
- Kornatowski, P. M., Mintchev, S., and Floreano, D. (2017). “An origami-inspired cargo drone,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, BC), 6855–6862. doi: 10.1109/IROS.2017.8206607
- Krajník, T., Nitsche, M., Faigl, J., Vaněk, P., Saska, M., Pfeučil, L., et al. (2014). A practical multirobot localization system. *J. Intell. Robot. Syst.* 76, 539–562. doi: 10.1007/s10846-014-0041-x
- Krajník, T., Vonásek, V., Fišer, D., and Faigl, J. (2011). “AR-Drone as a platform for robotic research and education,” in *Research and Education in Robotics-EUROBOT 2011*, eds D. Obdržálek and A. Gottscheber (Berlin; Heidelberg: Springer Berlin Heidelberg), 172–186.
- Kuckling, J., Ligot, A., Bozhinoski, D., and Birattari, M. (2018). “Behavior trees as a control architecture in the automatic modular design of robot swarms,” in *Swarm Intelligence*, eds M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. Reina, and V. Trianni (Cham: Springer International Publishing), 30–43.
- Kumar, V., and Michael, N. (2012). Opportunities and challenges with autonomous micro aerial vehicles. *Int. J. Robot. Res.* 31, 1279–1291. doi: 10.1177/0278364912455954

- Kushleyev, A., Mellinger, D., Powers, C., and Kumar, V. (2013). Towards a swarm of agile micro quadrotors. *Auton. Robots* 35, 287–300. doi: 10.1007/s10514-013-9349-9
- Lajoie, P.-Y., Ramtoul, B., Chang, Y., Carbone, L., and Beltrame, G. (2019). Door-slam: distributed, online, and outlier resilient slam for robotic teams. *arXiv [preprint]*. arXiv:1909.12198v2. doi: 10.1109/LRA.2020.2967681
- Laković, N., Brkić, M., Batinić, B., Bajić, J., Rajs, V., and Kulundžić, N. (2019). “Application of low-cost VL53L0X ToF sensor for robot environment detection,” in *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH)* (East Sarajevo), 1–4. doi: 10.1109/INFOTEH.2019.8717779
- Lamers, K., Tijmons, S., De Wagter, C., and de Croon, G. (2016). “Self-supervised monocular distance learning on a lightweight micro air vehicle,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 1779–1784. doi: 10.1109/IROS.2016.7759284
- Lamming, T., Anton, E., and Henke, R. (2012). “Technology assessment on aircraft-level: modeling of innovative aircraft systems in conceptual aircraft design,” in *10th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)* (Fort Worth, TX). doi: 10.2514/6.2010-9072
- Leahy, K., Zhou, D., Vasile, C.-I., Oikonomopoulos, K., Schwager, M., and Belta, C. (2016). Persistent surveillance for unmanned aerial vehicles subject to charging and temporal logic constraints. *Autonom. Robots* 40, 1363–1378. doi: 10.1007/s10514-015-9519-z
- Lecoeur, J., Dacke, M., Floreano, D., and Baird, E. (2019). The role of optic flow pooling in insect flight control in cluttered environments. *Sci. Rep.* 9:7707. doi: 10.1038/s41598-019-44187-2
- Ledergerber, A., Hamer, M., and D’Andrea, R. (2015). “A robot self-localization system using one-way ultra-wideband communication,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg), 3131–3137. doi: 10.1109/IROS.2015.7353810
- Lee, D., Zhou, J., and Lin, W. T. (2015). “Autonomous battery swapping system for quadcopter,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)* (Denver, CO). doi: 10.1109/ICUAS.2015.7152282
- Lee, J., Su, Y., and Shen, C. (2007). “A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi,” in *IECON 2007–33rd Annual Conference of the IEEE Industrial Electronics Society* (Taipei), 46–51. doi: 10.1109/IECON.2007.4460126
- Lehman, J., and Stanley, K. O. (2011). Abandoning objectives: evolution through the search for novelty alone. *Evol. Comput.* 19, 189–223. doi: 10.1162/EVCO_a_00025
- Lehnert, C., and Corke, P. (2013). “μav–design and implementation of an open source micro quadrotor,” in *Australasian Conference on Robotics and Automation (ACRA2013)*, eds J. Katupitiya, J. Guivant, and R. Eaton (Sydney, NSW: University of New South Wales; Australian Robotics & Automation Association), 1–8.
- Leishman, R. C., Macdonald, J. C., Beard, R. W., and McLain, T. W. (2014). Quadrotors and accelerometers: State estimation with an improved dynamic model. *IEEE Control Syst. Mag.* 34, 28–41. doi: 10.1109/MCS.2013.2287362
- Leonard, J., Savvaris, A., and Tsouros, A. (2014). Energy management in swarm of unmanned aerial vehicles. *J. Intel. Robot. Sys.* 74, 233–250. doi: 10.1007/s10846-013-9893-8
- Lew, T., Emmei, T., Fan, D. D., Bartlett, T., Santamaria-Navarro, A., Thakker, R., et al. (2019). “Contact inertial odometry: collisions are your friend,” in *International Symposium on Robotics Research (ISRR)* (Hanoi).
- Li, X., and Parker, L. E. (2007). “Sensor analysis for fault detection in tightly-coupled multi-robot team tasks,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation* (Roma), 3269–3276. doi: 10.1109/ROBOT.2007.363977
- Li, X., and Parker, L. E. (2009). “Distributed sensor analysis for fault detection in tightly-coupled multi-robot team tasks,” in *2009 IEEE International Conference on Robotics and Automation* (Kobe), 3103–3110. doi: 10.1109/ROBOT.2009.5152389
- Lindsey, Q., Mellinger, D., and Kumar, V. (2012). Construction with quadrotor teams. *Autonom. Robots* 33, 323–336. doi: 10.1007/s10514-012-9305-0
- Liu, W., Loianno, G., Mohta, K., Daniilidis, K., and Kumar, V. (2018). “Semi-dense visual-inertial odometry and mapping for quadrotors with swap constraints,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD), 3904–3909. doi: 10.1109/ICRA.2018.8463163
- López, E., Barea, R., Gómez, A., Saltos, Á., Bergasa, L. M., Molinos, E. J., et al. (2016). “Indoor slam for micro aerial vehicles using visual and laser sensor fusion,” in *Robot 2015: Second Iberian Robotics Conference*, eds L. P. Reis, A. P. Moreira, P. U. Lima, L. Montano, and V. Muñoz-Martinez (Cham: Springer International Publishing), 531–542.
- Macdonald, J., Leishman, R., Beard, R., and McLain, T. (2014). Analysis of an improved IMU-based observer for multirotor helicopters. *J. Intell. Robot. Syst.* 74, 1049–1061. doi: 10.1007/s10846-013-9835-5
- Mademlis, I., Nikolaidis, N., Tefas, A., Pitas, I., Wagner, T., and Messina, A. (2019). Autonomous uav cinematography: a tutorial and a formalized shot-type taxonomy. *ACM Comput. Surv.* 52, 105:1–105:33. doi: 10.1145/3347713
- Mahony, R., Kumar, V., and Corke, P. (2012). Multirotor aerial vehicles: modeling, estimation, and control of quadrotor. *IEEE Robot. Autom. Mag.* 19, 20–32. doi: 10.1109/MRA.2012.2206474
- Mairaj, A., Baba, A. I., and Javaid, A. Y. (2019). Application specific drone simulators: recent advances and challenges. *Simul. Modell. Pract. Theory* 94, 100–117. doi: 10.1016/j.simp.2019.01.004
- Martinelli, A. (2011). “Closed-form solution for attitude and speed determination by fusing monocular vision and inertial sensor measurements,” in *2011 IEEE International Conference on Robotics and Automation (ICRA)* (Shanghai), 4538–4545. doi: 10.1109/ICRA.2011.5980081
- Martinoli, A., and Easton, K. (2003). “Modeling swarm robotic systems,” in *Experimental Robotics VIII*, eds B. Siciliano and P. Dario (Berlin; Heidelberg: Springer Berlin Heidelberg), 297–306.
- Matthies, L., Brockers, R., Kuwata, Y., and Weiss, S. (2014). “Stereo vision-based obstacle avoidance for micro air vehicles using disparity space,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 3242–3249. doi: 10.1109/ICRA.2014.6907325
- McCallum, R. (1997). *Reinforcement learning with selective perception and hidden state* (Ph.D. thesis), University of Rochester, Rochester, NY, United States.
- McGuire, K., de Croon, G., Wagter, C. D., Tuyls, K., and Kappen, H. (2017). Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone. *IEEE Robot. Autom. Lett.* 2, 1070–1076. doi: 10.1109/LRA.2017.2658940
- McGuire, K. N., De Wagter, C., Tuyls, K., Kappen, H. J., and de Croon, G. C. H. E. (2019). Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment. *Sci. Robot.* 4:eaw9710. doi: 10.1126/scirobotics.aaw9710
- Meier, L., Honegger, D., and Pollefeys, M. (2015). “PX4: a node-based multithreaded open source robotics framework for deeply embedded platforms,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA), 6235–6240. doi: 10.1109/ICRA.2015.7140074
- Melhuish, C., and Welsby, J. (2002). “Gradient ascent with a group of minimalist real robots: implementing secondary swarming,” in *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2 (Yasmine Hammamet), 509–514. doi: 10.1109/ICSMC.2002.1173464
- Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., and von Stryk, O. (2012). “Comprehensive simulation of quadrotor UAVs using ROS and Gazebo,” in *Simulation, Modeling, and Programming for Autonomous Robots*, eds I. Noda, N. Ando, D. Bruggali, and J. J. Kuffner (Berlin; Heidelberg: Springer Berlin Heidelberg), 400–411.
- Michael, N., Mellinger, D., Lindsey, Q., and Kumar, V. (2010). The GRASP multiple micro-UAV testbed. *IEEE Robot. Autom. Mag.* 17, 56–65. doi: 10.1109/MRA.2010.937855
- Michelson, R. C., and Reece, S. (1998). “Update on flapping wing micro air vehicle research—ongoing work to develop a flapping wing, crawling entomopter,” in *13th Bristol International RPV/UAV Systems Conference Proceedings*, Vol. 30 (Bristol), 30–31.
- Millan-Romera, J. A., Perez-Leon, H., Castillejo-Calle, A., Maza, I., and Ollero, A. (2019). “ROS-MAGNA, a ROS-based framework for the definition and management of multi-UAS cooperative missions,” in *International Conference on Unmanned Aircraft Systems (ICUAS)* (Atlanta, GA), 1477–1486. doi: 10.1109/ICUAS.2019.8797829
- Mintchev, S., de Rivaz, S., and Floreano, D. (2017). Insect-inspired mechanical resilience for multicopters. *IEEE Robot. Autom. Lett.* 2, 1248–1255. doi: 10.1109/LRA.2017.2658946

- Mirjan, A., Augugliaro, F., D'Andrea, R., Gramazio, F., and Kohler, M. (2016). *Building a Bridge with Flying Robots*. Cham: Springer International Publishing, 34–47.
- Mohr, B. B., and Fitzpatrick, D. L. (2008). Micro air vehicle navigation system. *IEEE Aerosp. Electron. Syst. Mag.* 23, 19–24. doi: 10.1109/MAES.2008.4493438
- Montijano, E., Cristofalo, E., Zhou, D., Schwager, M., and Sagüés, C. (2016). Vision-based distributed formation control without an external positioning system. *IEEE Trans. Robot.* 32, 339–351. doi: 10.1109/TRO.2016.2523542
- Moore, R. J. D., Dantu, K., Barrows, G. L., and Nagpal, R. (2014). “Autonomous MAV guidance with a lightweight omnidirectional vision sensor,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 3856–3861. doi: 10.1109/ICRA.2014.6907418
- Mueggler, E., Baumli, N., Fontana, F., and Scaramuzza, D. (2015). “Towards evasive maneuvers with quadrotors using dynamic vision sensors,” in *2015 European Conference on Mobile Robots (ECMR)* (Lincoln, UK), 1–8. doi: 10.1109/ECMR.2015.7324048
- Mueller, M., and Drouin, A. (2007). “Paparazzi—the free autopilot. Build your own UAV,” in *24th Chaos Communication Congress* (Berlin), 27–30.
- Mulgaonkar, Y., Cross, G., and Kumar, V. (2015). “Design of small, safe and robust quadrotor swarms,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA), 2208–2215. doi: 10.1109/ICRA.2015.7139491
- Mulgaonkar, Y., and Kumar, V. (2014). “Autonomous charging to enable long-endurance missions for small aerial robots,” in *Micro- and Nanotechnology Sensors, Systems, and Applications VI*, Vol. 9083, eds T. George, M. S. Islam, and A. K. Dutta (Baltimore, MD: International Society for Optics and Photonics, SPIE), 404–418.
- Mulgaonkar, Y., Makineni, A., Guerrero-Bonilla, L., and Kumar, V. (2018). Robust aerial robot swarms without collision avoidance. *IEEE Robot. Autom. Lett.* 3, 596–603. doi: 10.1109/LRA.2017.2775699
- Mulgaonkar, Y., Whitzer, M., Morgan, B., Kroninger, C. M., Harrington, A. M., and Kumar, V. (2014). “Power and weight considerations in small, agile quadrotors,” in *Micro- and Nanotechnology Sensors, Systems, and Applications VI*, Vol. 9083, eds T. George, M. S. Islam, and A. K. Dutta (Baltimore, MD: International Society for Optics and Photonics, SPIE), 376–391. doi: 10.1117/12.2051112
- Nägeli, T., Conte, C., Domahidi, A., Morari, M., and Hilliges, O. (2014). “Environment-independent formation flight for micro aerial vehicles,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Chicago, IL), 1141–1146. doi: 10.1109/IROS.2014.6942701
- Nagi, J., Giusti, A., Gambardella, L. M., and Di Caro, G. A. (2014). “Human-swarm interaction using spatial gestures,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Chicago, IL), 3834–3841. doi: 10.1109/IROS.2014.6943101
- Nascimento, T. P., and Saska, M. (2019). Position and attitude control of multi-rotor aerial vehicles: a survey. *Annu. Rev. Control* 48, 129–146. doi: 10.1016/j.arcontrol.2019.08.004
- Nedjah, N., and Junior, L. S. (2019). Review of methodologies and tasks in swarm robotics towards standardization. *Swarm Evol. Comput.* 50:100565. doi: 10.1016/j.swevo.2019.100565
- Nelson, A. L., Barlow, G. J., and Doitsidis, L. (2009). Fitness functions in evolutionary robotics: a survey and analysis. *Robot. Auton. Syst.* 57, 345–370. doi: 10.1016/j.robot.2008.09.009
- Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., et al. (2011). “Kinectfusion: real-time dense surface mapping and tracking,” in *2011 10th IEEE International Symposium on Mixed and Augmented Reality* (Basel), 127–136. doi: 10.1109/ISMAR.2011.6092378
- Nex, F., and Remondino, F. (2014). Uav for 3D mapping applications: a review. *Appl. Geomat.* 6, 1–15. doi: 10.1007/s12518-013-0120-x
- Nguyen, T., Qiu, Z., Nguyen, T. H., Cao, M., and Xie, L. (2019). Distance-based cooperative relative localization for leader-following control of MAVs. *IEEE Robot. Autom. Lett.* 4, 3641–3648. doi: 10.1109/LRA.2019.2926671
- Nieuwenhuisen, M., Beul, M., Rosu, R. A., Quenzel, J., Pavlichenko, D., Houben, S., et al. (2017). “Collaborative object picking and delivery with a team of micro aerial vehicles at MBZIRC,” in *2017 European Conference on Mobile Robots (ECMR)* (Paris), 1–6. doi: 10.1109/ECMR.2017.8098676
- Nolfi, S. (2002). Power and the limits of reactive agents. *Neurocomputing* 42, 119–145. doi: 10.1016/S0925-2312(01)00598-7
- Noth, A., and Siegwart, R. (2010). *Solar-Powered Micro-Air Vehicles and Challenges in Downscaling*. Berlin; Heidelberg: Springer Berlin Heidelberg, 285–297.
- Obert, E. (2009). *Aerodynamic Design of Transport Aircraft*. Amsterdam: IOS Press.
- Odelga, M., Stegagno, P., and Bühlhoff, H. H. (2016). “Obstacle detection, tracking and avoidance for a teleoperated UAV,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (Stockholm), 2984–2990. doi: 10.1109/ICRA.2016.7487464
- Oh, K.-K., Park, M.-C., and Ahn, H.-S. (2015). A survey of multi-agent formation control. *Automatica* 53, 424–440. doi: 10.1016/j.automatica.2014.10.022
- Olejnik, D., Duisterhof, B., Karásek, M., Scheper, K. Y. W., van Dijk, T., and de Croon, G. C. H. E. (2019). “A tailless flapping wing mav performing monocular visual servoing tasks,” in *International Micro Air Vehicle Conference and Competition (IMAV)* (Madrid).
- Oleynikova, H., Honegger, D., and Pollefeys, M. (2015). “Reactive avoidance using embedded stereo vision for MAV flight,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA), 50–56. doi: 10.1109/ICRA.2015.7138979
- Opromolla, R., Fasano, G., Rufino, G., Grassi, M., and Savvaris, A. (2016). “Lidar-inertial integration for UAV localization and mapping in complex environments,” in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)* (Arlington, VA), 649–656. doi: 10.1109/ICUAS.2016.7502580
- Opromolla, R., Inchingolo, G., and Fasano, G. (2019). Airborne visual detection and tracking of cooperative UAVs exploiting deep learning. *Sensors* 19:4332. doi: 10.3390/s19194332
- Oung, R., and D'Andrea, R. (2011). The distributed flight array. *Mechatronics* 21, 908–917. doi: 10.1016/j.mechatronics.2010.08.003
- Palunko, I., Fierro, R., and Cruz, P. (2012). “Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: a dynamic programming approach,” in *2012 IEEE International Conference on Robotics and Automation* (Saint Paul, MN), 2691–2697. doi: 10.1109/ICRA.2012.6225213
- Pan, Z., An, L., and Wen, C. (2019). Recent advances in fuel cells based propulsion systems for unmanned aerial vehicles. *Appl. Energy* 240, 473–485. doi: 10.1016/j.apenergy.2019.02.079
- Parsons, S., and Wooldridge, M. (2002). Game theory and decision theory in multi-agent systems. *Autonom. Agents Multi Agent Syst.* 5, 243–254. doi: 10.1007/978-1-4615-1107-6
- Pascoal, K., Kaminer, I., and Oliveira, P. (2000). Navigation system design using time-varying complementary filters. *IEEE Trans. Aerosp. Electron. Syst.* 36, 1099–1114. doi: 10.1109/7.892661
- Pestana, J., Sanchez-Lopez, J. L., de la Puente, P., Carrio, A., and Campoy, P. (2014). “A vision-based quadrotor swarm for the participation in the 2013 international micro air vehicle competition,” in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)* (Orlando, FL), 617–622. doi: 10.1109/ICUAS.2014.6842305
- Petricca, L., Ohlckers, P., and Grinde, C. (2011). Micro-and nano-air vehicles: state of the art. *Int. J. Aerosp. Eng.* 2011:214549. doi: 10.1155/2011/214549
- Pincirol, C., and Beltrame, G. (2016). “Buzz: an extensible programming language for heterogeneous swarm robotics,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 3794–3800. doi: 10.1109/IROS.2016.7759558
- Pincirol, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., et al. (2012). Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell.* 6, 271–295. doi: 10.1007/s11721-012-0072-5
- Pitonakova, L., Crowder, R., and Bullock, S. (2018). Information exchange design patterns for robot swarm foraging and their application in robot control algorithms. *Front. Robot. AI* 5:47. doi: 10.3389/frobt.2018.00047
- Portelli, G., Ruffier, F., Roubieu, F. L., and Franceschini, N. (2011). Honeybees' speed depends on dorsal as well as lateral, ventral and frontal optic flows. *PLoS ONE* 6:e19486. doi: 10.1371/journal.pone.0019486
- Powers, C., Mellinger, D., Kushleyev, A., Kothmann, B., and Kumar, V. (2013). “Influence of aerodynamics and proximity effects in quadrotor flight,” in *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, eds J. P. Desai, G. Dudek, O. Khatib, and V. Kumar (Heidelberg: Springer International Publishing), 289–302.
- Preiss, J. A., Honig, W., Sukhatme, G. S., and Ayanian, N. (2017). “Crazyswarm: A large nano-quadcopter swarm,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Singapore), 3299–3304. doi: 10.1109/ICRA.2017.7989376

- Qin, L., He, X., and Zhou, D. (2014). A survey of fault diagnosis for swarm systems. *Syst. Sci. Control Eng.* 2, 13–23. doi: 10.1080/21642583.2013.873745
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., et al. (2009). “ROS: an open-source robot operating system,” in *ICRA Workshop on Open Source Software*, Vol. 3 (Kobe), 5.
- Quintero, S. A. P., Collins, G. E., and Hespanha, J. P. (2013). “Flocking with fixed-wing UAVs for distributed sensing: a stochastic optimal control approach,” in *2013 American Control Conference* (Washington, DC), 2025–2031. doi: 10.1109/ACC.2013.6580133
- Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Trans. Syst. Man Cybernet.* 13, 257–266.
- Remes, B. D. W., Esden-Tempski, P., Van Tienen, F., Smeur, E., De Wagter, C., and De Croon, G. C. H. E. (2014). “Lisa-S 2.8 g autopilot for GPS-based flight of MAVs,” in *International Micro Air Vehicle Conference and Competition (IMAV)* (Delft).
- Reynolds, C. W. (1987). “Flocks, herds and schools: a distributed behavioral model,” in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87* (New York, NY: ACM), 25–34.
- Roberts, J. F., Stirling, T., Zufferey, J.-C., and Floreano, D. (2012). 3-D relative positioning sensor for indoor flying robots. *Autonom. Robots* 33, 5–20. doi: 10.1007/s10514-012-9277-0
- Roelofs, S., Gillet, D., and Martinoli, A. (2015). “Reciprocal collision avoidance for quadrotors using on-board visual detection,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg), 4810–4817. doi: 10.1109/IROS.2015.7354053
- Ross, P. E. (2014). Open-source drones for fun and profit. *IEEE Spectr.* 51, 54–59. doi: 10.1109/MSPEC.2014.6745885
- Ross, S., Melik-Barkhudarov, N., Shankar, K. S., Wendel, A., Dey, D., Bagnell, J. A., et al. (2013). “Learning monocular reactive UAV control in cluttered natural environments,” in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe), 1765–1772. doi: 10.1109/ICRA.2013.6630809
- Ruiz-Espitia, O., Martinez-Carranza, J., and Rascon, C. (2018). “AIRA-UAS: an evaluation corpus for audio processing in unmanned aerial system,” in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)* (Dallas, TX), 836–845. doi: 10.1109/ICUAS.2018.8453466
- Sa, I., and Corke, P. (2014). “Vertical infrastructure inspection using a quadcopter and shared autonomy control,” in *Field and Service Robotics: Results of the 8th International Conference*, eds K. Yoshida and S. Tadokoro (Berlin; Heidelberg: Springer Berlin Heidelberg), 219–232.
- Sabatini, A. M., and Genovese, V. (2013). A stochastic approach to noise modeling for barometric altimeters. *Sensors* 13, 15692–15707. doi: 10.3390/s131115692
- Saha, S., Natraj, A., and Waharte, S. (2014). “A real-time monocular vision-based frontal obstacle detection and avoidance for low cost UAVs in GPS denied environment,” in *2014 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology* (Yogyakarta), 189–195. doi: 10.1109/ICARES.2014.7024382
- Şahin, E. (2005). “Swarm robotics: from sources of inspiration to domains of application,” in *Swarm Robotics*, eds E. Şahin and W. M. Spears (Berlin; Heidelberg: Springer), 10–20.
- Şahin, E., Girgin, S., Bayindir, L., and Turgut, A. E. (2008). *Swarm Robotics*. Berlin; Heidelberg: Springer Berlin Heidelberg, 87–100.
- Saldaña, D., Gabrich, B., Li, G., Yim, M., and Kumar, V. (2018). “ModQuad: the flying modular structure that self-assembles in midair,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD), 691–698. doi: 10.1109/ICRA.2018.8461014
- Saldaña, D., Gupta, P. M., and Kumar, V. (2019). Design and control of aerial modules for inflight self-disassembly. *IEEE Robot. Autom. Lett.* 4, 3410–3417. doi: 10.1109/LRA.2019.2926680
- Saldaña, D., Prorok, A., Sundaram, S., Campos, M. F. M., and Kumar, V. (2017). “Resilient consensus for time-varying networks of dynamic agents,” in *2017 American Control Conference (ACC)* (Seattle, WA), 252–258. doi: 10.23919/ACC.2017.7962962
- Sanchez-Lopez, J. L., Fernández, R. A. S., Bavl, H., Sampedro, C., Molina, M., Pestana, J., et al. (2016). “AEROSTACK: an architecture and open-source software framework for aerial robotics,” in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)* (Arlington, VA), 332–341. doi: 10.1109/ICUAS.2016.7502591
- Santamaria-Navarro, A., Solà, J., and Andrade-Cetto, J. (2015). “High-frequency MAV state estimation using low-cost inertial and optical flow measurement units,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg), 1864–1871. doi: 10.1109/IROS.2015.7353621
- Saska, M. (2015). “MAV-swarms: unmanned aerial vehicles stabilized along a given path using onboard relative localization,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)* (Denver, CO), 894–903. doi: 10.1109/ICUAS.2015.7152376
- Saska, M., Baca, T., Spurný, V., Loianno, G., Thomas, J., Krajník, T., et al. (2016a). “Vision-based high-speed autonomous landing and cooperative objects grasping-towards the MBZIRC competition,” in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems-Vision-based High Speed Autonomous Navigation of UAVs (Workshop)* (Daejeon), 9–14.
- Saska, M., Baca, T., Thomas, J., Chudoba, J., Preucil, L., Krajník, T., et al. (2017). System for deployment of groups of unmanned micro aerial vehicles in gps-denied environments using onboard visual relative localization. *Autonom. Robots* 41, 919–944. doi: 10.1007/s10514-016-9567-z
- Saska, M., Vakula, J., and Přeucil, L. (2014). “Swarms of micro aerial vehicles stabilized under a visual relative localization,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 3570–3575. doi: 10.1109/ICRA.2014.6907374
- Saska, M., Vonásek, V., Chudoba, J., Thomas, J., Loianno, G., and Kumar, V. (2016b). Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. *J. Intell. Robot. Syst.* 84, 469–492. doi: 10.1007/s10846-016-0338-z
- Saulnier, K., Saldaña, D., Prorok, A., Pappas, G. J., and Kumar, V. (2017). Resilient flocking for mobile robot teams. *IEEE Robot. Autom. Lett.* 2, 1039–1046. doi: 10.1109/LRA.2017.2655142
- Scaramuzza, D., Achtelik, M. C., Doitsidis, L., Friedrich, F., Kosmatopoulos, E., Martinelli, A., et al. (2014). Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robot. Autom. Mag.* 21, 26–40. doi: 10.1109/MRA.2014.2322295
- Scaramuzza, D., and Zhang, Z. (2019). Visual-inertial odometry of aerial robots. *arXiv [preprint]*. arXiv:1906.03289.
- Schauwecker, K., Ke, N. R., Scherer, S. A., and Zell, A. (2012). “Markerless visual control of a quad-rotor micro aerial vehicle by means of on-board stereo processing,” in *Autonomous Mobile Systems 2012*, eds P. Levi, O. Zweigle, K. Häußermann, nd B. Eckstein (Berlin; Heidelberg: Springer Berlin Heidelberg), 11–20.
- Schauwecker, K., and Zell, A. (2014). On-board dual-stereo-vision for the navigation of an autonomous MAV. *J. Intell. Robot. Syst.* 74, 1–16. doi: 10.1007/s10846-013-9907-6
- Scheper, K. (2019). *Abstraction as a tool to bridge the reality gap in evolutionary robotics* (Ph.D. thesis), Delft University of Technology, Delft, Netherlands.
- Scheper, K. Y. W., and de Croon, G. C. H. E. (2017). Abstraction, sensory-motor coordination, and the reality gap in evolutionary robotics. *Artif. Life* 23, 124–141. doi: 10.1162/ARTL_a_00227
- Scheper, K. Y. W., Tijmons, S., de Visser, C. C., and de Croon, G. C. H. E. (2016). Behavior trees for evolutionary robotics. *Artif. Life* 22, 23–48. doi: 10.1162/ARTL_a_00192
- Schiano, F., Franchi, A., Zelazo, D., and Giordano, P. R. (2016). “A rigidity-based decentralized bearing formation controller for groups of quadrotor UAVs,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 5099–5106.
- Schilling, F., Lecoœur, J., Schiano, F., and Floreano, D. (2019). Learning vision-based flight in drone swarms by imitation. *IEEE Robot. Autom. Lett.* 4, 4523–4530. doi: 10.1109/LRA.2019.2935377
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). “Grad-cam: visual explanations from deep networks via gradient-based localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)* (Venice), 618–626. doi: 10.1109/ICCV.2017.74
- Shah, S., Dey, D., Lovett, C., and Kapoor, A. (2018). “AirSim: high-fidelity visual and physical simulation for autonomous vehicles,” in *Field and Service Robotics*, eds M. Hutter and R. Siegwart (Cham: Springer International Publishing), 621–635.

- Shen, S., Michael, N., and Kumar, V. (2011). "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *2011 IEEE International Conference on Robotics and Automation* (Shanghai), 20–25. doi: 10.1109/ICRA.2011.5980357
- Shen, S., Mulgaonkar, Y., Michael, N., and Kumar, V. (2014). "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 4974–4981. doi: 10.1109/ICRA.2014.6907588
- Shi, J., He, B., Zhang, L., and Zhang, J. (2016). "Vision-based real-time 3D mapping for UAV with laser sensor," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 4524–4529. doi: 10.1109/IROS.2016.7759666
- Shilov, K. (2014). "The next generation design of autonomous mav flight control system SMARTAP," in *International Micro Air Vehicle Conference and Competition (IMAV)* (Delft).
- Silva, F., Duarte, M., Correia, L., Oliveira, S. M., and Christensen, A. L. (2016). Open issues in evolutionary robotics. *Evol. Comput.* 24, 205–236. doi: 10.1162/EVCO_a_00172
- Silva, F., Urbano, P., Correia, L., and Christensen, A. L. (2015). odNEAT: an algorithm for decentralised online evolution of robotic controllers. *Evol. Comput.* 23, 421–449. doi: 10.1162/EVCO_a_00141
- Smisek, J., Jancosek, M., and Pajdla, T. (2013). *3D With Kinect*. London: Springer London, 3–25.
- Snap, J., van den Berg, J., Guy, S. J., and Manocha, D. (2009). "Independent navigation of multiple mobile robots with hybrid reciprocal velocity obstacles," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (St. Louis, MO), 5917–5922. doi: 10.1109/IROS.2009.5354821
- Snap, J., van den Berg, J., Guy, S. J., and Manocha, D. (2011). The hybrid reciprocal velocity obstacle. *IEEE Trans. Robot.* 27, 696–706. doi: 10.1109/TRO.2011.2120810
- Soto-Gerrero, D., and Ramirez-Torres, J.-G. (2016). "A human-machine interface with unmanned aerial vehicles," in *Robotics and Mechatronics*, eds S. Zeghloul, M. A. Laribi, and J. P. Gazeau (Cham: Springer International Publishing), 233–242.
- Spurný, V., Báča, T., Saska, M., Pěnička, R., Krajník, T., Thomas, J., et al. (2019). Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. *J. Field Robot.* 36, 125–148. doi: 10.1002/rob.21816
- Standiff, S. B., Dolan, J. M., and Trebi-Ollennu, A. (2006). "Mission reliability estimation for multirobot team design," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Beijing), 2206–2211. doi: 10.1109/IROS.2006.282562
- Stegagno, P., Basile, M., Bülthoff, H. H., and Franchi, A. (2014). "A semi-autonomous uav platform for indoor remote operation with visual and haptic feedback," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 3862–3869. doi: 10.1109/ICRA.2014.6907419
- Stegagno, P., Cognetti, M., Franchi, A., and Oriolo, G. (2011). "Mutual localization using anonymous bearing measurements," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Francisco, CA), 469–474. doi: 10.1109/IROS.2011.6094635
- Suleiman, A., Zhang, Z., Carlone, L., Karaman, S., and Sze, V. (2018). "Navion: a fully integrated energy-efficient visual-inertial odometry accelerator for autonomous navigation of nano drones," in *2018 IEEE Symposium on VLSI Circuits* (Honolulu, HI), 133–134. doi: 10.1109/VLSIC.2018.8502279
- Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. 2nd Edn. Cambridge, MA: MIT Press.
- Swatman, R. (2016a). *Intel Stuns During CES Keynote With Record for Most Drones Airborne Simultaneously—Watch Incredible Footage*. Guinness World Records. Available online at: <https://www.guinnessworldrecords.com/news/brand-or-agency/2016/1/intel-stuns-during-ces-keynote-with-record-for-most-drones-airborne-simultaneous-411677>
- Swatman, R. (2016b). *Intel Launches 500 Drones Into Sky and Breaks World Record in Spectacular Style*. Guinness World Records. Available online at: <https://www.guinnessworldrecords.com/news/2016/1/intel-launches-500-drones-into-sky-and-breaks-world-record-in-spectacular-style-449886>
- Szabo, T. (2015). *Autonomous collision avoidance for swarms of MAVs: based solely on RSSI measurements* (Master's thesis), Delft University of Technology, Delft, Netherlands.
- Tagliabue, A., Kamel, M., Siegwart, R., and Nieto, J. (2019). Robust collaborative object transportation using multiple MAVs. *Int. J. Robot. Res.* 38, 1020–1044. doi: 10.1177/0278364919854131
- Tarapore, D., Christensen, A. L., Lima, P. U., and Carneiro, J. (2013). "Abnormality detection in multiagent systems inspired by the adaptive immune system," in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, AAMAS '13* (Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems), 23–30.
- Tarapore, D., Christensen, A. L., and Timmis, J. (2015a). "Abnormality detection in robots exhibiting composite swarm behaviours," in *The 2019 Conference on Artificial Life*, Vol. 27 (York), 406–413.
- Tarapore, D., Christensen, A. L., and Timmis, J. (2017). Generic, scalable and decentralized fault detection for robot swarms. *PLoS ONE* 12:e182058. doi: 10.1371/journal.pone.0182058
- Tarapore, D., Lima, P. U., Carneiro, J., and Christensen, A. L. (2015b). To err is robotic, to tolerate immunological: fault detection in multirobot systems. *Bioinspir. Biomim.* 10:016014. doi: 10.1088/1748-3190/10/1/016014
- Tarapore, D., Timmis, J., and Christensen, A. L. (2019). Fault detection in a swarm of physical robots based on behavioral outlier detection. *IEEE Trans. Robot.* 35, 1516–1522. doi: 10.1109/TRO.2019.2929015
- Teixeira, L., Maffra, F., Moos, M., and Chli, M. (2018). Vi-rpe: visual-inertial relative pose estimation for aerial vehicles. *IEEE Robot. Autom. Lett.* 3, 2770–2777. doi: 10.1109/LRA.2018.2837687
- Thurrowgood, S., Soccol, D., Moore, R. J. D., Bland, D., and Srinivasan, M. V. (2009). "A vision based system for attitude estimation of UAVs," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (St. Louis, MO), 5725–5730. doi: 10.1109/IROS.2009.5354041
- Tijs, E., de Croon, G. C. H. E., Wind, J., Remes, B., De Wagter, C., de Bree, H. E., et al. (2010). "Hear-and-avoid for micro air vehicles," in *International Micro Air Vehicle Competition and Competition (IMAV)* (Braunschweig), Vol. 69.
- Toksoz, T., Redding, J., Michini, M., Michini, B., How, J. P., Vavrana, M., et al. (2011). "Automated battery swap and recharge to enable persistent UAV missions," in *Infotech@Aerospace 2011*, Vol. 21 (St. Louis, MO). doi: 10.2514/6.2011-1405
- Trianni, V. (2008). *Evolutionary Swarm Robotics: Evolving Self-organising Behaviours in Groups of Autonomous Robots*, Vol. 108. Berlin; Heidelberg: Springer. doi: 10.1007/978-3-540-77612-3
- Trianni, V. (2014). Evolutionary robotics: model or design? *Front. Robot. AI* 1:13. doi: 10.3389/frobt.2014.00013
- Trianni, V., and Campo, A. (2015). *Fundamental Collective Behaviors in Swarm Robotics*. Berlin; Heidelberg: Springer Berlin Heidelberg, 1377–1394.
- Trianni, V., Groß, R., Labella, T. H., Şahin, E., and Dorigo, M. (2003). "Evolving aggregation behaviors in a swarm of robots," in *Advances in Artificial Life*, eds W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. T. Kim (Berlin; Heidelberg: Springer Berlin Heidelberg), 865–874.
- Tripicchio, P., Satler, M., Unetti, M., and Avizzano, C. A. (2018). Confined spaces industrial inspection with micro aerial vehicles and laser range finder localization. *Int. J. Micro Air Vehicles* 10, 207–224. doi: 10.1177/1756829318757471
- Troub, B., DePineuil, B., and Montalvo, C. (2017). Simulation analysis of a collision-tolerant micro-airship fleet. *Int. J. Micro Air Vehicles* 9, 297–305. doi: 10.1177/1756829317705326
- Tsykunov, E., Labazanov, L., Tleugazy, A., and Tsetserukou, D. (2018). "Swarmtouch: tactile interaction of human with impedance controlled swarm of nano-quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid), 4204–4209. doi: 10.1109/IROS.2018.8594424
- Tsykunov, E., and Tsetserukou, D. (2019). "WiredSwarm: high resolution haptic feedback provided by a swarm of drones to the user's fingers for VR interaction," in *VRST '19: 25th ACM Symposium on Virtual Reality Software and Technology* (New York, NY), 102:1–102:2. doi: 10.1145/3359996.3364789
- Valentini, G. (2017). *Achieving Consensus in Robot Swarms: Design and Analysis of Strategies for the Best-of-n Problem*, Vol. 706. Cham: Springer International Publishing. doi: 10.1007/978-3-319-53609-5

- van den Berg, J., Guy, S. J., Lin, M., and Manocha, D. (2011). "Reciprocal n-body collision avoidance," in *Robotics Research*, eds C. Pradalier, R. Siegwart, and G. Hirzinger (Berlin; Heidelberg, Springer Berlin Heidelberg), 3–19.
- van den Berg, J., Lin, M., and Manocha, D. (2008). "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation* (Pasadena, CA), 1928–1935. doi: 10.1109/ROBOT.2008.4543489
- van der Helm, S., Coppola, M., McGuire, K. N., and de Croon, G. C. H. E. (2019). On-board range-based relative localization for micro air vehicles in indoor leader–follower flight. *Autonom. Robots*. doi: 10.1007/s10514-019-09843-6
- van Hecke, K., de Croon, G., van der Maaten, L., Hennes, D., and Izzo, D. (2018). Persistent self-supervised learning: from stereo to monocular vision for obstacle avoidance. *Int. J. Micro Air Vehicles* 10, 186–206. doi: 10.1177/1756829318756355
- Vásárhelyi, G., Virágh, C., Somorjai, G., Nepusz, T., Eiben, A. E., and Vicsek, T. (2018). Optimized flocking of autonomous drones in confined environments. *Sci. Robot.* 3:eat3536. doi: 10.1126/scirobotics.eat3536
- Vasile, C. I., and Belta, C. (2014). "An automata-theoretic approach to the vehicle routing problem," in *Robotics: Science and Systems* (Berkeley, CA). doi: 10.15607/rss.2014.x.045
- Vedder, B., Eriksson, H., Skarin, D., Vinter, J., and Jonsson, M. (2015). "Towards collision avoidance for commodity hardware quadcopters with ultrasound localization," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)* (Denver, CO), 193–203. doi: 10.1109/ICUAS.2015.7152291
- Vempati, A. S., Khurana, H., Kabelka, V., Flueckiger, S., Siegwart, R., and Beardsley, P. (2019). A virtual reality interface for an autonomous spray painting UAV. *IEEE Robot. Autom. Lett.* 4, 2870–2877. doi: 10.1109/LRA.2019.2922588
- Verma, L., Fakharzadeh, M., and Choi, S. (2013). Wifi on steroids: 802.11ac and 802.11ad. *IEEE Wireless Commun.* 20, 30–35. doi: 10.1109/MWC.2013.6704471
- Vincenzi, D. A., Terwilliger, B. A., and Ison, D. C. (2015). Unmanned aerial system (UAS) human-machine interfaces: new paradigms in command and control. *Proc. Manuf.* 3, 920–927. doi: 10.1016/j.promfg.2015.07.139
- Voskuil, M., van Bogaert, J., and Rao, A. G. (2018). Analysis and design of hybrid electric regional turboprop aircraft. *CEAS Aeronaut. J.* 9, 15–25. doi: 10.1007/s13272-017-0272-1
- Walter, V., Staub, N., Franchi, A., and Saska, M. (2019). UVDAR system for visual relative localization with application to leader-follower formations of multirotor UAVs. *IEEE Robot. Autom. Lett.* 4, 2637–2644. doi: 10.1109/LRA.2019.2901683
- Walter, V., Staub, N., Saska, M., and Franchi, A. (2018). "Mutual localization of UAVs based on blinking ultraviolet markers and 3D time-position hough transform," in *IEEE 14th International Conference on Automation Science and Engineering (CASE)* (Munich), 298–303. doi: 10.1109/COASE.2018.8560384
- Weimerskirch, H., Martin, J., Clerquin, Y., Alexandre, P., and Jiraskova, S. (2001). Energy saving in flight formation. *Nature* 413, 697–698. doi: 10.1038/35099670
- Weinstein, A., Cho, A., Loianno, G., and Kumar, V. (2018). Visual inertial odometry swarm: an autonomous swarm of vision-based quadrotors. *IEEE Robot. Autom. Lett.* 3, 1801–1807. doi: 10.1109/LRA.2018.2800119
- Weiss, S., Achtelik, M., Kneip, L., Scaramuzza, D., and Siegwart, R. (2011). Intuitive 3D maps for MAV terrain exploration and obstacle avoidance. *J. Intell. Robot. Syst.* 61, 473–493. doi: 10.1007/s10846-010-9491-y
- Winfield, A. F., and Nembrini, J. (2006). Safety in numbers: fault-tolerance in robot swarms. *Int. J. Modell. Identif. Control* 1:30. doi: 10.1504/IJMIC.2006.008645
- Winfield, A. F. T., Harper, C. F., and Nembrini, J. (2006). "Towards the application of swarm intelligence in safety critical systems," in *The First Institution of Engineering and Technology International Conference on System Safety* (London), 7.
- Winfield, A. F. T., Harper, C. J., and Nembrini, J. (2005). *Towards Dependable Swarms and a New Discipline of Swarm Engineering*. Berlin; Heidelberg: Springer Berlin Heidelberg, 126–142.
- Wood, R., Nagpal, R., and Wei, G.-Y. (2013). Flight of the robobees. *Sci. Am.* 308, 60–65. doi: 10.1038/scientificamerican0313-60
- Yang, G.-Z., Bellingham, J., Dupont, P. E., Fischer, P., Floridi, L., Full, R., et al. (2018). The grand challenges of science robotics. *Sci. Robot.* 3. doi: 10.1126/scirobotics.aar7650
- Yousif, K., Bab-Hadiashar, A., and Hoseinnezhad, R. (2015). An overview to visual odometry and visual SLAM: applications to mobile robotics. *Intell. Ind. Syst.* 1, 289–311. doi: 10.1007/s40903-015-0032-7
- Yuan, Q., Zhan, J., and Li, X. (2017). Outdoor flocking of quadcopter drones with decentralized model predictive control. *ISA Trans.* 71, 84–92. doi: 10.1016/j.isatra.2017.07.005
- Zou, D., Tan, P., and Yu, W. (2019). Collaborative visual slam for multiple agents: a brief survey. *Virtual Reality Intell. Hardware* 1, 461–482. doi: 10.1016/j.vrih.2019.09.002
- Zufferey, J.-C., Beyeler, A., and Floreano, D. (2010). Autonomous flight at low altitude using light sensors and little computational power. *Int. J. Micro Air Vehicles* 2, 107–117. doi: 10.1260/1756-8293.2.2.107
- Zufferey, J.-C., Hauert, S., Stirling, T., Leven, S., Roberts, J., and Floreano, D. (2013). "Aerial collective systems," in *Handbook of Collective Robotics* (New York, NY: Pan Stanford), 609–659. doi: 10.1201/b14908-21

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Coppola, McGuire, De Wagter and de Croon. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Attraction, Dynamics, and Phase Transitions in Fire Ant Tower-Building

Gary K. Nave Jr.^{1,2*}, Nelson T. Mitchell², Jordan A. Chan Dick², Tyler Schuessler³,
Joaquin A. Lagarrigue^{2,4} and Orit Peleg^{1,2,5*}

¹ BioFrontiers Institute, University of Colorado Boulder, Boulder, CO, United States, ² Computer Science, University of Colorado Boulder, Boulder, CO, United States, ³ Applied Mathematics, University of Colorado Boulder, Boulder, CO, United States, ⁴ Psychology and Neuroscience, University of Colorado Boulder, Boulder, CO, United States, ⁵ Santa Fe Institute, Santa Fe, NM, United States

OPEN ACCESS

Edited by:

Heiko Hamann,
Universität zu Lübeck, Germany

Reviewed by:

Adam L. Cronin,
Tokyo Metropolitan University, Japan
Daniel Strömbom,
Lafayette College, United States
Wolfgang Elmenreich,
Alpen-Adria-Universität
Klagenfurt, Austria

*Correspondence:

Gary K. Nave Jr.
gary.k.nave@gmail.com
Orit Peleg
Orit.Peleg@colorado.edu

Specialty section:

This article was submitted to
Computational Intelligence in
Robotics,
a section of the journal
Frontiers in Robotics and AI

Received: 20 November 2019

Accepted: 13 February 2020

Published: 04 March 2020

Citation:

Nave GK Jr, Mitchell NT, Chan Dick JA, Schuessler T, Lagarrigue JA and Peleg O (2020) Attraction, Dynamics, and Phase Transitions in Fire Ant Tower-Building. *Front. Robot. AI* 7:25.
doi: 10.3389/frobt.2020.00025

Many insect species, and even some vertebrates, assemble their bodies to form multi-functional materials that combine sensing, computation, and actuation. The tower-building behavior of red imported fire ants, *Solenopsis invicta*, presents a key example of this phenomenon of collective construction. While biological studies of collective construction focus on behavioral assays to measure the dynamics of formation and studies of swarm robotics focus on developing hardware that can assemble and interact, algorithms for designing such collective aggregations have been mostly overlooked. We address this gap by formulating an agent-based model for collective tower-building with a set of behavioral rules that incorporate local sensing of neighboring agents. We find that an attractive force makes tower building possible. Next, we explore the trade-offs between attraction and random motion to characterize the dynamics and phase transition of the tower building process. Lastly, we provide an optimization tool that may be used to design towers of specific shapes, mechanical loads, and dynamical properties, such as mechanical stability and mobility of the center of mass.

Keywords: social insects, agent based modeling (ABM), self-assembly, phase transition, collective construction, swarms and collective behavior

1. INTRODUCTION

Collective aggregation is a prevalent behavior among social animals, where many individuals cluster together while feeding, defending against predators, or as a thermoregulation strategy, effectively reducing the exposed surface area per individual. Examples of species that aggregate include vertebrates, such as penguins (Waters et al., 2012) and bats (Roverud and Chappell, 1991; Kerth, 2008) as well as insects, such as beetles (Deneubourg et al., 1990), ants (Theraulaz et al., 2002; Reynaert et al., 2006), and cockroaches (Ame et al., 2004; Jeanson et al., 2005). While these aggregations are often planar, eusocial insects, such as honey bees (Seeley, 2010; Kastberger et al., 2011), army ants (Franks, 1989), and fire ants (Mlot et al., 2011) extend this strategy and create three-dimensional assemblages. These self-assemblages are composed of a multitude of individuals who link their bodies, doing so without a global overseer and with limited cognitive abilities (Anderson et al., 2002).

Ants in particular are capable of a wide variety of self-assemblages and collective behavior. For example, ants of the genus *Oecophylla* build chains for gap crossing and during nest construction (Lioni et al., 2001). In addition, army ants are known for their construction of bivouacs (Franks, 1989), and are also capable of building bridges out of their bodies to cross gaps along a foraging

trail (Reid et al., 2015). Finally, as we will discuss further, fire ants gather together to form rafts and towers when their habitat floods (Mlot et al., 2011, 2012; Phonekeo et al., 2017).

The structures that these insects create are, in essence, autonomous materials that embed sensing, computation, and actuation. These properties are some of the long-standing aspirations in the fields of multi-functional materials and robotic materials (Şahin, 2004; Hughes et al., 2019). Self-assembling agents have already begun to inspire robotic applications (Bonabeau et al., 1999; Brambilla et al., 2013; Hamann, 2018). For example, Del Dottore et al. (2018) have described the concept of “growing robots,” which are systems of a large number of individual robots working together to mimic biological growth in plants or groups of molecules or cells. Other collective robots are directly inspired by eusocial insects, such as the S-bots (Şahin et al., 2002; Groß et al., 2006), which form chains to collectively move larger payloads, just like ants working together to move larger food (Buffin and Pratt, 2016). Also inspired by ants, Swisler and Rubenstein (2018) have developed robots with a new docking mechanism to form self-assembling structures. Another class of robots, inspired by termites (Werfel et al., 2014), build three-dimensional structures out of external building materials. Finally, the cube-shaped M-Blocks (Romanishin et al., 2015) construct aggregations out of their own bodies, using magnetism and angular momentum to climb on top of neighbors. These works represent examples from the emerging field of multi-agent robotic systems built out of many inexpensive individual robots and utilizing control strategies that may include redundancies to overcome individual malfunctions. While much of the focus in robotics has been on developing the hardware, the algorithmic development of assembling processes has often been overlooked. We address this gap by borrowing tools from computational material science and characterize the dynamics of 3-dimensional aggregation formation inspired by fire ant towers.

In nature, red imported fire ant (*Solenopsis invicta*) towers tend to occur in the event of flooding. Initially, fire ants gather together to form hydrophobic rafts (Mlot et al., 2011, 2012) to float above the water surface. When the rafts approach vegetation emerging from the surface, they may attach to the vegetation and form towers on top of their floating rafts, as pictured in **Figure 1A**. In a recent, Phonekeo et al. (2017) described an experimental assay of the tower-building process in fire ants. The experimental setup involved fire ants constructing towers around a vertical rod to represent the emergent vegetation. In their analysis, the authors propose four rules which allow ants to build towers:

1. Do not move if ants are on top of you.
2. If atop other ants, repeatedly move a short distance in a random direction.
3. Upon reaching available space adjacent to non-moving ants, stop and link with them.
4. The top layer of the tower is not stable unless there is a complete innermost ring of ants gripping each other around the rod.

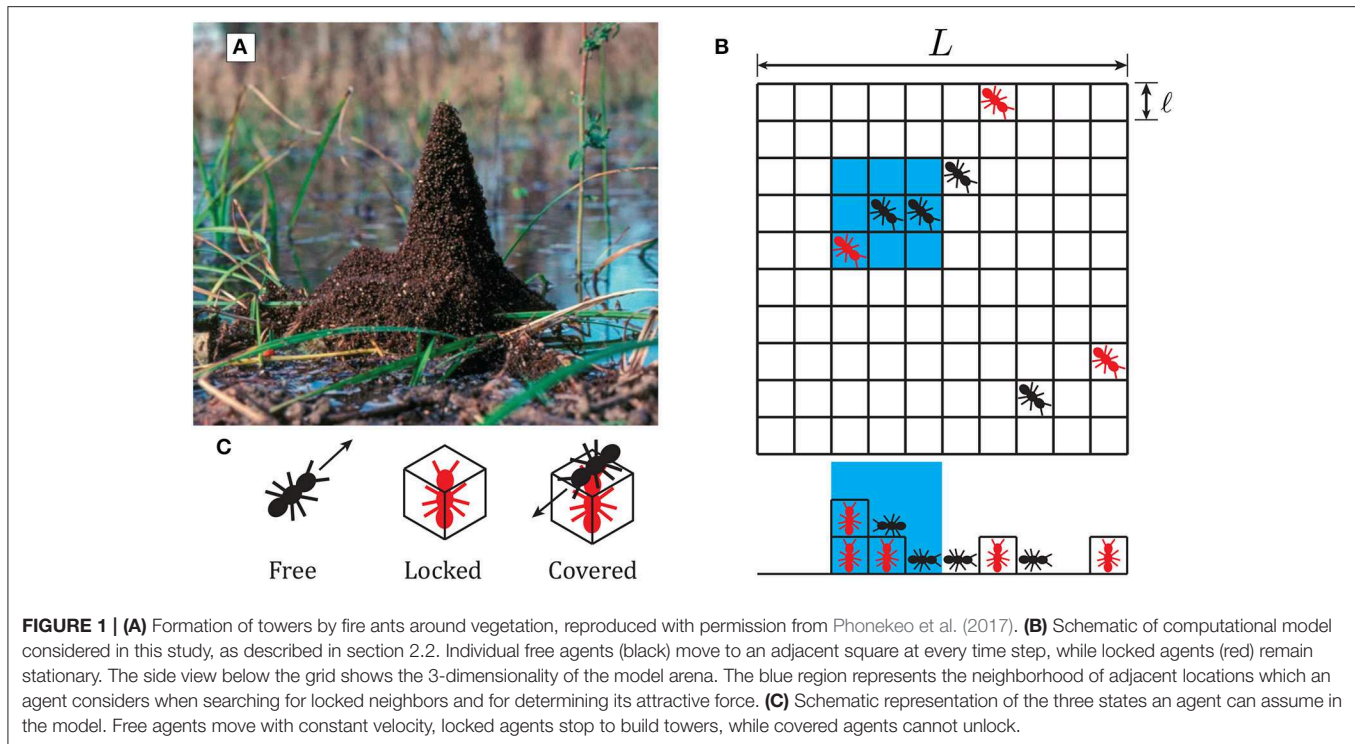
Note that the “available space adjacent to non-moving ants” is primarily discussed by Phonekeo et al. (2017) in the context

of a ring around the vertical rod or vegetation. We will take a more general definition of an available space in the present study, discussed below in section 2.1.

The work of Phonekeo et al. (2017) shows an agreement between the resulting tower shapes in the long-timescale limit; however, it does not explore the time dynamics and parameter space systematically. This is what the present work aims to do, since local rules, such as these provide a systematic way of analyzing collective behavior through agent-based modeling, and importantly, they are directly implementable in swarm robotic systems. By simulating the behavior of individuals following a set of local rules, it is possible to investigate how local interactions between agents lead to global emergent behavior and explore the space of possible behavior beyond what is possible with experiments.

Modeling efforts of collective behavior using local behavioral rules include the boids model (or Vicsek model) (Reynolds, 1987; Vicsek et al., 1995), which simulates agents moving under attraction, repulsion, and alignment as well as more complicated models (Couzin and Krause, 2003; Mishra et al., 2012; Wilensky and Rand, 2015). However, boids-type models best describe the behavior of more sparse collectives, such as flocks of birds or schools of fish. To model ants building a tower, we must account for dense aggregations where the interaction range is limited to a short length scale, preferably defined by the size of an individual agent. Models of more dense collective assemblies include aggregation in slime mold based on chemical signal amplification (Levine et al., 1997; Umeda and Inouye, 1999), and nest building in wasps using an agent-based model in which swarms of builders deposit bricks and build up a nest (Theraulaz and Bonabeau, 1995; Bonabeau et al., 2000). Agent-based modeling has been successfully applied to studies of ant collective behavior as well (Dorigo et al., 2000) to modeling traffic organization in ant foraging (Goss et al., 1989; Couzin and Franks, 2003; Strömbom and Dussutour, 2018), bridge and chain formation (Lioni et al., 2001; Garnier et al., 2013), and trail clearing (Bochynek et al., 2017). However, for the present study we must consider both moving ants as in traffic organization and trail clearing, which climb the tower and form the shape, as well as stationary ants that support the structure as in bridge and chain formation. Based on the similarity to the aggregation of inanimate systems, such as colloids (Deneubourg et al., 2002; Vernerey et al., 2018), we reason that ant tower building would experience dynamic phase separation processes including nucleation (Vlasov, 2019), jamming (Bak, 1996), and ripening (Voorhees, 1985). These phase transitions are also observed at the thermodynamic transition between phases of matter, which have been studied experimentally (Panagiotou et al., 1984) as well as computationally (Rovere et al., 1990; Navarro and Fielding, 2015). Hence, we formulate an agent-based model with a set of behavioral rules that lead to aggregation and experience dynamical phase transitions.

Section 2 describes the details of the model we study in the present work and lays out the modifications to the local rules (presented above) that we introduce to achieve tower-building. In section 3, we explore the parameter space of the local rules to identify the impacts of each component: locking, unlocking, and



attraction. We find that towers undergo a phase transition when varying the attraction parameter, and explore how this phase transition changes across various densities. Finally, we introduce an optimization algorithm to generate the largest possible tower for a given density of agents in the system. In section 4, we discuss the significance of the results and talk about implications for both the understanding of collective biological systems and the design of multi-agent robotic control strategies.

2. AGENT-BASED MODEL

We consider a system of N individual agents simulated to move in a $L \times L \times \infty$ arena, discretized into a cubic lattice made of voxels of volume $\ell \times \ell \times \ell$. The volume of an individual agent is set to be the volume of a voxel, where $\ell \equiv 1$. At each time step, an individual agent can move into one of its 26 neighboring voxels: 9 above, 9 below, and 8 on the same level. A schematic of agents moving within the arena is shown in **Figure 1B**. In the present work, we will not consider the effects of solid wall boundaries and will instead implement periodic boundaries. The horizontal plane of the arena, therefore, contains periodic boundary conditions—when an agent leaves the right side of the arena, for example, it re-enters the left side. Periodic boundaries are also taken into account when distances between agents are calculated. The equations that define the periodic boundary conditions are given in (S1) and (S2) in **Appendix 2**. The vertical direction of the arena is semi-infinite, extending upward from a solid floor.

Agents move horizontally and climb up if the voxel they intended to move into is occupied by a locked agent. Note that the local rules described above, from Phonekeo et al. (2017), refer to agents “linking” with one another, while in the present work

we will refer to an agent that stops to support tower building as “locked.” Each pixel along the horizontal plane has an associated height equal to the number of locked agents on top of each other in that location. The free agents, therefore, are moving under 2-dimensional rules along the surface defined by locked agents, which is embedded in 3-dimensions. If an agent attempts to climb on top of neighbors to a voxel that is more than ℓ higher, it does not move at this time step. Agents may move down any distance but never move below the floor.

Agents in the model may take on three different states, depicted in **Figure 1C**: free, locked, or covered. A free agent may move around the arena according to a specific set of behavioral rules with a constant velocity of one voxel per time step. All agents determine their intended movement before moving, and movement order is chosen randomly at each time step. To prevent two individuals from occupying the same position, if two free agents attempt to move to the same voxel, the second agent to arrive randomly chooses an unoccupied voxel horizontally adjacent to the target voxel. Locked agents are those which have decided to become a part of a tower and allow their neighbors to climb on top of them. We explore different schemes for the decision to “lock” as defined in sections 2.1 and 2.2. Covered agents are locked agents with at least one other agent on top of them. Each time step consists of first evaluating movement for all individuals and then evaluating locking decisions for all individuals based on their new configuration. We will not consider the effect of stability and assume that each agent has infinite strength to support neighbors.

It is likely that pheromones play a role in fire ant tower building, but for the present study, we consider whether this behavior can arise from solely physical proximity to neighbors.

Hence, an agent can sense which of its surroundings 26 voxels are occupied by another agent. This local model will allow for easier implementation by collective robotic systems, as it merely requires local sensing.

Unless specified otherwise, all simulations contain $N = 1,000$ agents moving in a $100 \times 100 \times \infty$ arena, corresponding to a density of $\rho = \frac{N}{L^2} = 0.1$. Based on preliminary simulations, we have chosen to evaluate each trial for 500,000 time steps, for which 97.8% of all simulations considered reached a steady state, where the largest tower size remained approximately constant ($\pm 5\%$ of N) for at least the last 100,000 time steps of the simulation. Exceptions will be discussed below in section 3.2.

2.1. Diffusion-Limited Aggregation

We start by investigating whether a dynamic simulation of the proposed local rules above can lead to tower-building. As we are not considering effects of stability, we will ignore rule (iv) in the present study. We simulate the rules (i)–(iii) from Mlot et al. (2012) and Phonekeo et al. (2017) with a naive approach to what constitutes an available space adjacent to non-moving agents, assuming no direct knowledge of the agents about where they are relative to the rest of the tower. At each time step, each individual agent randomly chooses an adjacent square to move into, performing a random walk and fulfilling rule (ii). When an agent arrives in a voxel with at least one locked neighbor sharing a corner, edge, or side, it decides to lock, fulfilling rule (iii). Locked agents remain in place, and allow others to move on top of them. Finally, when agents climb on top of locked agents, the locked agent's status changes to covered, fulfilling rule (i). For the sake of simple implementation, we also allow agents to start tower building with a constant probability of spontaneous locking, $P_{sl} = \frac{1}{20,000}$.

This model leads to aggregations which grow horizontally rather than upward. An example of a final configurations from one such simulation is shown in **Appendix 1** and correspond to the boxed-in panel of **Figure S1**. This is illustrated in **Supplementary Video S1**, where each tower growing outward in fractal shapes from a center point. This behavior arises due the higher likelihood of an agent performing a random walk to find other agents near the outer edge of the aggregation.

These results closely resemble a phenomenon known as diffusion-limited aggregation (DLA) (Witten and Sander, 1981). DLA was developed to model the aggregation of metal particles which gather in wispy, fractal shapes, similar to the simulated agent aggregation in **Figure S1** for $P_u = 0, k_{nl} = 1$. DLA has also been observed in experimental colloidal aggregation systems, as in Reynaert et al. (2006). Without any rule modifications, DLA is unable to form dense aggregations of agents, because agents on the edge of the aggregation shadow those closer to the center. Hence, we propose several modifications to the behavioral rules which are necessary to mimic the time dynamics of tower shapes experimentally observed by Phonekeo et al. (2017).

2.2. Rule Modifications to Achieve Tower-Building

2.2.1. Probability of Unlocking

First, we allow locked agents to unlock with a constant probability, as long as they are not covered by other

agents. This allows individuals past the first locked neighbor they encounter and move further in toward the center of an aggregation. To model this, we introduce a constant probability of unlocking P_u which applies equally to all uncovered locked agents. This rule introduces a distinction between locked agents and covered agents—covered agents cannot unlock.

2.2.2. Neighbor-Influenced Locking Probability

Second, we loosen the requirement that agents must lock upon encountering another locked agent, and instead allow for their probability of locking to increase with an increasing number of locked neighbors. This new rule (ii) replaces the previously discussed rule that individuals lock immediately upon finding a locked neighbor. Instead, an individual has a probability to lock based on the number of locked agents in its neighborhood. We define this probability of neighbor-influenced locking as $P_{nl} = k_{nl}N_n$, with N_n representing the number of locked agents in an individual's neighborhood and k_{nl} specifying the increase in probability for each additional neighbor. The neighborhood is defined as a distance of one above below, or horizontally adjacent to the agent's location, highlighted by the blue region in **Figure 1B**.

The overall probability that a free agent chooses to lock is given by,

$$P_l = \min \{P_{sl} + P_{nl}, 1\}, \\ = \min \{P_{sl} + k_{nl}N_n, 1\}, \quad (1)$$

where P_{sl} is the probability of spontaneously locking. Note that the model allows for up to 26 neighbors, so the value of $P_{sl} + k_{nl}N_n$ may be > 1 . In this case, locking is guaranteed. Therefore, a min function is used to state that when $P_{sl} + k_{nl}N_n > 1$, the locking probability is $P_l = 1$. Additionally, the inverse of the neighbor locking factor, $\frac{1}{k_{nl}}$, may be thought of as the number of neighbors required to guarantee locking.

The probability of spontaneous locking provides a baseline probability of locking, to allow for individuals to randomly seed towers. In our simulations, we keep this probability small and set it to $P_{sl} = \frac{1}{20,000}$. The neighbor-influenced locking factor provides the urgency with which an agent locks next to locked neighbors.

2.2.3. Attraction Forces

As we show below in section 3.1 and **Appendix 1**, the two rule modifications above are unable to reproduce large tower-like structures. Therefore, we extend the random walk model discussed above, and add an attractive “force” representing a behavioral tendency to cluster together. Under this effect, individual agents search their immediate local neighborhood for other agents, and move toward the center of all neighbors. This motion is then perturbed by the randomness associated with a simple random walk model. The resulting velocity is given by,

$$\mathbf{v}_i = \mathbf{v}_{random} + \frac{c}{n_i} \sum_{j=1}^{n_i} (\mathbf{x}_j - \mathbf{x}_i), \quad (2)$$

where n_i is the number of neighbors in the agent's immediate neighborhood sharing at least one corner, edge, or side with

the agent's current position, and c is ratio of the magnitude of attraction relative to the magnitude of randomness. Each agent moves toward the available voxel most closely aligned with the direction of \mathbf{v}_i . The resulting normalized velocity, $\hat{\mathbf{v}}_i$ is defined in (S3) in **Appendix 2**. Each agent moves into the voxel defined by the surface height at the resulting pixel.

Software that simulates agents following these modified rules in MATLAB is provided in **Supplementary Code S8**.

2.3. Set of Modified Behavioral Rules

With the three modifications mentioned above, we modify the first three rules of Phonekeo et al. (2017) and Mlot et al. (2011) into four new local rules:

1. Do not move if agents are on top of you.
2. Upon reaching available space adjacent to non-moving agents, stop and lock with them with probability $P_l = k_{nl}N_n$.
3. If stopped and locked, but uncovered by other agents, spontaneously unlock with probability P_u .
4. If free to move, move generally toward your neighbors with some random noise as defined by equation (2).

2.4. Measurements of Tower Geometry

Each simulation is post-processed to measure the geometry of each tower in order to determine how tower-like the aggregation is. For the final configuration of each simulation, a 2-dimensional height map is constructed by assigning each pixel in the 2D projection of the arena a value equal to its maximum height (**Figure 2A**). We treat the resulting $L \times L$ array of pixel values as an image and apply connected-component analysis (Shapiro, 1996) to identify different towers—a labeled image is generated where any two non-zero pixels that share a corner or edge have the same label. Each agent in the simulation is then given the label corresponding to its horizontal position within the labeled image. As we are interested in building a single large tower, properties for the tower containing the largest number of agents from each simulation are reported. Three tower properties are considered: number of individuals per tower, maximum tower height, and the ratio of the tower height to its equivalent diameter. Equivalent diameter is defined as the diameter of a circle with area equivalent to the tower's base (**Figure 2A**).

3. RESULTS

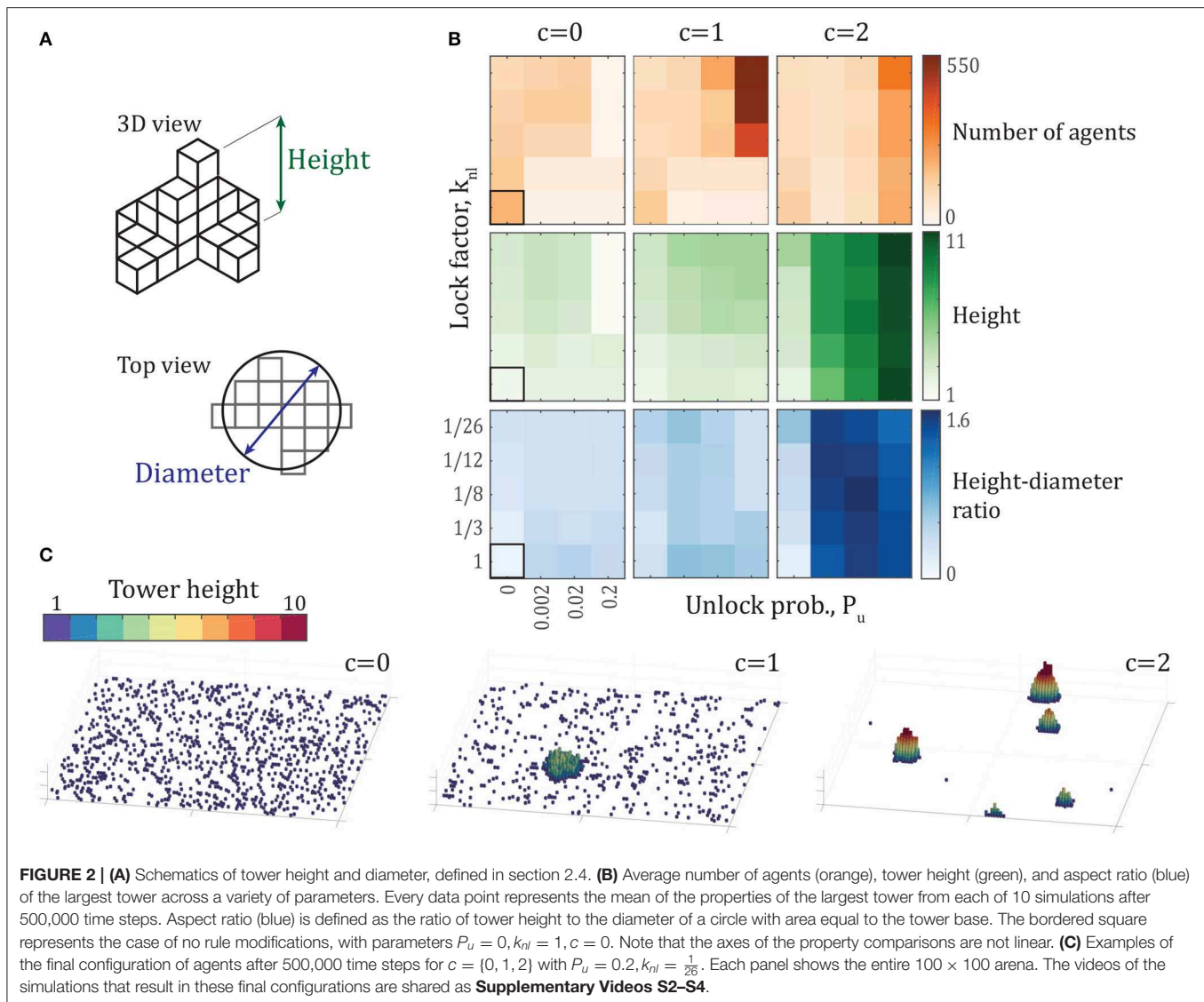
To gain an intuition for the effects of the modifications to the tower-building rules discussed in section 2.2, simulations were run over a range of locking and unlocking parameters, k_{nl} and P_u , across multiple attraction parameters, c , and in section 3.3, across varying densities of agents in the system, ρ . We begin with a parameter sweep across the locking and unlocking parameters and attraction parameter in section 3.1. Then, selecting a pair of locking and unlocking parameters, we systematically vary attraction c to show a rapid phase transition, and investigate the time dynamics of tower properties, both near and far from the phase transition in section 3.2. In section 3.3, we vary the density of agents along with attraction, and observe, in section 3.4, that the center of mass of the towers may continue to move. Finally,

we optimize for tower size and height in section 3.5 and identified a set of parameters where a tower formed of nearly all individuals in the simulation.

3.1. Tower Geometry

To explore the range of possible tower shapes in the model, we sweep the parameter space of the three rule modifications, including probability of unlocking P_u , neighbor-locking factor k_{nl} , and attraction factor c . Resulting tower properties and example final configurations from these simulations are shown in **Figure 2**. Every data point represents the mean of the largest tower's properties for each of 10 simulations. The left column of the array of tower properties, representing simulations with $c = 0$, shows that without attraction, towers tend to contain a small number of agents, a small height, and an especially low height-diameter ratio. These simulations with $c = 0$ represent the first two rule modifications—individual unlocking and neighbor-influenced locking—alone. From the measured tower properties in **Figure 2B**, we see the effects of the first two rule modifications without attraction. The aggregations with the largest number of agents are found in the simulations with parameters $k_{nl} = 1$ and $P_u = 0$, representing the case of no rule modifications at all. These aggregations lead to diffusion-limited aggregation as discussed above and shown in **Supplementary Video S1**. The locking and unlocking rule modifications, therefore, decrease the number of agents in the largest aggregation. They do provide an increase in tower height and the height-diameter ratio. This increase is modest, however, with the tallest average tower height reaching 3.4 agents tall for $P_u = 0.002, k_{nl} = \frac{1}{12}$, corresponding to a height-diameter ratio of 0.314. The largest height-diameter ratio occurs for the parameters $P_u = 0.02, k_{nl} = 1$, reaching a value of 0.49, with a corresponding average height of 2.2 and 19.9 agents in the largest tower for each simulation. Finally, the simulations with $c = 0$ and $P_u = 0.2$ with a small lock factor $k_{nl} \leq \frac{1}{8}$ finish the simulations without forming aggregations. **Supplementary Video S2** and the $c = 0$ configuration snapshot in **Figure 2C** show the dynamics and final configuration, respectively, of one such simulation which is unable to form aggregations, with parameters $P_u = 0.2, k_{nl} = \frac{1}{26}, c = 0$. These tower measurements show that without attraction, all of the tested parameter sets produce aggregations that remain small in number of individuals, do not reach average heights more than 3.4 layers, and remain wide and shallow.

When an attractive force is added, larger aggregations form, as shown by the center column of **Figure 2B** for an attraction ratio of $c = 1$. As unlock probability P_u increases and lock factor k_{nl} decreases, larger aggregations form, with the largest reaching over 500 individuals. However, these largest aggregations have the smallest height-diameter ratios of this set, showing that these large aggregations are particularly wide, as is visible in the $c = 1$ example in **Figure 2C** and **Supplementary Video S3**. Increasing the attraction ratio to $c = 2$ finally reveals a more typical tower-like shape, with taller aggregations, even reaching a height of 11 agents. Interestingly, these taller towers contain fewer agents than the $c = 1$ case. The reason for this is clear in the snapshots shown in **Figure 2C** and **Supplementary Video S4**—stronger attraction yields more densely-packed towers with larger height-diameter



ratios—the towers are so dense that multiple, smaller towers form instead of most individuals aggregating into a single tower.

3.2. Phase Transition and Time Dynamics of Tower-Building

The example configurations shown in **Figure 2C** represent the same set of locking and unlocking parameters, $P_u = 0.2, k_{nl} = \frac{1}{26}$ across $c = \{0, 1, 2\}$. These locking and unlocking parameters give the largest towers for both $c = 1$ and $c = 2$, but no aggregations at all for $c = 0$. To investigate the effects of the attraction ratio c further, we selected a fixed pair of locking and unlocking parameters and explored both the height and number of agents in the largest tower in the system for a densely sampled range of the attraction parameter c . The results of these simulations are shown in **Figures 3A,C**. The presence of a phase transition occurs between $c = 0.92$ to $c = 1.06$, where the number of agents in the largest tower climbs from close to 0 to over 700 agents. The results show that as c increases beyond that critical value, the number of

individuals in the largest tower decreases (**Figure 3A**) while the height of the largest tower increases (**Figure 3C**).

In **Figure 3B**, we show the time dynamics of the number of agents for tower for two cases, close to the phase transition and further from it. To illustrate tower growth further from the phase transition, **Figure 3B** shows the time histories of all 10 simulations for $P_u = 0.2, k_{nl} = \frac{1}{26}, c = 2.0$ in green and the mean of all simulations in black. One of these simulations is shown in **Supplementary Video S4**. The tower formation process in this model demonstrates two time scales: the time scale of initial nucleation, and the time scale of growth. Nucleation generally occurs within the first 5,000 time steps, the first 1% of each simulation. After nucleation, towers often continue to grow slowly through the rest of the simulation. Occasionally, two towers will merge into one, which manifests as a sharp jump in the time histories of **Figure 3B**. Some of these tower collisions last through the rest of the simulation, while others briefly merge and then separate again, which shows up as a sharp peak in the

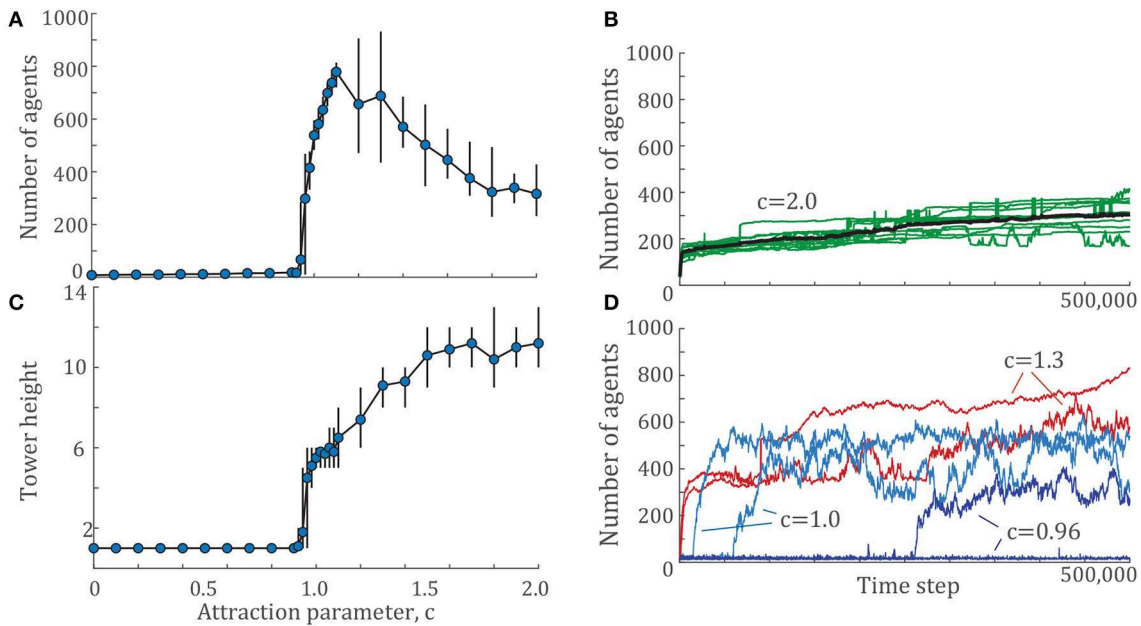


FIGURE 3 | (A) The number of agents and **(C)** tower height of the largest tower across a range of attraction parameters, c for a given pair of locking and unlocking values, $P_u = 0.2, k_{nl} = \frac{1}{26}$. Each data point shows the mean value over 10 simulations of 500,000 time steps each. Error bars indicate the maximum and minimum values observed. **(B)** Example time histories from 10 simulations of $P_u = 0.2, k_{nl} = \frac{1}{26}, c = 2.0$ in green with the mean at each time step shown in black. **(D)** Time histories from two examples for several c values near the phase transition, $c = \{0.96, 1.0, 1.3\}$.

time history of tower size. The fast nucleation followed by slow growth seen for $c = 2.0$ is typical for most simulations in the present work.

However, there are some examples, particularly within the phase transition regime, for which a critical slowing down occurs. Trajectories close to the phase transition are shown in **Figure 3D**. Two trajectories are shown for each of $c = \{0.96, 1.0, 1.3\}$ with $P_u = 0.2, k_{nl} = \frac{1}{26}$. The critical slowing down is particularly evident for the $c = 0.96$ trajectories, where agents aggregate into a tower after 250,000 time steps while the other never transitions out of the disordered state. The $c = 1.0$ trajectories also show variation in nucleation time, although in this case, all simulations have transitioned to their aggregated state, in which the largest tower contains at least 100 agents. The variation in tower size is highest for these examples, varying in size by 200 or more individuals. There are also cases where the towers continue to grow in size, even after 500,000 time steps, which can also be seen in the case of $c = 1.3$.

As discussed in section 2, the simulation time of 500,000 was chosen because nearly all simulations have reached a steady state. The cases highlighted in **Figure 3D** represent the exceptions, and there is no guarantee that these simulations will ever converge. The figure shows that $c = \{0.96, 0.98\}$ are the only cases that give a mixture of aggregated and non-aggregated results.

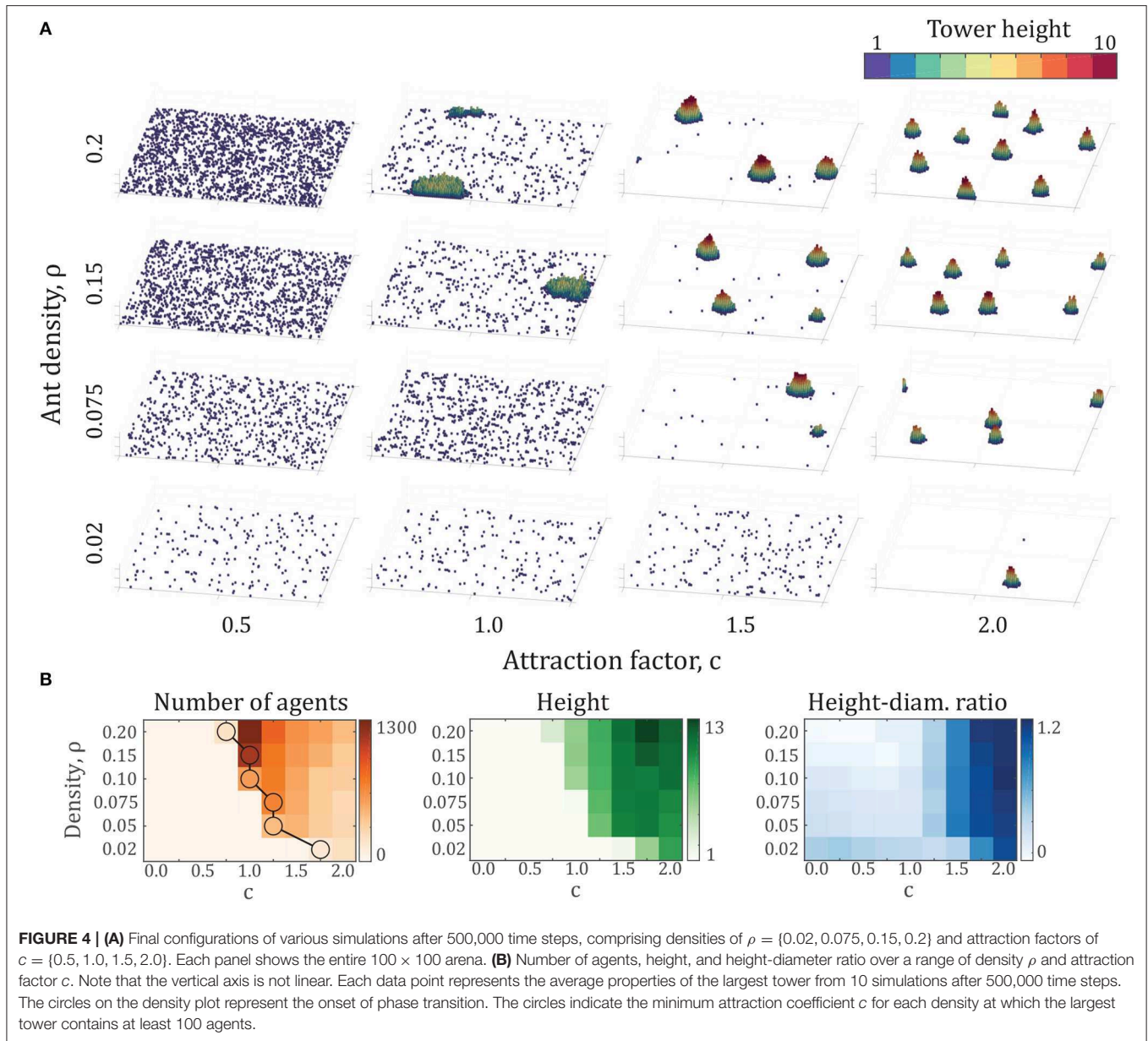
3.3. The Effect of Density

The parameters varied up to this point in the model represent entirely behavioral parameters, that is, those associated with

the decision-making of individuals. While these parameters are testable within multi-agent robotic examples, they do not represent a variable that can be systematically changed in experiments with live fire ants, or robots, in order to test the predictions of the model. To develop a set of testable predictions, we turn to explore the parameter of density of agents, ρ .

In our model, density is varied by changing the number of individuals in a fixed arena size. The computational complexity of the model is $\mathcal{O}(N^2)$, so practical limits of computational time place an upper bound on density we explore here. In a $100 \times 100 \times \infty$ arena, our test set is $N = \{200, 500, 750, 1000, 1500, 2000\}$ which corresponds to the densities $\rho = \{0.02, 0.05, 0.75, 0.1, 0.15, 0.2\}$. We will use unlocking and locking parameters of $P_u = 0.2, k_{nl} = \frac{1}{26}$ for consistency with section 3.2.

The results of these simulations are presented in **Figure 4**, showing several key differences and similarities across densities. As density increases, less attraction is required for tower formation. The data points highlighted by circles in **Figure 4B** show the critical attraction ratio c^* , which represents the minimum value of c for which the largest aggregation is at least 100 individuals, representing the onset of the phase transition. This result also implies that there exists a critical density across a range of attraction factors, below which no tower formation occurs. Another key result is that the largest towers, in terms of number of agents per tower, occur shortly after the transition from no towers at all.



Beyond this point, tower height and ratio increase while number of agents decreases. Finally, tower shape remains close to constant across densities, particularly in height-diameter ratio.

3.4. Moving Towers

The introduction of unlocking probability effectively adds noise to the system (equivalent to higher temperatures in thermodynamic systems), which allows towers to move. Agents locking on one side of the tower while others unlock on the other side can lead to tower motion. Traces of the center of area for each tower from two example simulations may be seen in **Figure 5B**. To quantify this phenomenon, we consider the motion of towers as a Brownian random walk and investigate the

diffusion coefficient of each tower. The diffusion coefficient (D) for a Brownian random walk follows the relationship,

$$\text{MSD} = 2Dt,$$

$$\text{MSD} = \frac{1}{T-t} \sum_{t_0=0}^{T-t} |\mathbf{x}(t+t_0) - \mathbf{x}(t_0)|^2, \quad (3)$$

for each trajectory of length T . Therefore, we measure the mean square displacement (MSD) of each tower in each simulation over a variety of times, $t = \{0, 250, 500, \dots, 12, 500\}$, and perform a linear fit for each tower trajectory. The average slope of these lines is then twice the diffusion coefficient (**Figure 5A**).

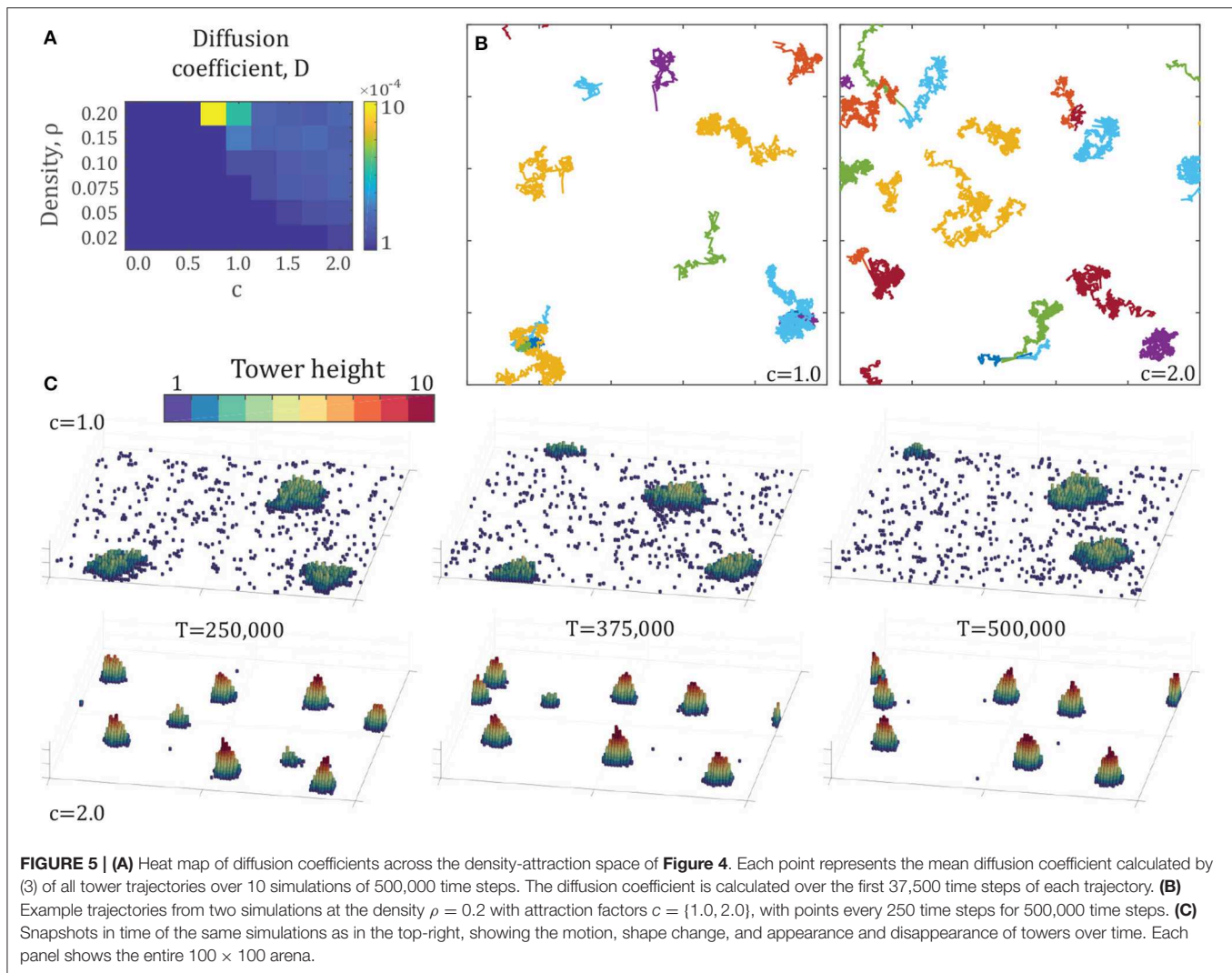


FIGURE 5 | (A) Heat map of diffusion coefficients across the density-attraction space of **Figure 4**. Each point represents the mean diffusion coefficient calculated by (3) of all tower trajectories over 10 simulations of 500,000 time steps. The diffusion coefficient is calculated over the first 37,500 time steps of each trajectory. **(B)** Example trajectories from two simulations at the density $\rho = 0.2$ with attraction factors $c = \{1.0, 2.0\}$, with points every 250 time steps for 500,000 time steps. **(C)** Snapshots in time of the same simulations as in the top-right, showing the motion, shape change, and appearance and disappearance of towers over time. Each panel shows the entire 100×100 arena.

These results show that the maximum diffusion occurs in the highest density regime, and for the lowest attraction parameters that generate aggregations, particularly for $\rho = 0.2$ at $c = 0.75$ and $c = 1.0$. These towers have lower height-diameter ratios, as seen in **Figure 4B**, which leads to a larger proportion of agents on the surface of the tower, and therefore a higher probability that individuals on the surface will be unlocking. The towers at $c = 0.75$ have a smaller number of agents than those of $c = 1.0$, which leads to an even higher proportion of individuals on the surface. This is illustrated in **Figure 5C**, showing the time-evolution of tower configurations for two example simulations (see also **Supplementary Videos S5, S6**).

3.5. Tower Optimization

One question that still remains is, what parameter values are optimal for tower building? To answer this, we need to think about what may constitute optimal. It may be that the optimal tower reaches as high as possible, which would, in practice, allow as many agents as possible to attach to a support structure. Or, for robotics applications, this would allow the tower to reach

higher heights. On the other hand, it may be best to include as many individuals as possible in the tower, and the optimal tower would be the one that includes every single agent in the tower. As observed in Phonekeo et al. (2017), fire ants built towers that equally distribute load among the individuals. Therefore, an optimal tower from their perspective may be one that optimizes for load distribution. In this section, we use a genetic algorithm to explore optimal tower building considering each of these optimization targets.

To search for an optimal tower, we employ the Covariance Matrix Adaptation-Evolutionary Strategy (CMA-ES) algorithm developed by Hansen et al. (2003). This algorithm randomly generates parameter sets within the search space and evaluates a cost function for each parameter set. From the results of this function evaluation, it updates the covariance matrix to expand in the direction of the most optimal value. Using the updated covariance matrix, the algorithm generates new parameter sets and repeats the process until convergence, generally defined as finding a parameter set with a cost function below some threshold.

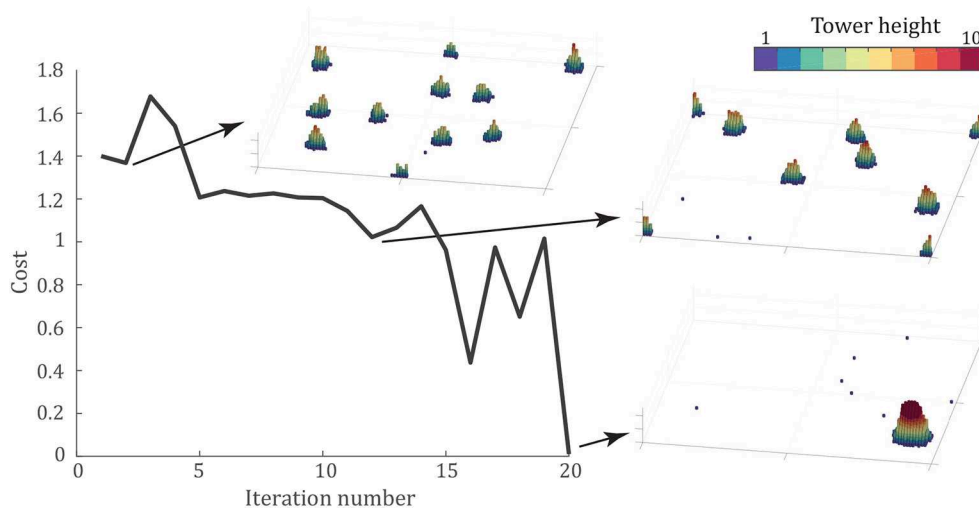


FIGURE 6 | Optimization of the cost function given by (4) using the CMA-ES algorithm (Hansen et al., 2003). Panels show the final configuration of the best simulation at iterations 2, 12, and 20 after 50,000 time steps. Each panel shows the entire 100×100 arena. The algorithm converges in 20 iterations, to the parameter set $P_u = 0.938$, $k_{nl} = 0.029$, and $c = 2.56$, with the optimal configuration shown in the bottom-right.

We applied the CMA-ES algorithm to the tower-building model introduced above, using the average final properties of three trials for each parameter set across 10 parameter sets per iteration. For the optimization, we choose a cost function defining the optimal tower as the largest tower, both in terms of tower height and number of individuals within the tower. Therefore, the cost function is given by,

$$f = \left(1 - \frac{N_{tower}}{N_{max}}\right) + \max \left\{0, \left(1 - \frac{h_{tower}}{h_{max}}\right)\right\}, \quad (4)$$

where N_{tower} and h_{tower} represent the number of individuals and height of the largest tower, N_{max} is the number of individuals in the simulation, and h_{max} is a prescribed maximum height. The height term is included to ensure that the results are effectively tower-like, preventing the optimal tower from simply achieving a large, wide aggregation. From the results of the attraction sweep in **Figure 3A**, we observe that $h_{max} = 14$ is an approximate upper bound on tower height, so it is therefore chosen as h_{max} for the purpose of this optimization. Note that a tower height of $h_{tower} \geq h_{max}$ results in a zero second term, and the simulation therefore allows for a taller tower. For the purposes of optimization, we reduce the simulation time to 50,000 time steps. This serves the practical role of making iterated simulation possible, but also places an effective minimization of convergence time. Therefore, we are optimizing for a tower that maximizes both height and number of agents quickly (within 50,000 time steps).

Figure 6 shows the progression of the minimum cost at each iteration of the CMA-ES algorithm along with snapshots of intermediate results to show the algorithm's progress. The optimal tower occurs for the parameters, $P_u = 0.938$, $k_{nl} = 0.029$, and $c = 2.56$, which led to a tower of 993 agents reaching 16 agents tall after 50,000 time steps. The final cost function, averaged over three trials, was $f = 0.01$.

A video of one simulation with these parameters is shown in **Supplementary Video S7**.

The CMA-ES optimization code of Hansen et al. (2003) applied to the present model may allow future research and consideration of other conditions of optimal tower-building. For example, when designing a robotic system where each individual robot has a maximum load capability, it may be necessary to calculate the maximum load experienced by an individual in the tower and add that term to cost function.

4. DISCUSSION

In this work, we have extended a previously proposed set of local rules to replicate the tower-building behavior of red imported fire ants, *Solenopsis invicta*. This model and its insights will allow for the design of control strategies for tower-building swarm robotics and greater insight into the collective behavior of social insects. The results presented above show that individuals moving under the influence of local attraction are able to form large towers. We find that an attractive force is necessary for significant tower-building and show the impacts of this attractive force over a range of locking and unlocking parameters as well as a range of densities. We find that the system contains a sudden phase transition as the attraction parameter is varied, and that this phase transition is density-dependent. Finally, the largest towers, in both height and number of individuals, occur with a combination of very strong attraction and highly probable unlocking.

On the other hand, without attraction, no towers form, as shown in the $c = 0$ case of **Figure 2** and **Supplementary Video S2** and discussed further in the **Appendix 1** and **Figure S1**. The effective force of attraction may also be thought of as a desire of the ants to climb, because

the tallest available square to move toward will also have the most neighbors.

Near the phase transition, a critical slowing down occurs, and there are parameter sets that do not result in tower formation within a simulation time of 500,000 time steps. This critical slowing down is reminiscent of other examples of systems with phase transitions, such as the spin-glass model, the Ising model, and molecular dynamics models (Dasgupta et al., 1979; Hu, 2013). Further from the phase transition ($c \gg c^*$), towers form rapidly, but the possibility exists for these towers to encounter one another and merge into larger towers. The number of individuals in the tower and tower motion are largest just after the phase transition, but the largest height occurs with stronger attraction. Phase transitions have previously been observed experimentally and computationally in other ant and insect systems, such as in Pharaoh's ant foraging (Beekman et al., 2001) and in marching desert locusts (Buhl et al., 2006).

Our results also illustrate the exploration-exploitation trade-off, which balances attraction forces with random movement and unlocking events. Following this trade-off, stronger attraction may lead to higher towers with fewer individuals, as the attraction rapidly draws individuals from the edge of the aggregation toward the center of the tower, and therefore upward. This balance of unlock probability and attraction is found through the combined optimization of number of individuals and tower height, which discovered that with an unlock probability of $P_u = 0.938$ and an attraction of $c = 2.56$, it is possible to include nearly all of the individuals in a simulation, with a tower reaching a height of 16 layers. This large unlock probability of the largest towers in our simulations connects with the observation from Phonekeo et al. (2017) that, in the experimental system, the fire ants are constantly rebuilding their tower and circulating ants throughout the tower. The work of Phonekeo et al. (2017) showed that fire ants build towers of constant load, and future optimization work could incorporate the load experienced by each individual to achieve towers that prioritize stability. More refined ant models may also incorporate the mechanics and viscoelastic properties of fire ant aggregations (Tennenbaum et al., 2016), which are observed to change depending on the number of active ants, such as the free ants included in the present model (Tennenbaum and Fernandez-Nieves, 2017).

The results of the parameter sweep in density values showed both similarities and differences across densities. In general, for a fixed attraction ratio c , the tower height-diameter ratio remains fairly constant, even as the numbers of agents per tower and tower height vary. The biggest difference across densities is the change in critical attraction parameter, c^* . These observations lead to testable hypotheses for animal experiments. Below a certain density threshold, tower formation should cease, due to the move past the critical attraction. Additionally, the height-diameter ratio should remain constant across a large range of densities. Finally, we have shown that the towers built in our simulations move over time, with a diffusion coefficient that is dependent on both attraction and density, and should be taken into account when considering practical application to robotics.

This work also lays the groundwork for future robotic studies, where robots are able to build towers out of themselves in a

manner similar to, for example, the M-blocks of Romanishin et al. (2015) or the Roombots of Spröwitz et al. (2014), which have also been proposed for bridge-building applications (Nguyen-Duc et al., 2019). The tower is a ubiquitous structure in building, and designing rigorous control strategies for tower-building represents a fundamental starting point toward fully autonomous, locally-sensed swarm building applications. In practice, a tower of robots could be useful in the case of, for example, seeing over obstacles, providing scaffolding for climbing, or clearly marking a location of interest. Robotic tower-builders would need to have the following capabilities: sense neighbors, climb onto and off of one another, and support appropriate loads. At the moment, there is no robot with all of these capabilities, and we believe that this would be a promising avenue for future robotics research. The control strategies introduced in the present study could also be further modified to more closely replicate experimentally-observed fire ant behavior, developing a control strategy for interacting with a support structure.

DATA AVAILABILITY STATEMENT

All datasets generated for this study are available upon request.

AUTHOR CONTRIBUTIONS

NM, JC, TS, and JL developed the initial version of the present model as a project in a course taught by OP. GN refined the model, conducted the simulations, and wrote and edited the manuscript. OP supervised the research and edited the manuscript.

FUNDING

The study was conducted with institutional funding.

ACKNOWLEDGMENTS

We thank Prof. David L. Hu, and members of the Peleg lab for insightful discussions, and the BioFrontiers Institute for the utmost support.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2020.00025/full#supplementary-material>

Supplementary Video S1 | The diffusion-limited aggregation case of the model, with $P_u = 0, k_{rl} = 1, c = 0$. The video is shown at a speed of 120 time steps per second.

Supplementary Video S2 | Simulation in which no aggregations form, with $P_u = 0.2, k_{rl} = \frac{1}{26}, c = 0$. The video is shown at a speed of 10,000 time steps per second.

Supplementary Video S3 | Simulation in which large, wide aggregations form, with $P_u = 0.2, k_{rl} = \frac{1}{26}, c = 1$. The video is shown at a speed of 10,000 time steps per second.

Supplementary Video S4 | Simulation in which many steep towers form, with $P_U = 0.2$, $k_{rl} = \frac{1}{26}$, $c = 2$. The video is shown at a speed of 10,000 time steps per second.

Supplementary Video S5 | Simulation with large, wide moving aggregations in a dense environment, with $P_U = 0.2$, $k_{rl} = \frac{1}{26}$, $c = 1$, and $N = 2,000$ individuals. The video is shown at a speed of 10,000 time steps per second.

Supplementary Video S6 | Simulation with many steep moving aggregations in a dense environment, with $P_U = 0.2$, $k_{rl} = \frac{1}{26}$, $c = 2$, and $N = 2,000$ individuals. The video is shown at a speed of 10,000 time steps per second.

Supplementary Video S7 | Simulation of the results of tower optimization, with $P_U = 0.938$, $k_{rl} = 0.029$, $c = 2.56$. The video is shown at a speed of 2,500 time steps per second.

Supplementary Code S8 | Three MATLAB code files, included in a .zip file. TowerSimulation.m provides a function to run a single simulation, TowerAnalysis.m provides the analysis of the resulting towers, and TowerVideo.m provides the code used to visualize the results of each simulation. A maintained repository of these codes is available at: <https://github.com/peleg-lab/TowerBuilding>.

REFERENCES

- Ame, J.-M., Rivault, C., and Deneubourg, J.-L. (2004). Cockroach aggregation based on strain odour recognition. *Anim. Behav.* 68, 793–801. doi: 10.1016/j.anbehav.2004.01.009
- Anderson, C., Theraulaz, G., and Deneubourg, J.-L. (2002). Self-assemblages in insect societies. *Insect. Soc.* 49, 99–110. doi: 10.1007/s00040-002-8286-y
- Bak, P. (1996). *How Nature Works: The Science of Self-organized Criticality*. New York, NY: Springer-Verlag.
- Beekman, M., Sumpter, D. J., and Ratnieks, F. L. (2001). Phase transition between disordered and ordered foraging in pharaoh's ants. *Proc. Natl. Acad. Sci. U.S.A.* 98, 9703–9706. doi: 10.1073/pnas.161285298
- Bochynek, T., Tanner, J. L., Meyer, B., and Burd, M. (2017). Parallel foraging cycles for different resources in leaf-cutting ants: a clue to the mechanisms of rhythmic activity. *Ecol. Entomol.* 42, 849–852. doi: 10.1111/een.12437
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY: Oxford University Press.
- Bonabeau, E., Guérin, S., Snyers, D., Kuntz, P., and Theraulaz, G. (2000). Three-dimensional architectures grown by simple 'stigmergic' agents. *Biosystems* 56, 13–32. doi: 10.1016/S0303-2647(00)00067-8
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* 7, 1–41. doi: 10.1007/s11721-012-0075-2
- Buffin, A., and Pratt, S. (2016). Cooperative transport by the ant novomessor cockerelli. *Insect. Soc.* 63, 429–438. doi: 10.1007/s00040-016-0486-y
- Buhl, J., Sumpter, D. J., Couzin, I. D., Hale, J. J., Despland, E., Miller, E. R., et al. (2006). From disorder to order in marching locusts. *Science* 312, 1402–1406. doi: 10.1126/science.1125142
- Couzin, I. D., and Franks, N. R. (2003). Self-organized lane formation and optimized traffic flow in army ants. *Proc. R. Soc. Lond. B Biol. Sci.* 270, 139–146. doi: 10.1098/rspb.2002.2210
- Couzin, I. D., and Krause, J. (2003). "Self-organization and collective behavior in vertebrates," in *Advances in the Study of Behavior* eds P. J. B. Slater, J. S. Rosenblatt, C. T. Snowdon, and T. J. Roper (San Diego, CA: Academic Press), 32, 1–75.
- Dasgupta, C., Ma, S.-K., and Hu, C.-K. (1979). Dynamic properties of a spin-glass model at low temperatures. *Phys. Rev. B* 20:3837. doi: 10.1103/PhysRevB.20.3837
- Del Dottore, E., Sadeghi, A., Mondini, A., Mattoli, V., and Mazzolai, B. (2018). Toward growing robots: a historical evolution from cellular to plant-inspired robotics. *Front. Robot. AI* 5:16. doi: 10.3389/frobt.2018.00016
- Deneubourg, J. L., Grégoire, J. C., and Le Fort, E. (1990). Kinetics of larval gregarious behavior in the bark beetle *Dendroctonus micans* (coleoptera: Scolytidae). *J. Insect Behav.* 3, 169–182. doi: 10.1007/BF01417910
- Deneubourg, J. L., Lioni, A., and Detrain, C. (2002). Dynamics of aggregation and emergence of cooperation. *Biol. Bull.* 202, 262–267. doi: 10.2307/1543477
- Dorigo, M., Bonabeau, E., and Theraulaz, G. (2000). Ant algorithms and stigmergy. *Future Gen. Comput. Syst.* 16, 851–871. doi: 10.1016/S0167-739X(00)00042-X
- Franks, N. R. (1989). Thermoregulation in army ant bivouacs. *Physiol. Entomol.* 14, 397–404. doi: 10.1111/j.1365-3032.1989.tb01109.x
- Garnier, S., Murphy, T., Lutz, M., Hurme, E., Leblanc, S., and Couzin, I. D. (2013). Stability and responsiveness in a self-organized living architecture. *PLoS Comput. Biol.* 9:e1002984. doi: 10.1371/journal.pcbi.1002984
- Goss, S., Aron, S., Deneubourg, J.-L., and Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften* 76, 579–581. doi: 10.1007/BF00462870
- Groß, R., Tuci, E., Dorigo, M., Bonani, M., and Mondada, F. (2006). "Object transport by modular robots that self-assemble," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006* (Orlando, FL: IEEE), 2558–2564.
- Hamann, H. (2018). *Swarm Robotics: A Formal Approach*. Berlin: Springer.
- Hansen, N., Müller, S. D., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol. Comput.* 11, 1–18. doi: 10.1162/106365603321828970
- Hu, C.-K. (2013). Slow dynamics in proteins and polymer chains. *AIP Conf. Proc.* 1518, 541–550. doi: 10.1063/1.4794631
- Hughes, D., Heckman, C., and Correll, N. (2019). Materials that make robots smart. *Int. J. Robot. Res.* 38, 1338–1351. doi: 10.1177/0278364919856099
- Jeanson, R., Rivault, C., Deneubourg, J.-L., Blanco, S., Fournier, R., Jost, C., et al. (2005). Self-organized aggregation in cockroaches. *Anim. Behav.* 69, 169–180. doi: 10.1016/j.anbehav.2004.02.009
- Kastberger, G., Weihmann, F., and Hoetzl, T. (2011). "Self-assembly processes in honeybees: the phenomenon of shimmering," in *Honeybees of Asia* eds H. R. Hepburn and S. E. Radloff (Berlin; Heidelberg: Springer-Verlag), 397–443.
- Kerth, G. (2008). Causes and consequences of sociality in bats. *Bioscience* 58, 737–746. doi: 10.1641/B580810
- Levine, H., Tsimring, L., and Kessler, D. (1997). Computational modeling of mound development in dictyostelium. *Phys. D Nonlin. Phenom.* 106, 375–388. doi: 10.1016/S0167-2789(97)00038-9
- Lioni, A., Sauwens, C., Theraulaz, G., and Deneubourg, J.-L. (2001). Chain formation in *Oecophylla longinoda*. *J. Insect Behav.* 14, 679–696. doi: 10.1023/A:1012283403138
- Mishra, S., Tunström, K., Couzin, I. D., and Huepe, C. (2012). Collective dynamics of self-propelled particles with variable speed. *Phys. Rev. E* 86:011901. doi: 10.1103/PhysRevE.86.011901
- Mlot, N. J., Tovey, C., and Hu, D. L. (2012). Dynamics and shape of large fire ant rafts. *Commun. Integr. Biol.* 5, 590–597. doi: 10.4161/cib.21421
- Mlot, N. J., Tovey, C. A., and Hu, D. L. (2011). Fire ants self-assemble into waterproof rafts to survive floods. *Proc. Natl. Acad. Sci. U.S.A.* 108, 7669–7673. doi: 10.1073/pnas.1016658108
- Navarro, R. M., and Fielding, S. M. (2015). Clustering and phase behaviour of attractive active particles with hydrodynamics. *Soft Matter* 11, 7525–7546. doi: 10.1039/C5SM01061F
- Nguyen-Duc, J., Mutlu, M., Hauser, S., Barnerdino, A., and Ijspeert, A. (2019). "Cooperative bridge building by self-reconfigurable modular robots based on ants' stigmergic behaviour," in *9th International Symposium on Adaptive Motion of Animals and Machines (AMAM 2019)* (Lausanne).
- Panagiotou, A., Curtin, M., Toki, H., Scott, D., and Siemens, P. (1984). Experimental evidence for a liquid-gas phase transition in nuclear systems. *Phys. Rev. Lett.* 52:496. doi: 10.1103/PhysRevLett.52.496
- Phonekeo, S., Mlot, N., Monastkova, D., Hu, D. L., and Tovey, C. (2017). Fire ants perpetually rebuild sinking towers. *R. Soc. Open Sci.* 4:170475. doi: 10.1098/rsos.170475
- Reid, C. R., Lutz, M. J., Powell, S., Kao, A. B., Couzin, I. D., and Garnier, S. (2015). Army ants dynamically adjust living bridges in response to a cost-benefit trade-off. *Proc. Natl. Acad. Sci. U.S.A.* 112, 15113–15118. doi: 10.1073/pnas.1512241112

- Reynaert, S., Moldenaers, P., and Vermant, J. (2006). Control over colloidal aggregation in monolayers of latex particles at the oil-water interface. *Langmuir* 22, 4936–4945. doi: 10.1021/la060052n
- Reynolds, C. W. (1987). “Flocks, herds and schools: a distributed behavioral model,” in *ACM SIGGRAPH Computer Graphics*, ed M. C. Stone (Anaheim, CA: ACM), Vol. 21, 25–34.
- Romanishin, J. W., Gilpin, K., Claici, S., and Rus, D. (2015). “3D m-blocks: self-reconfiguring robots capable of locomotion via pivoting in three dimensions,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA: IEEE), 1925–1932.
- Rovere, M., Heermann, D., and Binder, K. (1990). The gas-liquid transition of the two-dimensional lennard-jones fluid. *J. Phys. Condens. Matter* 2:7009. doi: 10.1088/0953-8984/2/33/013
- Roverud, R. C., and Chappell, M. A. (1991). Energetic and thermoregulatory aspects of clustering behavior in the neotropical bat *noctilio albiventris*. *Physiol. Zool.* 64, 1527–1541. doi: 10.1086/physzool.64.6.30158228
- Şahin, E. (2004). “Swarm robotics: from sources of inspiration to domains of application,” in *International Workshop on Swarm Robotics* (Santa Monica, CA: Springer), 10–20.
- Şahin, E., Labella, T. H., Trianni, V., Deneubourg, J.-L., Rasse, P., Floreano, D., et al. (2002). “Swarm-bot: pattern formation in a swarm of self-assembling mobile robots,” in *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4 (Hammamet: IEEE), 6.
- Seeley, T. D. (2010). *Honeybee Democracy*. Princeton, NJ: Princeton University Press.
- Shapiro, L. G. (1996). “Connected component labeling and adjacency graph construction,” in *Topological Algorithms for Digital Image Processing, Volume 19 of Machine Intelligence and Pattern Recognition*, eds T. Y. Kong and A. Rosenfeld (Amsterdam: Elsevier Science, BV), 1–30.
- Spröwitz, A., Moeckel, R., Vespignani, M., Bonardi, S., and Ijspeert, A. J. (2014). Roombots: a hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot. *Robot. Auton. Syst.* 62, 1016–1033. doi: 10.1016/j.robot.2013.08.011
- Strömbom, D., and Dussutour, A. (2018). Self-organized traffic via priority rules in leaf-cutting ants. *PLoS Comput. Biol.* 14:e1006523. doi: 10.1371/journal.pcbi.1006523
- Swissler, P., and Rubenstein, M. (2018). “Fireant: a modular robot with full-body continuous docks,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane QLD: IEEE), 6812–6817.
- Tennenbaum, M., and Fernandez-Nieves, A. (2017). Activity-driven changes in the mechanical properties of fire ant aggregations. *Phys. Rev. E* 96:052601. doi: 10.1103/PhysRevE.96.052601
- Tennenbaum, M., Liu, Z., Hu, D., and Fernandez-Nieves, A. (2016). Mechanics of fire ant aggregations. *Nat. Mater.* 15:54. doi: 10.1038/nmat4450
- Theraulaz, G., and Bonabeau, E. (1995). Modelling the collective building of complex architectures in social insects with lattice swarms. *J. Theor. Biol.* 177, 381–400. doi: 10.1006/jtbi.1995.0255
- Theraulaz, G., Bonabeau, E., Nicolis, S. C., Solé, R. V., Fourcassié, V., Blanco, S., et al. (2002). Spatial patterns in ant colonies. *Proc. Natl. Acad. Sci. U.S.A.* 99, 9645–9649. doi: 10.1073/pnas.152302199
- Umeda, T., and Inouye, K. (1999). Theoretical model for morphogenesis and cell sorting in dictyostelium discoideum. *Phys. D Nonlin. Phenom.* 126, 189–200. doi: 10.1016/S0167-2789(98)00273-5
- Vernerey, F., Benet, E., Blue, L., Fajrial, A., Sridhar, S. L., Lum, J., et al. (2018). Biological active matter aggregates: inspiration for smart colloidal materials. *Adv. Colloid Interface Sci.* 263, 38–51. doi: 10.1016/j.cis.2018.11.006
- Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., and Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.* 75:1226. doi: 10.1103/PhysRevLett.75.1226
- Vlasov, V. A. (2019). Activated complex theory of nucleation. *Eur. Phys. J. E* 42:36. doi: 10.1140/epje/i2019-11797-7
- Voorhees, P. W. (1985). The theory of ostwald ripening. *J. Stat. Phys.* 38, 231–252. doi: 10.1007/BF01017860
- Waters, A., Blanchette, F., and Kim, A. D. (2012). Modeling huddling penguins. *PLoS ONE* 7:e50277. doi: 10.1371/journal.pone.0050277
- Werfel, J., Petersen, K., and Nagpal, R. (2014). Designing collective behavior in a termite-inspired robot construction team. *Science* 343, 754–758. doi: 10.1126/science.1245842
- Wilensky, U., and Rand, W. (2015). *An Introduction to Agent-Based Modeling*. Cambridge, MA: MIT Press.
- Witten, T. Jr., and Sander, L. M. (1981). Diffusion-limited aggregation, a kinetic critical phenomenon. *Phys. Rev. Lett.* 47:1400. doi: 10.1103/PhysRevLett.47.1400

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Nave, Mitchell, Chan Dick, Schuessler, Lagarrigue and Peleg. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Phenotypic Plasticity Provides a Bioinspiration Framework for Minimal Field Swarm Robotics

Edmund R. Hunt^{1,2*†}

¹ Department of Engineering Mathematics, University of Bristol, Bristol, United Kingdom, ² Bristol Robotics Laboratory, University of the West of England, Bristol, United Kingdom

OPEN ACCESS

Edited by:

Heiko Hamann,
University of Lübeck, Germany

Reviewed by:

Melanie E. Moses,
University of New Mexico,
United States
Mauro Birattari,
Université libre de Bruxelles, Belgium
Farshad Arvin,
University of Manchester,
United Kingdom

*Correspondence:

Edmund R. Hunt
edmund.hunt@bristol.ac.uk

†ORCID:

Edmund R. Hunt
orcid.org/0000-0002-9647-124X

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 22 November 2019

Accepted: 11 February 2020

Published: 16 March 2020

Citation:

Hunt ER (2020) Phenotypic Plasticity
Provides a Bioinspiration Framework
for Minimal Field Swarm Robotics.
Front. Robot. AI 7:23.
doi: 10.3389/frobt.2020.00023

The real world is highly variable and unpredictable, and so fine-tuned robot controllers that successfully result in group-level “emergence” of swarm capabilities indoors may quickly become inadequate outside. One response to unpredictability could be greater robot complexity and cost, but this seems counter to the “swarm philosophy” of deploying (very) large numbers of simple agents. Instead, here I argue that bioinspiration in swarm robotics has considerable untapped potential in relation to the phenomenon of phenotypic plasticity: when a genotype can produce a range of distinctive changes in organismal behavior, physiology and morphology in response to different environments. This commonly arises following a natural history of variable conditions; implying the need for more diverse and hazardous simulated environments in offline, pre-deployment optimization of swarms. This will generate—indicate the need for—plasticity. Biological plasticity is sometimes irreversible; yet this characteristic remains relevant in the context of minimal swarms, where robots may become mass-producible. Plasticity can be introduced through the greater use of adaptive threshold-based behaviors; more fundamentally, it can link to emerging technologies such as smart materials, which can adapt form and function to environmental conditions. Moreover, in social animals, individual heterogeneity is increasingly recognized as functional for the group. Phenotypic plasticity can provide meaningful diversity “for free” based on early, local sensory experience, contributing toward better collective decision-making and resistance against adversarial agents, for example. Nature has already solved the challenge of resilient self-organisation in the physical realm through phenotypic plasticity: swarm engineers can follow this lead.

Keywords: phenotypic plasticity, reaction norms, swarm diversity, resilience, minimal robotics, swarm robotics

INTRODUCTION

The self-organized societies of social insects such as ants are well-known in swarm robotics (Şahin, 2005); yet they could be the “tip of the iceberg” of available bioinspiration. Here, I focus specifically on the general concept of *phenotypic plasticity* as a powerful, complementary framework for thinking about real-world deployment of minimal robot swarms. In fact, social insects are prime exhibitors of phenotypic plasticity (Kennedy et al., 2017), but it is widespread and of fundamental importance in the rest of the natural world. In brief, I argue the following main points:

1. Plasticity is typically selected for by evolution following a natural history of unstable environmental conditions. In offline evolutionary swarm optimization, simulated environments need to be more heterogeneous and hazardous to generate and understand the value of plasticity.
2. In the context of large numbers of agents, elements of this plasticity could be (partially) irreversible, as in nature. This could be further enabled by cost-effective expendability, up to and including recyclable or biodegradable robots.
3. In addition to the value of individual plasticity for responding to environmental variation, otherwise unremarkable variation in response thresholds (for example) can contribute to adaptive group-level diversity; swarm engineers can exploit this.

I first provide some background perspective on swarm robotics before introducing the biological phenomenon of phenotypic plasticity.

Background: The “Swarm Principle” of Individual-Level Simplicity

Swarm robotics is predicated on the idea that large numbers of agents working collectively can solve tasks that would be impossible for a single individual (Hamann, 2018). It is specifically inspired by biology in that it relies on self-organization (Camazine et al., 2001) as the mechanism of coordination, particularly as seen in social insects (Şahin, 2005). This includes concepts such as stigmergy (e.g., Hunt et al., 2019a). Closely allied to this is the reliance on *emergence* of swarm problem-solving capabilities that cannot be reduced to, or predicted from, individual-level components (Şahin, 2005; Bjercknes et al., 2007; Brambilla et al., 2013).

As technology continues to develop, with ever-advancing computer processing power and methods in artificial intelligence, the temptation may be to build swarms of agents that are individually highly complex both in their hardware and controllers. However, this would not align with the “swarm principle” of relying on emergence to do the “heavy lifting” of solving the task. It would also defeat the object in “complexity engineering” of maintaining low-level understandability (Frei and Giovanna, 2012). Finally, it may be prohibitive in terms of cost, when real-world environments have hazards resulting in a risk—or even an expectation—of robots being lost or destroyed. Instead, swarm controllers are classically based on reactive control (Hamann, 2018), based on simple reflexes to a stimulus (e.g., Walter, 1950; Mitrano et al., 2019), or taking into account an internal state (the model-based reflex agent of Russell and Norvig, 1995, for example Nouyan et al., 2009). This “behavior-based robotics” (Arkin, 1998) is in keeping with studies of reaction thresholds in biology (Bonabeau et al., 1999). It is also compatible with relatively simple and affordable hardware that can be easily understood: for example the “e-puck” (Mondada et al., 2009), “Kilobot” (Rubenstein et al., 2012), and “Crazyflie” (McGuire et al., 2019). There is still relatively limited real-world swarm deployment (e.g., Schmickl et al., 2011; Duarte et al., 2016): there is a clear opportunity to shape the design principles for minimal swarms.

Previous Examples of Adaptation in Homogeneous Robot Swarms

There are several examples in the swarm robotics literature in which individual robots, though identically programmed with the same controller, end up behaving differently according to their experience of the environment. I briefly group these according to three prominent approaches, before going on to explain the complementarity of the proposed approach.

Off-Line (Pre-deployment) Evolutionary Optimization

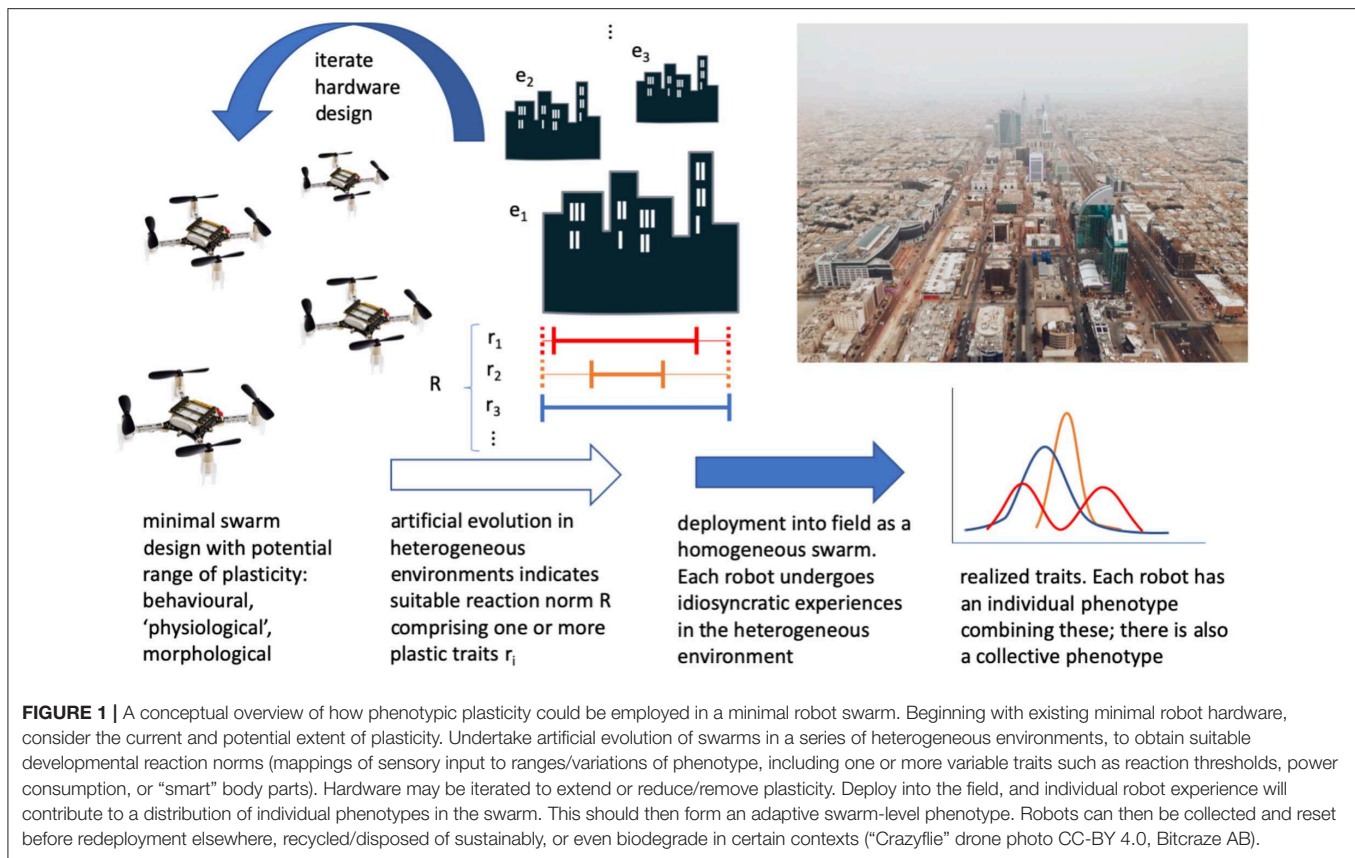
Designing emergent (Matić, 1993) and adaptive (Matić, 1995) group behaviors is challenging, and so one can use evolutionary optimization in simulation before deployment (Dorigo et al., 2004; Trianni, 2008; Hecker and Moses, 2015; Birattari et al., 2019). In this way, adaptation of behavior can be seen in task specialization, for example, as an effective group-level strategy (Ferrante et al., 2015), though its effectiveness is tuned to the particular simulated environment. Furthermore, the simulated environments employed in evolutionary robotics can be rather simple and homogeneous. As a result, there can be little in the way of a mechanism to generate plasticity, as it is not rewarded by the artificial evolutionary process. Including sufficient heterogeneity in the class of simulated environments is indispensable to identifying a suitable variety and extent of plasticity for swarm robots (Figure 1).

On-Line (On-Deployment) Evolutionary Optimization

Embodied evolutionary robotics is a promising avenue for real-world deployment (Trueba et al., 2011; Haasdijk et al., 2014; Jones et al., 2019) but in practice the requisite computing power may be a step away from the minimal robotics needed for swarm ubiquity. Evolutionary approaches (off- or on-line) could struggle in the field, owing to unanticipated circumstances or merely because of the so-called “reality gap” between the world and (inner) simulation (Brooks, 1992; Jakobi et al., 1995).

Learning (On-Deployment)

Learning is an example of behavioral plasticity. For example, if one simulates improved task performance through repetition there can be emergent task specialization (Brutschy et al., 2012). Task sequencing has been demonstrated at run-time without prior knowledge of the correct ordering, demonstrating a form of reinforcement learning, albeit with abstractions of the tasks themselves (Garattoni and Birattari, 2018). In practice, robot learning tends to employ (evolved) neural networks (Nolfi et al., 1994; Floreano and Mondada, 1996; Nolfi and Floreano, 2000; Nitschke et al., 2012; Hüttenrauch et al., 2018), so-called neuro-evolution methods. Neural network-based approaches can have difficulty in scaling to more complex problems (Brambilla et al., 2013); and again, for truly minimal swarms, this may be a step toward undue computational complexity. I suggest “personality” adaptation as an example minimal bioinspired approach to learning (section Behavioral Plasticity).



Phenotypic Plasticity: Evolving Adaptive Reaction Norms

Broadly defined, phenotypic plasticity is the ability of an organism’s genotype to produce different phenotypes in response to different environmental conditions (Kelly et al., 2012). This includes behavioral, physiological, and morphological plasticity as I later describe in their respective sections (see also **Figure 2**). These are ordered by how rapidly an adjustment is typically made through that plasticity mode. Plasticity varies, as we see in social insects: some are resilient to environmental change (e.g., invasive ants; Holway et al., 2002), while others such as bees struggle to cope with e.g., habitat loss, novel toxins, or pathogens (Goulson et al., 2015). Its importance may in part depend on mobility: for instance, it is particularly important in plants, which are unable to change their environment (Schlichting, 1986). Early experience is often key to phenotypic development (e.g., Weaver et al., 2004), which can be seen as a form of “memory” of the environment to which the organism (or agent) is exposed in the initial phase of its life (deployment).

The term developmental reaction norm (DRN) describes the range of phenotypes generated by a given genotype (“controller,” smart materials, etc.) in response to experienced environmental cues (Schlichting and Pigliucci, 1998). DRNs can themselves be plastic or non-plastic, i.e., the phenotype can remain fixed or change in response to changing environmental conditions. Therefore, there are at least five attributes to DRNs:

amount of plasticity (large/small); pattern of response (e.g., monotonic increase/decrease or more complex reaction curves); rapidity of response; reversibility of response; and competence (possibility) of the developmental system to respond at a certain stage in an organism’s (robot’s) lifetime (Schlichting and Pigliucci, 1998). Moreover, in the “swarm” context, it is worth noting that individuals’ experiences can affect the extent of their plasticity at a given age (Stamps, 2016). This can also contribute to group-level diversity in phenotypic expression. Behavioral plasticity at the level of the whole group can be seen in, for example, the reaction thresholds of harvester ant colonies (Gordon et al., 2011). In social groups individual phenotypes interact, contributing to the complexity of the genotype and phenotype fitness landscapes (Moore et al., 1997; Wolf et al., 1999). The various attributes of developmental reaction norms are, in principle, subject to natural selection (Schlichting and Pigliucci, 1998; Dingemanse et al., 2010), and I propose that for swarm engineers, pre-deployment artificial evolution of DRNs can establish their extent (**Figure 1**). Plasticity occurs in response to environmental cues, so one must also consider the relevant environmental features (physical and social) that will elicit change—and how they will be sensed. For example, local cues about resource distributions can be used to adjust individuals’ foraging parameters (Just and Moses, 2018), and environmental heterogeneity generates variable foraging rates through behavioral plasticity in harvester ants (Beverly et al., 2009).

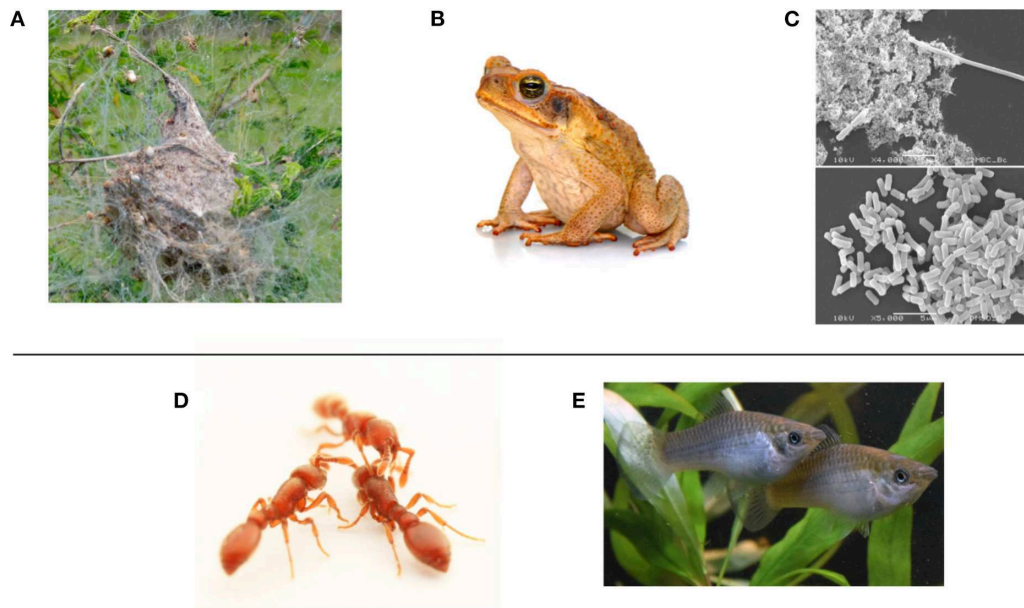


FIGURE 2 | Top row: Examples of phenotypic plasticity, which could be enacted with simple, environmentally-reactive control rules. They are ordered left to right in the typical speed of response. **(A)** Behavioral plasticity in the social spider *Stegodyphus dumicola* allows the colony to maintain a suitable distribution of “bold” and “shy” individuals (photo: Bernard Dupont, CC-BY-SA 2.0). **(B)** Physiological plasticity is observed in the invasive cane toad *Rhinella marina*, as it can adjust its core temperature to live in cool regions (photo: Sam Fraser-Smith, CC-BY-2.0). **(C)** Morphological plasticity in the bacterium *Bacillus cereus*. In the top pane it has undergone filamentation following antibacterial treatment; bottom pane is untreated (photo: Achara Dholvitayakhun, CC BY-SA 4.0). Bottom row: Biologists study clonal organisms to understand how phenotypic plasticity produces individual differences within social groups (“swarms”). The distribution of differences may be adaptive for their local ecology. The individual-level “interaction rules” and resultant plasticity used in their self-organization may be instructive for swarm robot controllers. **(D)** The clonal raider ant *Ooceraea biroi* (photo: courtesy of Daniel Kronauer). **(E)** the Amazon molly *Poecilia Formosa* (photo: courtesy of David Bierbach).

Emerging Technologies Favoring (Partially) Irreversible Plasticity

In the context of model-based reflex behaviors, if internal reaction thresholds are computer variables there is no design requirement to make their setting irreversible; though this may be suitable for time and geography-limited missions, where robots can be retrieved and reset for redeployment. Several emerging technologies favor irreversible plasticity, however. For example, the field of “soft” robotics employs soft structures to flexibly interact with unpredictable environments (Kim et al., 2013). Robot intelligence can be “outsourced” from the computer “brain” to the robot “body” (morphology) and its nonlinear responses, exploiting “embodied intelligence” (Bongard, 2011). This outsourcing can go a step further in collectives, as phenotypic diversity in soft swarms could result merely from past sensitivity (hysteresis) to exposure temperature, strain and other conditions. Moreover, soft robots raise the possibility of biodegradability (Rossiter et al., 2016), further relaxing constraints on ubiquitous deployment. Another exciting development is the possibility of “autonomous” or “robotic” materials (McEvoy and Correll, 2015), smart composites that can autonomously change shape, stiffness, appearance and other properties. In electronics, the idea of a “memristor”—a resistor with “memory” of the charge flowing through it—raises the possibility of “neuromorphic computing” that parallels in some way the synaptic plasticity of a brain

(Zidan et al., 2018; Wang et al., 2019). At smaller length scales, exciting possibilities exist for micro-scale swarms (e.g., Martel et al., 2009; Yigit et al., 2019). As robot swarms aim toward large numbers, and possibly smaller scales, the heterogeneity and stochasticity associated with minimal robots may become inevitable. Rather than seeing this as an engineering nuisance, swarm designers can embrace its possibilities (White et al., 2004; Ramachandran et al., 2018; Scholz et al., 2018; Li et al., 2019), and (partially) irreversible plasticity could contribute toward adaptation to field conditions.

SWARM-LEVEL STRENGTH IN INDIVIDUAL-LEVEL DIVERSITY

Phenotypic plasticity can produce helpful individual-level adaptations: for example, a suitable threshold to switch behaviors. Even more significantly in a swarm context, though, is the possibility of producing emergent functionality for the group. Even in what appear to be superficially similar units in cooperative biological groups there can be a surprising level of diversity (Blodgett et al., 2016); this heterogeneity is increasingly recognized as an adaptive group trait (Clobert et al., 2009; Kennedy et al., 2017). Thus, while plasticity in a certain trait may actually make a small or negligible contribution to the direct fitness of the individual, it may be nevertheless an important indirect contribution to the fitness of the swarm.

Diversity as a Shield Against Adversity

Robustness is frequently claimed for swarm robot systems, but if a homogeneous controller results in homogeneous behavior the swarm may be liable to systematic failure if it encounters unexpected environmental conditions or faulty or malicious agents (Higgins et al., 2009). This might be compared to inbreeding in biology, which is a cause of disease vulnerability. Conversely, diversity can help resistance (Ugelvig et al., 2010).

Fault tolerance in swarms is an important precondition for scalability (Winfield and Nembrini, 2006; Bjercknes and Winfield, 2013) and phenotypic plasticity may paradoxically help the swarm to cope with the unexpected. This is because it can result in a range of subtle—or substantial—individual differences, which will need to be made compatible with agent—agent interaction as a matter of course.

Diversity for Homeostasis

In biological systems phenotypic diversity can also promote positive collective success: for example in honeybees diversity in reaction thresholds for their cooling behavior promotes stability in nest thermoregulation (Jones et al., 2004). Although this example is driven by genetic heterogeneity, it could equally be designed in a robot context as a result of phenotypic plasticity.

Diversity for Decision-Making

If a swarm is to be autonomous it also needs to be capable of making collective decisions. Again, diversity of reaction thresholds or option assessment behavior, as seen in ants, may help this process (Masuda et al., 2015; O'Shea-Wheller et al., 2017). Such studies highlight the importance of heterogeneity among individuals, rather than precise calibration, for effective collective decision-making.

Diversity for Foraging and Search

Finally, variation in individual behavior can be important for foraging and search in systems as diverse as ants and immune systems (Beverly et al., 2009; Fricke et al., 2016).

BEHAVIORAL PLASTICITY

Behavioral plasticity allows organisms to make relatively rapid adjustments in their function to adapt to changing environmental conditions. Learning, which shapes behavior, can be seen as a form of plasticity (Agrawal, 2001) and allows “culture”—inter-generational transmission of behaviors through social learning (Whiten et al., 2017). In robot swarms this has been demonstrated in robot societies through imitation learning (Winfield and Erbas, 2011), and can arise simply from robot and sensor noise (Erbaş et al., 2013). Perhaps the most obvious opportunity for ready transposition into robot swarms, though, is seen in animal “personality” differences.

Animal and Robot “Personalities”

Modeling work in biological collective behavior often assumes agents are homogeneous in their characteristics, but there is

increasing recognition that consistent individual differences in behavior (“personality”) among group members can be important for group function in local ecologies (Dall et al., 2012). Examples of significant personality axes include: risk-taking behavior (boldness—shyness), exploratory behavior (neophilic—neophobic), activity levels (active—inactive), sociability (social—asocial), and aggression (aggressive—non-aggressive) (Réale et al., 2007). This can be observed at the level of the individual or the whole group, giving rise to the notion of collective personalities (Jandt et al., 2014). While early development is important to the formation of personality, it can be somewhat plastic over an individual's lifetime (Groothuis and Trillmich, 2011). As a result, group-level plasticity in personality is also observed (Norman et al., 2017). In *Stegodyphus* social spiders (Figure 2A), there is a link between social interactions and boldness change (Hunt et al., 2018); the group-level distribution of boldness is important for their collective performance (Hunt et al., 2019b).

In relation to swarm robotics, the notion of personality maps readily to adaptive threshold-based behaviors, for example the likelihood of switching behaviors in probabilistic finite state machines (Liu and Winfield, 2010; Castello et al., 2016). It can also map to very simple adaptations such as variable waiting times in response to changing swarm density (Wahby et al., 2019), which one might term “sociability,” for example. Simpler still, the decision to be active or inactive, which may make little sense at the level of the individual robot with a mission to complete, can be adaptive to a swarm that might need to keep some units in reserve; the identification of “lazy ants” (Charbonneau and Dornhaus, 2015) suggests plasticity in activity may be valuable. Thus, the growing literature on animal personality research—particularly on its ontogeny in social groups—may indicate simple behavioral mechanisms (“interaction rules”) that can be adapted in the context of self-organizing robots.

The Relevance of Highly Related and Clonal Animals

In social insects, caste determination (e.g., worker or queen) is driven by a varying combination of “nature” (genotype) and “nurture” (environment) (Schwander et al., 2010). To try and understand how the environment (particularly the social environment) shapes such phenotypic plasticity, biologists study highly related or even clonal organisms, which controls for the effect of genetics. Social spiders (Figure 2A) are highly inbred; and two emerging model organisms are the clonal raider ant *Ooceraea biroi* (e.g., Ulrich et al., 2018) and the Amazon molly *Poecilia Formosa*, a small freshwater fish (e.g., Bierbach et al., 2017) (Figures 2D,E). As well as being prime candidates to answer fundamental questions in ecology and evolution (Laskowski et al., 2019), such organisms could provide important bio-inspiration to the development of homogeneous swarm controllers that can result in heterogeneity that is adaptive at the swarm-level.

PHYSIOLOGICAL AND MORPHOLOGICAL PLASTICITY

An example of physiological plasticity in nature is the invasive cane toad *Rhinella marina* (Figure 2B). It succeeds as an invader into unfamiliar environments, at least in part, because it can adjust its core body temperature to new climates (McCann et al., 2018). It is also somewhat plastic in its social behavior (Gruber et al., 2017): an example of successfully combining multiple modes of plasticity. Physiological plasticity in a robotics context could mean something as simple as the availability of different power consumption modes: for example, a high energy mode for exploration and data transmission, and a standby mode for *in situ* monitoring of an environment. This could be critical to long-term swarm resilience.

Examples of morphological plasticity in nature include the water flea *Daphnia lumholtzi* (Green, 1967), which can respond drastically to the presence of predators by developing a sharp helmet and extended tail spine (Agrawal, 2001); or in bacteria that undergo filamentation (elongation) in response to stress (Figure 2C; Justice et al., 2008). At the group level, a form of collective mechanical adaptation is observed in honeybee swarms (Peleg et al., 2018). In swarm robotics research so far, a form of morphological plasticity is possible through self-assembly into connected groups of various forms (Brambilla et al., 2013). Examples of this include the “s-bot” which can physically attach to each other (Mondada et al., 2004), conceptual demonstrations in “Kilobots” (Rubenstein et al., 2014; Slavkov et al., 2018; Carrillo-Zapata et al., 2019), or the idea of a “mergeable nervous system” (Mathews et al., 2017). More broadly, one can design robots to adapt their own morphology (Divband Soorati et al., 2019; Hauser, 2019; Kriegman et al., 2019); in combination such “multi-robot organisms” (Levi and Kernbach, 2010) may self-organize a wide range of adaptations.

DISCUSSION

Swarm robotics relies on the power of emergence to produce engineered systems that are capable of “more than the sum of their parts”. This is possible even with very simple agents. As we take robot swarms into the field, the temptation may be to move away from the principle of individual-level simplicity in hardware and controllers. Instead, a different way forward may be to re-focus on the ingenuity of nature in building resilient social systems. Increasingly, phenotypic plasticity is recognized

as center-stage in producing adaptive biological variation, and would seem to be similarly indispensable in embodied collective artificial intelligences. We can, and should, attempt intensive off-line optimization of swarm controllers (Birattari et al., 2019), but this could be combined with possibilities to manifest plasticity in behavior, “physiology” and morphology in heterogeneous simulated environments. Their respective impact on swarm-level functions might be analyzed with respect to information flow (Pitonakova et al., 2016). In a “bottom-up” approach to swarm design (Crespi et al., 2008) a moderate amount of plasticity across these modes could be added with very limited cost, but potentially far-reaching implications for swarm resilience, contributing toward the practical realization of “dependable swarms” (Winfield et al., 2004).

For biologists, robots can be used as tools for understanding biological evolution (Doncieux et al., 2015). The systematic addition of various forms of “phenotypic plasticity” to robots could also contribute toward this aim. Meanwhile, for engineers, with plasticity and mass-producible minimal robots, the approach of sending large numbers of cheap and expendable units on missions (“fast, cheap and out of control”; Brooks and Flynn, 1989) might have a better chance of success. A review across plasticity modes and relevant organisms (e.g., for air, water or land) could become a routine part of a swarm design process. The symbiosis between biology and engineering seen in the field of swarm robotics can go from strength to strength.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article, further enquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

EH conceived of this perspective and wrote the paper.

FUNDING

EH acknowledges support from the Royal Academy of Engineering and the Office of the Chief Science Adviser for National Security under the UK Intelligence Community Postdoctoral Fellowship Programme; and the UK Engineering and Physical Sciences Research Council (EPSRC DTP Doctoral Prize grant no. EP/N509619/1).

REFERENCES

- Agrawal, A. A. (2001). Phenotypic plasticity in the interactions and evolution of species. *Science* 294, 321–326. doi: 10.1126/science.1060701
- Arkin, R. C. (1998). *Behavior-Based Robotics*. Cambridge, MA: MIT Press.
- Beverly, B. D., McLendon, H., Nacu, S., Holmes, S., and Gordon, D. M. (2009). How site fidelity leads to individual differences in the foraging activity of harvester ants. *Behav. Ecol.* 20, 633–638. doi: 10.1093/beheco/arp041
- Bierbach, D., Laskowski, K. L., and Wolf, M. (2017). Behavioural individuality in clonal fish arises despite near-identical rearing conditions. *Nat. Commun.* 8:15361. doi: 10.1038/ncomms15361
- Birattari, M., Ligot, A., Bozhinoski, D., Brambilla, M., Francesca, G., Garattoni, L., et al. (2019). Automatic off-line design of robot swarms: a manifesto. *Front. Robot. AI* 1–6. doi: 10.3389/frobt.2019.00059
- Bjerknes, J. D., and Winfield, A. F. T. (2013). “On Fault Tolerance and Scalability of Swarm Robotic Systems,” in *Distributed Autonomous Robotic Systems: The 10th International Symposium*, eds. A. Martinoli, F. Mondada,

- N. Correll, G. Mermoud, M. Egerstedt, M. A. Hsieh, et al. (Berlin; Heidelberg: Springer), 431–444.
- Bjerknes, J. D., Winfield, A. F. T., and Melhuish, C. (2007). “An analysis of emergent taxis in a wireless connected swarm of mobile robots,” in *2007 IEEE Swarm Intelligence Symposium* (Honolulu, HI: IEEE), 45–52. doi: 10.1109/SIS.2007.368025
- Blodgett, D. M., Redick, S. D., and Harlan, D. M. (2016). Surprising heterogeneity of pancreatic islet cell subsets. *Cell. Syst.* 3, 330–332. doi: 10.1016/j.cels.2016.10.009
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY: Oxford University Press.
- Bongard, J. (2011). The “What,” “How” and the “Why” of Evolutionary Robotics,” in *New Horizons in Evolutionary Robotics*, eds. S. Doncieux, N. Bredeche, and J.-B. Mouret (Berlin; Heidelberg: Springer), 29–35.
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* 7, 1–41. doi: 10.1007/s11721-012-0075-2
- Brooks, R. A. (1992). “Artificial life and real robots,” in *Proceedings of the First European Conference on Artificial Life* (Cambridge, MA: MIT Press), 3–10.
- Brooks, R. A., and Flynn, A. M. (1989). “Fast, cheap and out of control: a robot invasion of the solar system,” in *Journal of the British Interplanetary Society*, Vol. 42, 478–485. Available online at: <http://people.csail.mit.edu/brooks/papers/fast-cheap.pdf> (accessed December 08, 2004).
- Brutschy, A., Tran, N. L., Baiboun, N., Frison, M., Pini, G., Roli, A., et al. (2012). Costs and benefits of behavioral specialization. *Rob. Auton. Syst.* 60, 1408–1420. doi: 10.1016/j.robot.2012.05.018
- Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001). *Self-Organization in Biological Systems*. Princeton, NJ: Princeton University Press.
- Carrillo-Zapata, D., Sharpe, J., Winfield, A. F. T., Giuggioli, L., and Hauert, S. (2019). Toward controllable morphogenesis in large robot swarms. *IEEE Robot. Autom. Lett.* 4, 3386–3393. doi: 10.1109/LRA.2019.2926961
- Castello, E., Yamamoto, T., Libera, F. D., Liu, W., Winfield, A. F. T., Nakamura, Y., et al. (2016). Adaptive foraging for simulated and real robotic swarms: the dynamical response threshold approach. *Swarm Intell.* 10, 1–31. doi: 10.1007/s11721-015-0117-7
- Charbonneau, D., and Dornhaus, A. (2015). Workers ‘specialized’ on inactivity: behavioral consistency of inactive workers and their role in task allocation. *Behav. Ecol. Sociobiol.* 69, 1459–1472. doi: 10.1007/s00265-015-1958-1
- Clobert, J., Le Galliard, J.-F., Cote, J., Meylan, S., and Massot, M. (2009). Informed dispersal, heterogeneity in animal dispersal syndromes and the dynamics of spatially structured populations. *Ecol. Lett.* 12, 197–209. doi: 10.1111/j.1461-0248.2008.01267.x
- Crespi, V., Galstyan, A., and Lerman, K. (2008). Top-down vs bottom-up methodologies in multi-agent system design. *Auton. Robots* 24, 303–313. doi: 10.1007/s10514-007-9080-5
- Dall, S. R. X., Bell, A. M., Bolnick, D. I., and Ratnieks, F. L. W. (2012). An evolutionary ecology of individual differences. *Ecol. Lett.* 15, 1189–1198. doi: 10.1111/j.1461-0248.2012.01846.x
- Dingemanse, N. J., Kazem, A. J. N., Réale, D., and Wright, J. (2010). Behavioral reaction norms: animal personality meets individual plasticity. *Trends Ecol. Evol.* 25, 81–89. doi: 10.1016/j.tree.2009.07.013
- Divband Soorati, M., Heinrich, M. K., Ghofrani, J., Zahadat, P., and Hamann, H. (2019). Photomorphogenesis for robot self-assembly: adaptivity, collective decision-making, and self-repair. *Bioinspir. Biomim.* 14:05600. doi: 10.1088/1748-3190/ab2958
- Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. E. (2015). Evolutionary robotics: what, why, 433 and where to. *Front. Robot. AI* 2:4. doi: 10.3389/frobt.2015.00004
- Dorigo, M., Trianni, V., Sahin, E., Groß, R., and Labella, T. H., Baldassarre, G., et al. (2004). Evolving self-organizing behaviors for a swarm-bot. *Auton. Robots* 17, 223–245. doi: 10.1023/B:AURO.0000033973.24945.f3
- Duarte, M., Costa, V., Gomes, J., Rodrigues, T., Silva, F., Oliveira, S. M., et al. (2016). Evolution of collective behaviors for a real Swarm of aquatic surface robots. *PLoS ONE* 11:e0151834. doi: 10.1371/journal.pone.0151834
- Erbas, M. D., Winfield, A. F. T., and Bull, L. (2013). Embodied imitation-enhanced reinforcement learning in multi-agent systems. *Adapt. Behav.* 22, 31–50. doi: 10.1177/1059712313500503
- Ferrante, E., Turgut, A. E., Duéñez-Guzmán, E., Dorigo, M., and Wenseleers, T. (2015). Evolution of Self-Organized Task Specialization in Robot Swarms. *PLOS Comput. Biol.* 11:e1004273. doi: 10.1371/journal.pcbi.1004273
- Floreano, D., and Mondada, F. (1996). “Evolution of plastic neurocontrollers for situated agents. in from animals to animats,” in *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB)* (Cambridge, MA: MIT Press), 402–410.
- Frei, R., and Giovanna, M. S. (2012). The future of complexity engineering. *Open Eng.* 2, 164–188. doi: 10.2478/s13531-011-0071-0
- Fricke, G. M., Hecker, J. P., Cannon, J. L., and Moses, M. E. (2016). Immune-inspired search strategies for robot swarms. *Robotica* 34, 1791–1810. doi: 10.1017/S0263574716000382
- Garattoni, L., and Birattari, M. (2018). Autonomous task sequencing in a robot swarm. *Sci. Robot.* 3, 1–12. doi: 10.1126/scirobotics.aat0430
- Gordon, D. M., Guetz, A., Greene, M. J., and Holmes, S. (2011). Colony variation in the collective regulation of foraging by harvester ants. *Behav. Ecol.* 22, 429–435. doi: 10.1093/beheco/arq218
- Goulson, D., Nicholls, E., Botías, C., and Rotheray, E. L. (2015). Bee declines driven by combined stress from parasites, pesticides, and lack of flowers. *Science* 347:1255957. doi: 10.1126/science.1255957
- Green, J. (1967). The distribution and variation of *Daphnia lumholtzi* (Crustacea: Cladocera) in relation to fish predation in Lake Albert, East Africa. *J. Zool.* 151, 181–197. doi: 10.1111/j.1469-7998.1967.tb02109.x
- Groothuis, T. G. G., and Trillmich, F. (2011). Unfolding personalities: the importance of studying ontogeny. *Dev. Psychobiol.* 53, 641–655. doi: 10.1002/dev.20574
- Gruber, J., Whiting, M. J., Brown, G., and Shine, R. (2017). The loneliness of the long-distance toad: invasion history and social attraction in cane toads (*Rhinella marina*). *Biol. Lett.* 13:20170445. doi: 10.1098/rsbl.2017.0445
- Haasdijk, E., Bredeche, N., and Eiben, A. E. (2014). Combining environment-driven adaptation and task-driven optimisation in evolutionary robotics. *PLoS ONE* 9:e98466. doi: 10.1371/journal.pone.0098466
- Hamann, H. (2018). *Swarm Robotics: A Formal Approach*. Cham: Springer International Publishing. doi: 10.1007/978-3-319-74528-2
- Hauser, H. (2019). Resilient machines through adaptive morphology. *Nat. Mach. Intell.* 1, 338–339. doi: 10.1038/s42256-019-0076-6
- Hecker, J. P., and Moses, M. E. (2015). Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. *Swarm Intell.* 9, 43–70. doi: 10.1007/s11721-015-0104-z
- Higgins, F., Tomlinson, A., and Martin, K. M. (2009). “Survey on security challenges for swarm robotics,” in *2009 Fifth International Conference on Autonomic and Autonomous Systems* (Valencia: IEEE), 307–312. doi: 10.1109/ICAS.2009.62
- Holway, D. A., Lach, L., Suarez, A. V., Tsutsui, N. D., and Case, T. J. (2002). The causes and consequences of ant invasions. *Annu. Rev. Ecol. Syst.* 33, 181–233. doi: 10.1146/annurev.ecolsys.33.010802.150444
- Hunt, E. R., Jones, S., and Hauert, S. (2019a). Testing the limits of pheromone stigmergy in high density robot swarms. *R. Soc. Open Sci.* 6:190225. doi: 10.1098/rsos.190225
- Hunt, E. R., Mi, B., Fernandez, C., Wong, B. M., Pruitt, J. N., and Pinter-Wollman, N. (2018). Social interactions shape individual and collective personality in social spiders. *Proc. R. Soc. B Biol. Sci.* 285:20181366. doi: 10.1098/rspb.2018.1366
- Hunt, E. R., Mi, B., Geremew, R., Fernandez, C., Wong, B. M., Pruitt, J. N., et al. (2019b). Resting networks and personality predict attack speed in social spiders. *Behav. Ecol. Sociobiol.* 73:97. doi: 10.1007/s00265-019-2715-7
- Hüttenrauch, M., Šošić, A., and Neumann, G. (2018). “Local communication protocols for learning complex swarm behaviors with deep reinforcement learning,” in *Swarm Intelligence. ANTS 2018. Lecture Notes in Computer Science*, vol 11172, eds. M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. Reina, and V. Trianni (Cham: Springer), 71–83.
- Jakobi, N., Husbands, P., and Harvey, I. (1995). “Noise and the reality gap: the use of simulation in evolutionary robotics,” in *European Conference on Artificial Life*, eds. F. Morán, A. Moreno, J. J. Merelo, and P. Chacón (Berlin, Heidelberg: Springer), 704–720.

- Jandt, J. M., Bengston, S., Pinter-Wollman, N., Pruitt, J. N., Raine, N. E., Dornhaus, A., et al. (2014). Behavioral syndromes and social insects: personality at multiple levels. *Biol. Rev.* 89, 48–67. doi: 10.1111/bvr.12042
- Jones, J. C., Myerscough, M. R., Graham, S., and Oldroyd, B. P. (2004). Honey bee nest thermoregulation: diversity promotes stability. *Science* 305, 402–404. doi: 10.1126/science.1096340
- Jones, S., Winfield, A. F., Hauert, S., and Studley, M. (2019). Onboard evolution of understandable swarm behaviors. *Adv. Intell. Syst.* 1:1900031. doi: 10.1002/aisy.201900031
- Just, W. A., and Moses, M. E. (2018). “Flexibility through autonomous decision-making in robot swarms,” in *2017 IEEE IEEE Symposium Series on Computational Intelligence SSCI 2017 – Proceedings* (Honolulu, HI: IEEE), 1–8. doi: 10.1109/SSCI.2017.8285248
- Justice, S. S., Hunstad, D. A., Cegelski, L., and Hultgren, S. J. (2008). Morphological plasticity as a bacterial survival strategy. *Nat. Rev. Microbiol.* 6, 162–168. doi: 10.1038/nrmicro1820
- Kelly, S. A., Panhuis, T. M., and Stoehr, A. M. (2012). Phenotypic plasticity: molecular mechanisms and adaptive significance. *Compr. Physiol.* 2, 1417–1439. doi: 10.1002/cphy.c110008
- Kennedy, P., Baron, G., Qiu, B., Freitak, D., Helanterä, H., Hunt, E. R., et al. (2017). Deconstructing superorganisms and societies to address big questions in biology. *Trends Ecol. Evol.* 32, 861–872. doi: 10.1016/j.tree.2017.08.004
- Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends Biotechnol.* 31, 287–294. doi: 10.1016/j.tibtech.2013.03.002
- Kriegman, S., Walker, S., Shah, D. S., Kramer-Bottiglio, R., and Bongard, J. (2019). “Automated Shapeshifting for Function Recovery in Damaged Robots,” in *Robotics: Science and Systems XV* (Messe Freiburg). Available online at: <https://arxiv.org/pdf/1905.09264.pdf> (accessed May 22, 2019).
- Laskowski, K. L., Doran, C., Bierbach, D., Krause, J., and Wolf, M. (2019). Naturally clonal vertebrates are an untapped resource in ecology and evolution research. *Nat. Ecol. Evol.* 3, 161–169. doi: 10.1038/s41559-018-0775-0
- Levi, P., and Kernbach, S. (2010). *Symbiotic Multi-Robot Organisms: Reliability, Adaptability, Evolution*. Berlin; Heidelberg: Springer-Verlag. doi: 10.1007/978-3-642-11692-6
- Li, S., Batra, R., Brown, D., Chang, H.-D., Ranganathan, N., Hoberman, C., et al. (2019). Particle robotics based on statistical mechanics of loosely coupled components. *Nature* 567, 361–365. doi: 10.1038/s41586-019-1022-9
- Liu, W., and Winfield, A. F. T. (2010). Modeling and optimization of adaptive foraging in swarm robotic systems. *Int. J. Rob. Res.* 29, 1743–1760. doi: 10.1177/0278364910375139
- Martel, S., Mohammadi, M., Felfoul, O., Lu, Z., and Poupponneau, P. (2009). Flagellated magnetotactic bacteria as controlled MRI-trackable propulsion and steering systems for medical nanorobots operating in the human microvasculature. *Int. J. Rob. Res.* 28, 571–582. doi: 10.1177/0278364908100924
- Masuda, N., O’Shea-Wheller, T. A., Doran, C., and Franks, N. R. (2015). Computational model of collective nest selection by ants with heterogeneous acceptance thresholds. *R. Soc. Open Sci.* 2:140533. doi: 10.1098/rsos.140533
- Matarić, M. J. (1993). “Designing emergent behaviors: From local interactions to collective intelligence,” in *Proceedings of the Second International Conference on Simulation of Adaptive Behavior* (Honolulu, HI), 432–441.
- Matarić, M. J. (1995). Designing and understanding adaptive group behavior. *Adapt. Behav.* 4, 51–80. doi: 10.1177/105971239500400104
- Mathews, N., Christensen, A. L., O’Grady, R., Mondada, F., and Dorigo, M. (2017). Mergeable nervous systems for robots. *Nat. Commun.* 8, 1–7. doi: 10.1038/s41467-017-00109-2
- McCann, S. M., Kosmala, G. K., Greenlees, M. J., and Shine, R. (2018). Physiological plasticity in a successful invader: rapid acclimation to cold occurs only in cool-climate populations of cane toads (*Rhinella marina*). *Conserv. Physiol.* 6:cox072. doi: 10.1093/conphys/cox072
- McEvoy, M. A., and Correll, N. (2015). Materials that couple sensing, actuation, computation, and communication. *Science* 347:1261689. doi: 10.1126/science.1261689
- McGuire, K. N., De Wagter, C., Tuyls, K., Kappen, H. J., and de Croon, G. C. H. E. (2019). Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment. *Sci. Robot.* 4:eaaw9710. doi: 10.1126/scirobotics.aaw9710
- Mitrano, P., Burklund, J., Giancola, M., and Pinciroli, C. (2019). “A minimalistic approach to segregation in robot swarms,” in *2019 International Symposium on Multi-Robot and Multi-Agent Systems* (New Brunswick, NJ: IEEE), 105–111. doi: 10.1109/MRS.2019.8901068
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotcz, A., et al. (2009). “The e-puck, a robot designed for education in engineering,” in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competition*. 1, 59–65. Available online at: <https://infoscience.epfl.ch/record/135236> (accessed January 27, 2018).
- Mondada, F., Pettinaro, G. C., Guignard, A., Kwee, I. W., Floreano, D., Deneubourg, J. L., et al. (2004). Swarm-bot: a new distributed robotic concept. *Auton. Robots* 17, 193–221. doi: 10.1023/B:AURO.0000033972.50769.1c
- Moore, A. J., Brodie, E. D. III, and Wolf, J. B. (1997). Interacting phenotypes and the evolutionary process: I. direct and indirect genetic effects of social interactions. *Evolution* 51, 1352–1362. doi: 10.1111/j.1558-5646.1997.tb01458.x
- Nitschke, G. S., Schut, M. C., and Eiben, A. E. (2012). Evolving behavioral specialization in robot teams to solve a collective construction task. *Swarm Evol. Comput.* 2, 25–38. doi: 10.1016/j.swevo.2011.08.002
- Nolfi, S., and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, MA: MIT press.
- Nolfi, S., Miglino, O., and Parisi, D. (1994). “Phenotypic plasticity in evolving neural networks,” in *Proceedings - From Perception to Action Conference 1994* (Lausanne: IEEE), 146–157. doi: 10.1109/FPA.1994.636092
- Norman, V. C., Pamminger, T., and Hughes, W. O. H. (2017). The effects of disturbance threat on leaf-cutting ant colonies: a laboratory study. *Insectes Soc.* 64, 75–85. doi: 10.1007/s00040-016-0513-z
- Nouyan, S., Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2009). Teamwork in self-organized robot colonies. *IEEE Trans. Evol. Comput.* 13, 695–711. doi: 10.1109/TEVC.2008.2011746
- O’Shea-Wheller, T. A., Masuda, N., Sendova-Franks, A. B., and Franks, N. R. (2017). Variability in individual assessment behaviour and its implications for collective decision-making. *Proc. R. Soc. B Biol. Sci.* 284:20162237. doi: 10.1098/rspb.2016.2237
- Peleg, O., Peters, J. M., Salcedo, M. K., and Mahadevan, L. (2018). Collective mechanical adaptation of honeybee swarms. *Nat. Phys.* 14, 1193–1198. doi: 10.1038/s41567-018-0262-1
- Pitonakova, L., Crowder, R., and Bullock, S. (2016). Information flow principles for plasticity in foraging robot swarms. *Swarm Intell.* 10, 33–63. doi: 10.1007/s11721-016-0118-1
- Ramachandran, R. K., Elamvazhuthi, K., and Berman, S. (2018). “An optimal control approach to mapping GPS-denied environments using a stochastic robotic swarm,” in *BT - Robotics Research, vol. 1*, eds A. Bicchi and W. Burgard (Cham: Springer International Publishing), 477–493.
- Réale, D., Reader, S. M., Sol, D., McDougall, P. T., and Dingemanse, N. J. (2007). Integrating animal temperament within ecology and evolution. *Biol. Rev.* 82, 291–318. doi: 10.1111/j.1469-185X.2007.00010.x
- Rossiter, J., Winfield, J., and Ieropoulos, I. (2016). Here today, gone tomorrow: biodegradable soft robots. *Electroact. Polym. Actuators Devices* 2016:97981S. doi: 10.1117/12.2220611
- Rubenstein, M., Ahler, C., and Nagpal, R. (2012). “Kilobot: a low cost scalable robot system for collective behaviors,” in *2012 IEEE International Conference on Robotics and Automation (ICRA)* (Saint Paul, MN: IEEE), 3293–3298. doi: 10.1109/ICRA.2012.6224638
- Rubenstein, M., Cornejo, A., and Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science* 345, 795–799. doi: 10.1126/science.1254295
- Russell, S. J., and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice Hall.
- Şahin, E. (2005). “Swarm robotics: from sources of inspiration to domains of application,” in *Swarm Robotics - SAB 2004 International Workshop. Lecture Notes in Computer Science*, eds E. Sahin and W. M. Spears (Berlin; Heidelberg: Springer), 10–20. doi: 10.1007/978-3-540-30552-1_2
- Schlichting, C. D. (1986). The evolution of phenotypic plasticity in plants. *Annu. Rev. Ecol. Syst.* 17, 667–693. doi: 10.1146/annurev.es.17.110186.003315
- Schlichting, C. D., and Pigliucci, M. (1998). *Phenotypic Evolution: A Reaction Norm Perspective*. Sunderland, MA: Sinauer Associates Incorporated.
- Schmickl, T., Thenius, R., Moslinger, C., Timmis, J., Tyrrell, A., Read, M., et al. (2011). “CoCoRo—the self-aware underwater swarm,” in *2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops* (Ann Arbor, MI: IEEE), 120–126. doi: 10.1109/SASOW.2011.11
- Scholz, C., Engel, M., and Pöschel, T. (2018). Rotating robots move collectively and self-organize. *Nat. Commun.* 9, 1–8. doi: 10.1038/s41467-018-03154-7

- Schwander, T., Lo, N., Beekman, M., Oldroyd, B. P., and Keller, L. (2010). Nature versus nurture in social insect caste differentiation. *Trends Ecol. Evol.* 25, 275–282. doi: 10.1016/j.tree.2009.12.001
- Slavkov, I., Carrillo-Zapata, D., Carranza, N., Diego, X., Jansson, F., Kaandorp, J., et al. (2018). Morphogenesis in robot swarms. *Sci. Robot.* 3:eaau9178. doi: 10.1126/scirobotics.aau9178
- Stamps, J. A. (2016). Individual differences in behavioral plasticities. *Biol. Rev.* 91, 534–567. doi: 10.1111/brev.12186
- Trianni, V. (2008). *Evolutionary Swarm Robotics: Evolving Self-Organising Behaviours in Groups of Autonomous Robots*. Berlin; Heidelberg: Springer. doi: 10.1007/978-3-540-77612-3
- Trueba, P., Prieto, A., Caamaño, P., Bellas, F., and Duro, R. J. (2011). Task-driven species in evolutionary robotic Teams. in *Foundations on Natural and Artificial Computation*, eds J. M. Ferrández, J. R. Álvarez Sánchez, F. de la Paz, and F. J. Toledo (Berlin, Heidelberg: Springer), 138–147. doi: 10.1007/978-3-642-21344-1_15
- Ugelvig, L. V., Kronauer, D. J. C., Schrempf, A., Heinze, J., and Cremer, S. (2010). Rapid anti-pathogen response in ant societies relies on high genetic diversity. *Proc. R. Soc. B Biol. Sci.* 277, 2821–2828. doi: 10.1098/rspb.2010.0644
- Ulrich, Y., Saragosti, J., Tokita, C. K., Tarnita, C. E., and Kronauer, D. J. C. (2018). Fitness benefits and emergent division of labor at the onset of group living. *Nature* 560, 635–638. doi: 10.1038/s41586-018-0422-6
- Wahby, M., Petzold, J., Eschke, C., Schmickl, T., and Hamann, H. (2019). “Collective change detection: adaptivity to dynamic swarm densities and light conditions in robot swarms,” in *2019 Conference on Artificial Life* (Newcastle), 642–649. doi: 10.1162/isal_a_00233
- Walter, W. G. (1950). An Imitation of Life. *Sci. Am.* 182, 42–45. doi: 10.1038/scientificamerican0550-42
- Wang, F. Z., Li, L., Shi, L., Wu, H., and Chua, L. O. (2019). Φ memristor: real memristor found. *J. Appl. Phys.* 125:054504. doi: 10.1063/1.5042281
- Weaver, I. C. G., Cervoni, N., Champagne, F. A., D'Alessio, A. C., Sharma, S., Seckl, J. R., et al. (2004). Epigenetic programming by maternal behavior. *Nat. Neurosci.* 7, 847–854. doi: 10.1038/nn1276
- White, P. J., Kopanski, K., and Lipson, H. (2004). “Stochastic self-reconfigurable cellular robotics,” in *Proceedings of the IEEE International Conference on Robotics and Automation* (New Orleans, LA: IEEE), 2888–2893. doi: 10.1109/ROBOT.2004.1307499
- Whiten, A., Ayala, F. J., Feldman, M. W., and Laland, K. N. (2017). The extension of biology through culture. *Proc. Natl. Acad. Sci. U.S.A.* 114, 7775–7781. doi: 10.1073/pnas.1707630114
- Winfield, A. F. T., and Erbas, M. D. (2011). On embodied memetic evolution and the emergence of behavioral traditions in Robots. *Memetic. Comput.* 3, 261–270. doi: 10.1007/s12293-011-0063-x
- Winfield, A. F. T., Harper, C. J., and Nembrini, J. (2004). “Toward dependable swarms and a new discipline of swarm engineering,” in *International Workshop on Swarm Robotics* (Berlin: Springer), 126–142.
- Winfield, A. F. T., and Nembrini, J. (2006). Safety in numbers: fault tolerance in robot swarms. *Int. J. Model. Identif. Control* 1, 30–37. doi: 10.1504/IJMIC.2006.008645
- Wolf, J. B., Brodie, E. D. III, and Moore, A. J. (1999). Interacting phenotypes and the evolutionary process. II. Selection resulting from social interactions. *Am. Nat.* 153, 254–266. doi: 10.1086/303168
- Yigit, B., Alapan, Y., and Sitti, M. (2019). Programmable collective behavior in dynamically self-assembled mobile microrobotic swarms. *Adv. Sci.* 6:1801837. doi: 10.1002/advs.201801837
- Zidan, M. A., Strachan, J. P., and Lu, W. D. (2018). The future of electronics based on memristive systems. *Nat. Electron.* 1, 22–29. doi: 10.1038/s41928-017-0006-8

Conflict of Interest: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Hunt. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Swarm Robotic Behaviors and Current Applications

Melanie Schranz^{1*}, Martina Umlauf¹, Micha Sende¹ and Wilfried Elmenreich^{2*}

¹ Lakeside Labs GmbH, Klagenfurt, Austria, ² Institute of Networked and Embedded Systems, University of Klagenfurt, Klagenfurt, Austria

OPEN ACCESS

Edited by:

Anders Lyhne Christensen,
University of Southern Denmark,
Denmark

Reviewed by:

Sabine Hauert,
University of Bristol, United Kingdom
Alan Gregory Millard,
University of Lincoln, United Kingdom

*Correspondence:

Melanie Schranz
schranz@lakeside-labs.com
Wilfried Elmenreich
wilfried.elmenreich@aau.at

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 15 November 2019

Accepted: 03 March 2020

Published: 02 April 2020

Citation:

Schranz M, Umlauf M, Sende M and
Elmenreich W (2020) Swarm Robotic
Behaviors and Current Applications.
Front. Robot. AI 7:36.
doi: 10.3389/frobt.2020.00036

In swarm robotics multiple robots collectively solve problems by forming advantageous structures and behaviors similar to the ones observed in natural systems, such as swarms of bees, birds, or fish. However, the step to industrial applications has not yet been made successfully. Literature is light on real-world swarm applications that apply actual swarm algorithms. Typically, only parts of swarm algorithms are used which we refer to as basic swarm behaviors. In this paper we collect and categorize these behaviors into spatial organization, navigation, decision making, and miscellaneous. This taxonomy is then applied to categorize a number of existing swarm robotic applications from research and industrial domains. Along with the classification, we give a comprehensive overview of research platforms that can be used for testing and evaluating swarm behavior, systems that are already on the market, and projects that target a specific market. Results from this survey show that swarm robotic applications are still rare today. Many industrial projects still rely on centralized control, and even though a solution with multiple robots is employed, the principal idea of swarm robotics of distributed decision making is neglected. We identified mainly following reasons: First of all, swarm behavior emerging from local interactions is hard to predict and a proof of its eligibility for applications in an industrial context is difficult to provide. Second, current communication architectures often do not match requirements for swarm communication, which often leads to a system with a centralized communication infrastructure. Finally, testing swarms for real industrial applications is an issue, since deployment in a productive environment is typically too risky and simulations of a target system may not be sufficiently accurate. In contrast, the research platforms present a means for transforming swarm robotics solutions from theory to prototypal industrial systems.

Keywords: swarm intelligence, swarm robotics, swarm behavior, swarm robotic applications, cyber-physical systems

1. INTRODUCTION

Swarms typically consist of many individual, simple, and homogeneous or heterogeneous agents (Dorigo and Birattari, 2007). They traditionally cooperate without any central control, and act according to simple and local behavior. Only through their interactions a collective behavior emerges that is able to solve complex tasks. These characteristics lead to the main advantages of swarms: adaptability, robustness, and scalability. Swarms can be considered as a

kind of quasi-organism that can adapt to changes in the environment by following specific behaviors (Hamann and Schmickl, 2012), e.g.:

- Pursuing a specific goal.
- Aggregating or dispersing in the environment.
- Communicating (direct, indirect).
- Memorizing (local states, morphologies).

In swarm robotics, multiple robots—homogeneous or heterogeneous—are interconnected, forming a swarm of robots. Since individual robots have processing, communication and sensing capabilities locally on-board they are able to interact with each other, and react to the environment autonomously.

In this paper, we focus on swarm intelligence applied in the swarm robotics domain. The theoretical and mathematical foundations of traditional swarm algorithms are out of the scope of this paper, as this was already done by multiple other authors at a greater level of detail. For example, Bonabeau et al. (1999) depict phenomena in social insects that had been transferred successfully to algorithms. Biological swarm behaviors from which a number of computational algorithms were developed are also discussed by Parpinelli and Lopes (2011). Camazine et al. (2001) discuss general self-organization aspects in biological systems. Moreover, Garnier et al. (2007) provide a good overview of the biological principles of swarm intelligence. Floreano and Mattiussi (2008) discuss swarm intelligence alongside evolutionary computation, artificial neural networks, and bio robotics. Blum and Li (2008), Binitha and Sathya (2012), and Krause et al. (2013) address swarm intelligence algorithms for optimization. Hassanien and Alamry (2015) depict the natural inspirations of swarm intelligence-based optimization algorithms. Swarm intelligence-based optimization algorithms are analyzed by Yang et al. (2013), the link between swarm intelligence-based optimization algorithms and self-organization is examined by Yang et al. (2017). Rossi et al. (2018) classify existing multi-agent algorithms according to their underlying mathematical structure.

Despite the large number of swarm algorithms, the step to industrial applications has not been mastered successfully, yet. In our research work on real-world applications, we noticed that oftentimes industry applications use the term “swarm,” but typically do not implement particular swarm algorithms. They rather use parts of swarm algorithms and implement them using centralized control. We refer to such parts of swarm algorithms as basic swarm behaviors in the following.

The rest of the paper is organized as follows: In section 2 we propose a taxonomy of basic swarm behaviors. In section 3 we show where these behaviors are applied by giving a comprehensive overview on current swarm robotics research platforms, projects, and products. This overview is complemented by a discussion that analyses the current situation, and explores open challenges in swarm robotics applications. Finally, in section 4 we conclude the paper.

2. BASIC SWARM BEHAVIORS FOR SWARM ROBOTICS

In most swarm algorithms, individuals perform according to local rules and the overall behavior emerges organically from the interplay of the individuals of the swarm. Translated to the swarm robotics domain, individual robots exhibit a behavior that is based on a local rule set which can range from a simple reactive mapping between sensor inputs and actuator outputs to elaborate local algorithms. Typically, these local behaviors incorporate interactions with the physical world, including the environment and other robots (Floreano and Mattiussi, 2008). Each interaction consists of reading and interpreting the sensory data, processing this data, and driving the actuators accordingly. Such a sequence of interactions is defined as basic behavior that is repeatedly executed, either indefinitely or until a desired state is reached.

In the following subsections, we classify and list the basic swarm behaviors which are adapted and expanded from Brambilla et al. (2013) with additional swarm robotic behaviors including collective localization, collective perception, synchronization, self-healing, and self-reproduction. The behaviors are explained from a high level view describing the task of individual robots and the resulting global objective achieved by the swarm. We do not detail on the sensing and actuation part which is specific to each robotic platform.

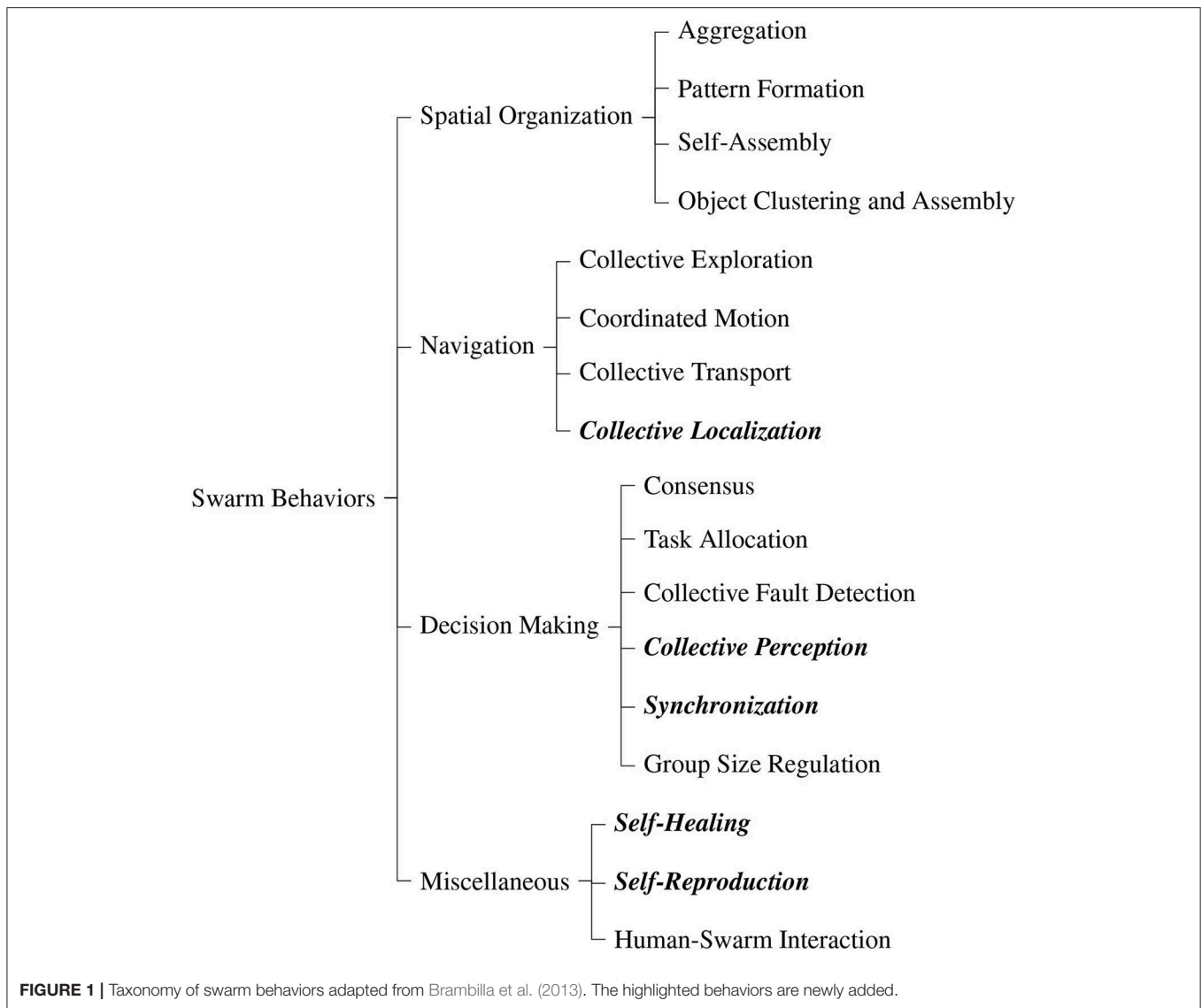
2.1. Taxonomy

The taxonomy of swarm behaviors is given in **Figure 1**. It is based on the classification by Brambilla et al. (2013) which we extended by several categories. In the following, we first give an overview of the taxonomy and then an in-detail description of the additional behavior categories in subsection 2.2. For these behaviors we also give the original inspiration. For a detailed description of the existing categories we kindly refer the reader to Brambilla et al. (2013).

2.1.1. Spatial Organization

These behaviors allow the movement of the robots in a swarm in the environment in order to spatially organize themselves or objects.

- **Aggregation** moves the individual robots to congregate spatially in a specific region of the environment. This allows individuals of the swarm to get spatially close to each other for further interaction.
- **Pattern formation** arranges the swarm of robots in a specific shape. A special case is chain formation where robots form a line, typically to establish multi-hop communication between two points.
- **Self-assembly** connects the robots in order to establish structures. They can either be connected physically or virtually through communication links. A special case is morphogenesis where the swarm evolves into a predefined shape.



- **Object clustering and assembly** lets the swarm of robots manipulate spatially distributed objects. Clustering and assembling of objects is essential for construction processes.

2.1.2. Navigation

These behaviors allow the coordinated movement of a swarm of robots in the environment.

- **Collective exploration** navigates the swarm of robots cooperatively through the environment in order to explore it. It can be used to get a situational overview, search for objects, monitor the environment, or establish a communication network.
- **Coordinated motion** moves the swarm of robots in a formation. The formation can have a well-defined shape, e.g., a line, or be arbitrary as in flocking.
- **Collective transport** by the swarm of robots enables to collectively move objects which are too heavy or too large for individual robots.

- **Collective localization** allows the robots in the swarm to find their position and orientation relative to each other via establishment of a local coordinate system throughout the swarm. See section 2.2 for more details.

2.1.3. Decision Making

These behaviors allow the robots in a swarm to take a common choice on a given issue.

- **Consensus** allows the individual robots in the swarm to agree on or converge toward a single common choice from several alternatives.
- **Task allocation** assigns arising tasks dynamically to the individual robots of the swarm. Its goal is to maximize performance of the entire swarm system. If the robots have heterogeneous capabilities, the tasks can be distributed accordingly to further increase the system's performance.
- **Collective fault detection** within the swarm of robots determines deficiencies of individual robots. It allows to

determine robots that deviate from the desired behavior of the swarm, e.g., due to hardware failures.

- **Collective perception** combines the data locally sensed by the robots in the swarm into a big picture. It allows the swarm to make collective decisions in an informed way, e.g., to classify objects reliably, allocate an appropriate fraction of robots to a specific task, or to determine the optimal solution to a global problem. See section 2.2 for more details.
- **Synchronization** aligns frequency and phase of oscillators of the robots in the swarm. Thereby, the robots have a common understanding of time which allows them to perform actions synchronously. See section 2.2 for more details.
- **Group size regulation** allows the robots in the swarm to form groups of desired size. If the size of the swarm exceeds the desired group size, it splits into multiple groups.

2.1.4. Miscellaneous

There are further behaviors of swarm robots that fit neither of the categories above.

- **Self-healing** allows the swarm to recover from faults caused by deficiencies of individual robots. The goal is thus to minimize the impact of robot failure on the rest of the swarm to increase its reliability, robustness, and performance (see also collective fault detection above). See section 2.2 for more details.
- **Self-reproduction** allows a swarm of robots either to create new robots or replicate the pattern created from many individuals. The goal is to increase the autonomy of the swarm by eliminating the need of a human engineer to create new robots. See section 2.2 for more details.
- **Human-swarm interaction** allows humans to control the robots in the swarm or receive information from them. The interaction can happen remotely, e.g., through a computer terminal or proximal in a shared environment, e.g., through visual or acoustic clues.

2.2. Detailed Description of Additional Swarm Behavior Categories

In the following, we describe the additional categories of basic swarm behaviors with which we extended the taxonomy by Brambilla et al. (2013), namely: Collective localization, collective perception, synchronization, self-healing, and self-reproduction.

2.2.1. Collective Localization

Collective localization allows the robots in the swarm to find their position and orientation relative to each other via establishment of a local coordinate system throughout the swarm.

2.2.1.1. Sources of inspiration

The approaches given below are engineered without any mentioned inspiration.

2.2.1.2. Approaches

There are two approaches which originate from the multi-robot research domain. First, creating a map of the environment and localizing relative to it. This approach is called simultaneous localization and mapping (SLAM). It can use and merge different sources of information, such as range sensors or visual sensors.

Second, using stationary landmarks with known positions and localizing relative to them. To avoid relying on external information, other robots can be used as landmarks. The robots can move alternately through the environment while keeping precise localization information. If the initial positions of the robots are known, then also absolute localization is possible. The dead-reckoning approach where robots use odometry for localization is another possibility but introduces an accumulating error which renders it useless for most scenarios.

2.2.1.3. Results

Thrun et al. (2000) present a mapping algorithm where multiple robots can localize in a globally fused map. It requires that the approximate initial positions of the robots are known by all other robots. It uses an incremental expectation maximization approach which allows robots to localize themselves in maps created by other robots. Experiments demonstrate that the robots can localize robustly in real time in large-scale environments using low-end computers. In the follow-up work, the requirement of known initial positions is relaxed, assuming that robots share an overlapping part of their explored maps (Thrun and Liu, 2005). It employs the concept of information filters that represents the robot positions by Gaussian Markov random fields. The robots are able to identify the correct alignments between different local maps by maximizing the correspondence of similar-looking landmark configurations. Fox et al. (2000) present a belief-based approach for collaborative multi-robot localization. It fuses localization information from different sources, such as odometry, environment measurements, and mutual robot detections by combining visual and range sensors. This allows to improve the robots' belief of the world by learning the detection model from data using a maximum likelihood estimator. Experiments demonstrate that a team of robots is superior in localization compared to single robots with a relatively small communication overhead. Kurazume and Hirose (2000) propose the method of cooperative positioning using robots as landmarks. There are two groups of robots that move alternately while using the other, stationary group as localization reference. An increased number of robots also increases the redundancy of position information. With the weighted least square method this redundancy decreases the localization error. The authors perform experiments where the robots use range sensors to measure their respective positions. Results show that this localization method performs better than the dead reckoning method including environments with uneven terrain. Rekleitis et al. (2001) propose a method where robots visually observe each other to improve the dead reckoning localization. They propose two algorithms based on triangulation and trapezoidation for small and large-scale environments, respectively. They perform experiments with two robots where one robot carries markers and the other a camera. The results demonstrate that joint localization leads to much more robust localization than odometry alone. When more robots are used the localization precision is increased. Howard et al. (2003) propose a maximum likelihood method combined with a distributed numerical optimization to eliminate the need for landmark robots to be stationary. It combines range measurements with

odometry. Robots observe each others motion and exchange this information to create a graph consisting of their positions and respective observations. Experiments with four robots demonstrate that this method is able to localize with adequate precision, and is robust to changes in the environment and to flawed odometry. Furthermore, robots are able to infer the position of other robots they have never seen before. Rubenstein et al. (2014b) apply the robot-landmark approach to a large swarm of 1,024 Kilobots. There are four pre-localized seed robots which define the coordinate system. The other robots localize relative to these seed robots using trilateration of infrared signals. The robots were able to self-assemble and let the swarm morph to a given shape.

2.2.2. Collective Perception

Collective perception combines the data locally sensed by the robots in the swarm into a big picture. It allows the swarm to make collective decisions in an informed way, e.g., to classify objects reliably, allocate an appropriate fraction of robots to a specific task, or to determine the optimal solution to a global problem.

2.2.2.1. Sources of inspiration

Many social insects are able to get a global view using only local information. Examples are honeybees that assess the current global workload balancing by evaluating simple cues like queuing delays (Ratnieks and Anderson, 1999) and ants that use pheromone trails to find shortest paths in large environments (Goss et al., 1989; Hölldobler and Wilson, 1994).

2.2.2.2. Approaches

For collectively determining the type of object observed, the predominant approach is classification of the object among a set of predefined models. Sometimes, the mobility of the robots is used to improve the perception of individual robots. The robots use explicit communication in order to propagate their findings and achieve consensus. The way the robots exchange the information is an important aspect. They have to add contextual information that allows the other robots to correctly interpret the data. Furthermore, the information can be simply forwarded and thus spread in the swarm or modified in order, e.g., to measure the distance to a specific location. There are also approaches from other research domains, such as camera networks (Schranz and Rinner, 2015), but the agents are typically stationary and often centrally controlled.

2.2.2.3. Results

Ye et al. (2002) propose a strategy where sensing agents collect, analyze, and categorize data, enrich it with contextual information, and forward it to synthesizing agents. The latter are then able to use the different aspects observed by the sensing agents to perceive events using an eigen-space method. Using simulation experiments, the authors demonstrate that events can be detected reliably using only the first few eigen values. Kornienko et al. (2005) develop a swarm of micro-robots for a collective classification task. Based on evidence theory, the swarm has to identify the geometries of objects in space by exchanging data from infrared depth sensing while having limited communication capabilities. Experiments show that a

wrong belief of an individual quickly converges to the correct belief after exchanging only few messages. King and Breedon (2010) present a simple model of a hexagonal grid world in which a swarm has to differentiate between differently shaped objects. They show that an increased number of agents leads to an overproportional decrease of object detection time. Giusti et al. (2012) present an approach for cooperative gesture recognition with a robot swarm. Each robot processes and classifies camera images locally. Using a distributed consensus protocol, the robots exchange their opinions over a low-bandwidth wireless channel to find a common decision by exploiting the different view points and mobility of the agents. The approach is evaluated through simulation and physical experiments on 13 robots. The results show that the recognition accuracy of the system scales effectively with the number of agents and is robust to communication failures. Stegagno et al. (2014) develop a method that allows a swarm of robots with different types of low resolution sensors to collectively classify objects. Each robot processes the sensor data locally and exchanges its estimation. Using the naive Bayes classifier together with the information received from other robots, the swarm is able to robustly classify objects. The more diverse the sensors are, the better the results. Olfati-Saber and Jalalkamali (2012) present a theoretic framework that employs mobility to improve the information sensed by the swarm using the Kalman-consensus filter. It is employed to track a target with a swarm of agents. Each agent tries to improve its sensing while avoiding collisions with the others. Simulations show that this solution can effectively track linear and non-linear maneuverable targets.

Mazdin and Rinner (2019) present a method for simultaneous coverage of surfaces with a swarm of robots. This method assigns robots to different view points in order to allow effective 3D reconstruction of objects. Simulation results show that this method is able to coordinate the robots while minimizing the mission duration and maximizing the coverage quality. Schmickl et al. (2007) compare different communication strategies for collective perception of a swarm, namely the hop-count strategy and the trophallaxis-inspired strategy. The presented solutions allow a swarm to collectively compare sizes of target areas which are too large for individual robots to perceive. Simulations show that the robots are able to aggregate in the target areas while their numbers are proportional to the size of the target area. Mermoud and Evans (2010) tackle a similar problem in which robots should distinguish good and bad spots using models of chemical reaction networks. They perform experiments with five robots and demonstrate that robots with limited sensing capabilities can collectively achieve good performance.

2.2.3. Synchronization

Synchronization aligns frequency and phase of oscillators of the robots in the swarm. Thereby, the robots have a common understanding of time which allows them to perform actions synchronously.

2.2.3.1. Sources of inspiration

During courtship, males of certain animal species synchronize their behavior. In some firefly species, the phase difference between the blinking of male and female flashing period

is important for mating (Buck, 1988). Hence the fireflies synchronize by influencing their flashing phase. Likewise, bushcrickets synchronize or alternate using chirps by altering their chirp periods in response to other chirps (Hartbauer et al., 2005). Igoshin et al. (2001) develop a model that describes the spatio-temporal wave patterns observed from myxobacteria cells. There are many more examples of coupled oscillating systems, e.g., pacemaker cells in the heart (Peskin, 1975) or clapping of spectators in a theater (Néda et al., 2000).

2.2.3.2. Approaches

The oscillators are synchronized to the same frequency with the phases being aligned among the robots in the swarm. Two approaches exist, either the oscillators continuously influence each other to adjust phase and frequency, or they are pulse-coupled meaning that they regularly fire a signal corresponding to their current phase. The latter one is mostly used as it requires fewer interactions between the robots. Robots interact either through acoustic, visual signals or radio communication.

2.2.3.3. Results

Hartbauer and Römer (2006) employ synchronized oscillators as a communication and navigation system. In a synchronized system, robots at a target area increase their frequency and thereby produce phase waves in the swarm that can be used by the robots to perform wave-front navigation, i.e., travel toward higher frequencies. The authors analyze the robustness of the system by simulating up to 300 robots. The results show that this communication system is robust to changes in signal strength, signaling period length, and communication obstacles. Nevertheless, the signaling period is an important parameter to be fit to the scenario. They conclude that pulse-coupled oscillator synchronization is especially suited for swarms of robots as it has low hardware requirements in terms of communication range and processing power. Christensen et al. (2009) apply synchronization to detect faulty robots in the swarm. Using pulse-coupled oscillators and visual signaling, the swarm can determine malfunctioning robots when their phases are not aligned to the rest of the swarm. The authors develop a discrete model that can be applied to robots. Simulations with 100 robots show that robots synchronize faster when they are mobile, synchronization time is linearly proportional to the swarm size where denser networks synchronize faster, and synchronization is robust to communication obstacles but decreases in performance. Experiments with ten physical robots confirm the simulation results, despite the inherent latencies associated with the sensor and actuator systems. In contrast to the bio-inspired approaches, Trianni and Nolfi (2009) synthesize synchronization strategies using artificial evolution. These strategies perform phase coupling between robots in order to allow synchronous movement. Simulations with up to 96 robots show that the strategies scale well and are mainly limited by collision avoidance behaviors. This is confirmed through experiments with up to three robots where sensor and actuator noise is introduced.

Bezzo et al. (2014) synchronize robots in a swarm to determine the network topology and detect changes. They develop a strategy

for estimating the degree of oscillator coupling in the swarm and synchronizing them continuously. Applying this strategy to formation control allows three simulated robots to move in a formation. Simulations with five robots show that the network topology can be detected reliably. Barciś et al. (2019) apply the novel concept of swarmalators (O’Keeffe et al., 2017) to robots. This concept couples the oscillator phase with spatial location in such a way that they mutually influence each other. They modify the original model taking into account the discrete nature of robots. Simulations of 100 robots and experiments with 10 robots show that the spatio-temporal patterns can be performed in stationary and dynamic scenarios. Perez-Diaz et al. (2015) perform a case study to analyze how motion and sensing capabilities influence the synchronization capabilities of a robot swarm. By altering the field of interaction (e.g., camera field of view) and the speed at which the robots travel, the emergence of synchrony can be influenced. The robot speed influences the time until synchrony is reached whereas a narrow field of interaction results in a low degree of synchronization. Furthermore, high robot densities limit the synchronization possibility due to signal occlusion and robot collisions.

2.2.4. Self-Healing

Self-healing allows the swarm to recover from faults caused by deficiencies of individual robots. The goal is to minimize the impact of robot failure on the rest of the swarm to increase its reliability, robustness, and performance. After detecting the fault, appropriate countermeasures must be taken.

2.2.4.1. Sources of inspiration

The immune system of vertebrates shows how biological systems protect complex organisms against diseases. This serves as inspiration for the artificial immune system (AIS). Timmis et al. (2010) give an overview of how AISs and swarm intelligence relate. They point out many similarities and conclude that both systems are complementary tools for solving complex engineering problems. Regeneration in biological systems allows animals to self-heal their body, e.g., salamanders, starfish, and lizards are able to regenerate lost limbs (Wallace, 1981). Another prominent example is the morphallaxis, i.e., tissue regeneration of Cnidarian hydra Shimizu et al. (1993). It exhibits what is sometimes referred to as scalable self-healing: When the hydra is dissected, each part can self-heal into a fully functional and independent hydra where its size is proportional to the number of cells.

2.2.4.2. Approaches

There are two ways to tackle the problem of self-healing. First, healthy robots can aid the faulty robots in recovering. It requires an explicit failure management routine which is typically inspired by the immune system. Second, the swarm can adjust its behavior while ignoring failing robots. This does not require any special handling of the failure case. It is typically inspired by biological regeneration.

2.2.4.3. Results

As self-healing is a relatively challenging topic for swarm robotics, only few embodied studies exist and most work is

done through simulation experiments. Dai et al. (2006) present a model for detecting and healing software components of swarm robots. It is part of the NASA autonomous nano technology swarm (ANTS) concept mission (Vassev et al., 2012). This model is only partly distributed as it relies on a central cyber disease library. Each robot runs one or more virtual neurons as background processes. They monitor certain system variables, such as CPU, memory, or network usage. In case of anomalies, it freezes the process in question and reports its behavior to a higher-level controller. It can perform further diagnosis, e.g., by assigning more neurons, and generate a prescription based on the cyber disease library. The prescription is applied by the executor process which reports back results. This allows the cyber disease library to learn and improve prescriptions. In case the prescription does not work, further escalation steps are possible, such as killing the faulty process or even rebooting the whole machine. A simulation case study shows that a memory leaking process is successfully detected and eventually killed. The results show that the system becomes more reliable and robust against failures and failure propagation. Even though faulty processes degrade the overall system performance, the performance improves compared to systems without the self-healing properties. Timmis et al. (2016) apply self-healing to overcome hardware failures. They present a solution that is inspired by granuloma formation, a process of containment and repair found in the immune system. They apply it to a swarm of robots performing flocking and taxis toward a beacon. When a robot has a discharged battery and loss of mobility, it would anchor the whole swarm which would then fail to reach the beacon. The proposed solution allows energy sharing between healthy and faulty robots. The faulty robots signal their need of help to the other robots within range. Depending on the required energy and the energy available at the healthy robots, a varying number of robots surround the discharged robots to share energy. Other robots ignore this robot cluster and regard it simply as an obstacle, continuing their mission. Simulation experiments with 10 robots show that the granuloma formation algorithm works well even when half of the robots in the swarm are experiencing low energy levels. Other algorithms are compared where only the nearest healthy robots perform the healing. They fail to heal the swarm when three or more robots have a discharged battery.

A broad body of research is directed toward pattern formation and morphogenesis in self-healing. Cheng et al. (2005) propose the SHAPEBUGS approach where agents evenly disperse within a predefined shape. First, the agents perform trilateration to establish a common coordinate system. This is aided by allowing a few agents to know their initial position. Then the agents use the contained gas model to move in a way that they are equally spaced within the desired shape. In case of agent failure, the other agents simply adapt their positions to again reach an equilibrium density. The agent model contains a proximity sensor and wireless communication to exchange positions. It furthermore requires a compass to determine the global orientation of the agents. Simulations show that the swarm can self-repair and restabilize in cases of agent death or displacement. Furthermore, it can overcome large degrees of sensor and movement errors

of the agents. Rubenstein and Shen (2008) relax some of these assumptions and still achieve similar results. The model differs in that the robots build compact shapes. Thereby, the size of the shape varies rather than its density. The model requires only a single sensor for local information and communication. The shape is given to the robots as a potential field where the robots aim to move to its center while avoiding collisions. The scale of the shape is calculated as function of the estimated swarm size. Additionally, each robot changes its color to a color that is predefined depending on the position. Thereby, the robots can form colored patterns or displays. When properly synchronized, they can even show time varying patterns. Simulation results show that the robots can perform scalable self-healing by recreating the desired shape for varying swarm sizes. In later work, Rubenstein et al. (2014b) demonstrate this on a large swarm of physical robots as described above. Arbuckle and Requicha (2010) propose a model where the robot swarm builds shapes by arranging on the boundaries of a polygon. They propose an external compilation process that uses the polygon to derive a set of parameterized local rules to be executed by a swarm of homogeneous, stateless robots. By attaching physically to each other, the robots can communicate directionally. The agents stay connected as long as they are communicating. By replying with predefined messages, the state of the system is “externalized” in the circulating messages. By attaching to each other, the agents grow the edges of the polygon while randomly wandering robots “diffuse” into the interior of the polygon by replacing boundary robots that in turn move into the polygon interior. Simulation experiments show that the swarm can build simple polygons. It can heal from failures due to communication faults, such as dropping messages. Robots that do not communicate anymore, drop out of the shape and are replaced by new ones. If the shape is broken into two, the swarm creates two shapes with the original size.

2.2.5. Self-Reproduction

Self-reproduction allows a swarm of robots either to create new robots or replicate the pattern created from many individuals. In the first case, the robots produce identical copies of themselves. The goal is to increase the autonomy of the swarm by eliminating the need of a human engineer to create new robots. In the second case, the robots copy a structure consisting of many individual robots. Existing approaches are not fully autonomous yet and typically require at least the building blocks to be provided to the swarm. In contrast to self-reconfiguration of formed patterns, the goal of self-replication is to assemble a functional robot from passive components.

2.2.5.1. Sources of inspiration

All biological organisms possess the ability to reproduce, either sexually or asexually.

2.2.5.2. Approaches

The theory of self reproducing machines already exists for several decades, e.g., von Neumann (1966) introduced the idea of an automaton model for self-reproduction. The research in this direction followed the general idea of template-replicating

systems, i.e., to create a new robot according to an existing model. Gross and Dorigo (2008) give an historic overview of the development in this direction. Other approaches follow the evolutionary design strategy. The existing approaches assume the robot hardware to be modular in order to have base building blocks (Yim et al., 2002). The finer the modules, the more difficult the process, but the more flexibly a new robot can be created.

2.2.5.3. Results

Lipson and Pollack (2000) evolve the design for simple electromechanical systems through simulation experiments. The building blocks are bars and actuators that are connected through joints and controlled by artificial neurons. The performance of the evolved systems is measured in terms of the distance it is able to locomote. The best performing designs are fabricated using rapid, additive manufacturing technology. This process allows robots to design new robots with minimized human intervention. The manufactured robots are able to locomote similarly to the simulation models. Suthakorn et al. (2003) present a fully autonomous, self-replicating robot built from Lego parts. It assembles a new robot from four pre-assembled subsystems that hold together using magnets and shape-constraining blocks. The controller of the replica is already pre-programmed with the same program as the original. Experiments show that the original robot, which is guided by lines on the ground, is able to detect the subsystems and assemble them into a fully functional replica. Zykov et al. (2007) present the design of a modular robot cube, called Molecube. These cubes can attach to each other using electromagnets and hence form complex patterns. These cubes have one actuated degree of freedom to control the shape of the assembled pattern. The authors present experiments where the spatial patterns and corresponding controllers are both manually created and automatically designed with artificial evolution. The results show that the swarm of Molecubes reproduces identical copies of its pattern, both in simulation and physical experiments. The only human interaction is by providing enough Molecubes as building material. The authors conclude that the more units are involved and the simpler and more homogeneous they are, the more information is being reproduced by the system itself, as compared with information pre-existing in the parts and environment.

3. SWARM APPLICATIONS

Even though swarm robotics is a relatively young field of research and has not been widely accepted in industry, this section is a first collection of existing applications and attempts at swarm robotic products. Swarm robotic researchers have designed and developed a number of platforms to test and analyze swarm algorithms. In their publications the authors always stated their attempt to envision future industry applications (Sharkey, 2007) out of the simplicity of swarm robotic research platforms. Thus, we split our survey into swarm robotics research platforms (Table 1) and industrial projects and products (Table 2). The industrial projects and products are mainly listed to serve as application examples in real-world environments above a technology readiness level (TRL) of four (Héder, 2017) where

the validation of the platform is already in the relevant environment. The research platforms enable researchers to verify, demonstrate, and experiment with swarm algorithms in laboratory environments, thus on a TRL of maximum four.

Both, research platforms and industrial projects and products are categorized according to the environment they are used in: terrestrial, aerial, aquatic, and/or space. All tables list the type of the application, the project or product name, the type of robot, the number of robots in the swarm, and the basic swarm behaviors corresponding to the definitions of section 2. The type of robot corresponds to one of the following categories: unmanned ground vehicle (UGV), unmanned aerial vehicle (UAV), unmanned surface vehicle (USV), unmanned underwater vehicle (UUV), or in general as UxV. The number of research platforms used in a swarm is proven with specific research publication. For the industrial swarm applications we refer the reader to the project's or product's website for this information. For each research platform, industrial project and product we list one or more basic swarm behaviors because no project or product uses integral swarm algorithms. They rather use parts of the algorithms, and adapt them to the underlying application. Additionally, the table for the research platforms differentiates between open-source (in hardware and software) and/or commercially available. We do not classify the swarm robotic platforms and products related to their price, dimension, and number of usage in different research/engineering projects.

The focus of this collection is on basic swarm behaviors embodied on robots. Therefore, we only list projects or products that are based on a robotic platform. We neglect research projects that focus solely on swarms with a purely theoretical or virtual nature.

3.1. Research Platforms

This section presents research platforms that are developed for educational and scientific purposes, summarized in Table 1. They allow to investigate the application of swarm algorithms to robots. Note, that other sophisticated robotic research platforms exist which are not included, as they are not developed with the intention of using them in swarm applications, e.g., the Balboa robot kit¹.

3.1.1. Terrestrial

The Kilobots swarm (Rubenstein et al., 2014a) is probably the best known swarm of robots developed for research and education. Kilobots are very small with a diameter of 33 mm, the locomotion is based on vibration motors and the communication is implemented using infrared light reflecting off the ground. They became very famous for their self-assembling capability forming different shapes with a swarm of 1,024 Kilobots (Wyss Institute, 2017). The Kilobot is available open-source² or commercially at K-Team³.

Jasmine is another widely used swarm robotic platform. The open-source⁴ platform was mainly built for large-scale

¹Balboa robot website: <https://www.pololu.com/product/3575>

²Kilobot website: <http://www.kilobotics.com/>

³K-Team website: <https://www.k-team.com/>

⁴Jasmine website: <http://www.swarmrobot.org/>

TABLE 1 | Classification of research platforms for swarm robotics.

Environment	Application	Project/Product name	Robot type	No. of robots	Basic swarm behaviors	Availability
Terrestrial	Research and Education	Kilobots	UGV	1,024 (Rubenstein et al., 2014a)	Pattern formation, coordinated motion	Open-source, commercial
		Jasmine		60 (Kernbach et al., 2013)	Aggregation, collective exploration, coordinated motion, task allocation, collective perception, self-healing (partially), human-swarm interaction (Zoids, APIS)	Open-source
		Alice		20 (Garnier et al., 2008)		n.a.
		AMiR		6 (Arvin et al., 2011)		n.a.
		Colias		14 (Arvin et al., 2014)		Open-source, commercial
		Mona		30 (Arvin et al., 2019)		n.a.
		R-One		n.a.		n.a.
		Elisa-3		38 (eli, 2014)		Open-source, commercial
		Khepera IV		10 (Pincirolì and Beltrame, 2016)		Commercial
		GRITSbot		100 (Pickem et al., 2017)		Open-source
		E-Puck		16 (Alkilabi et al., 2017)		Open-source, commercial
		Xpuck		16 (Jones et al., 2018)		n.a.
		Thymio II		8 (Vitanza et al., 2019)		Open-source, commercial
		Pheeno		4 (Wilson et al., 2018)		Open-source
		Spiderino		n.a.		n.a.
		I-Swarm		n.a.		n.a.
		Zoids		32 (Le Goc et al., 2016)		Open-source
		APIS		6 (Dhanaraj et al., 2019)		n.a.
		Wanda		11 (Kettler, 2012)		
		Droplet		n.a.	Aggregation, self-assembly, object clustering and assembly, collective exploration, coordinated motion, collective transport (partially), collective perception	Open-source
		Swarm-bot		35 (Groß et al., 2006)		
		Swarmanoid		n.a.		n.a.
		Termes		5 (Petersen et al., 2011)		
		Symbion and Replicator		n.a.		Open-source
		PolyBot		32 (Duff et al., 2001)		
		M-Tran III		24 (Kurokawa et al., 2008)		n.a.
		ATRON		7 (Brandt et al., 2007)		
		CONRO		8 (Castano et al., 2002)		n.a.
		Sambot		15 (Wei et al., 2010)		
		Molecube		8 (Zykov et al., 2007)	Pattern formation, self-assembly, self-reproduction	
Aerial		MAV	UAV	n.a.	n.a.	n.a.
		Distributed Flight Array		9 (Oung, 2013)	Self-assembly, coordinated motion	
		Crazyflie 2.1		49 (Preiss et al., 2017)	Aggregation, collective exploration, coordinated motion, collective localization, collective perception	Open-source, commercial
		FINKen-III		n.a.		n.a.
Aquatic	Environmental Monitoring	CoCoRo	UUV	41 (Schmickl, 2015)	Aggregation, collective exploration, collective localization, task allocation	n.a.
		Monsun		n.a.		Open-source
		CORATAM	USV	12 (Duarte et al., 2016)		
Outer Space	Space Exploration	Swarmies	UGV	20 ^a	Collective exploration, collective localization	n.a.
		Marsbee	UAV	3 (Kang, 2018)	Collective exploration, coordinated motion, task allocation	

^aProject's official website: <http://nasaswarmathon.com/>.

swarm robotic experiments equipped with a series of sensors, including touch, proximity, distance, and color sensors. The intention for large-scale swarms was also pursued with the

swarm robotic platform Alice (Caprari et al., 1998). Many additional sensors including a linear camera can extend the basic capabilities. A series of research platforms building upon each

TABLE 2 | Classification of industrial projects and products.

Environment	Application	Project/Product name	Robot type(s)	No. of robots	Basic swarm behaviors
Terrestrial	Agriculture	SwarmBot 3.0	UGV	5	Coordinated motion, task allocation
		Xaver		10	
	Emergency and Rescue	GUARDIANS		4	Pattern formation, collective exploration, coordinated motion
	Warehouse	Ocado, Amazon (Kiva), Alibaba		Max. 1,100 per warehouse	Coordinated motion, task allocation, self-healing
	Industrial Plant	SWILT	UxV	1,500 machines	Object clustering, collective exploration, group size regulation
Aerial	Military	OFFSET	UGV, UAV	250	Collective exploration, coordinated motion, task allocation, human-swarm interaction
		Perdix	UAV	103	Aggregation, pattern formation, coordinated motion, consensus, self-healing
	Emergency and Rescue	SMAVNET		19	Pattern formation, collective exploration, task allocation, collective perception
		SWARMIX		n.a.	
		CPSwarm		n.a.	
	Agriculture	SAGA		n.a.	Collective exploration, coordinated motion, collective localization, task allocation
	Entertainment	Spaxels		100	Pattern formation
		Flyfire		n.a.	
		Ehang GhostDrone 2.0		1,000	
		Intel Shooting Star		500	
		Lucie micro drone		n.a.	
Aquatic	Environmental Monitoring	Platypus	USV	25	Collective exploration, human-swarm interaction
		Apium Data Diver	USV, UUV	50	Pattern formation, collective exploration, coordinated motion, human-swarm interaction
		subCULTron	UUV	n.a.	Aggregation, collective exploration, collective localization
		Vertex Swarm		10	Pattern formation, collective exploration, coordinated motion, collective localization
		SWARMS	UUV, USV	8	
	Military	CARACaS	USV	5	
Terrestrial/ Aerial/ Aquatic	Surveillance	ROBORDER	UxV	n.a.	n.a.
	Maintenance	BugWright2			Aggregation, collective exploration, coordinated motion, task allocation
	Multiple	Sentien Robotics	UGV, UAV		Coordinated motion, task allocation, self-healing
Outer Space	Space Exploration	Swarm	UAV	3	Aggregation, coordinated motion
		Cluster II		4	

other is given with AMiR (Arvin et al., 2009), Colias (Arvin et al., 2014) (open-source⁵ and commercially⁶ available), and Mona (Arvin et al., 2017) (open-source⁷ and commercially⁸ available). The platform R-One (McLurkin et al., 2013) is also designed for the usage as swarm robotic platform. Although it uses a camera tracking system for ground-truth localization, and server software to connect all the pieces together, several experiments “close” to swarm intelligence can be performed. The Elisa-3 swarm robotic platform, open-source and commercially⁹ available, also uses an Arduino microcontroller with a high

number of sensors including eight IR proximity sensors, three-axis accelerometer, and four ground sensors. The robot is able to recharge autonomously using a charging station. The robots in the swarm communicate using either IR or radio. The Khepera IV (Soares et al., 2016) is designed for any indoor lab application. A Linux core, color camera, WLAN, Bluetooth, USB Host, accelerometer, gyroscope, microphone, loudspeaker, three top RGB LEDs, and improved odometry makes it a compact and complete research platform for swarms in different scenarios. The Khepera IV is commercially available at K-Team¹⁰. The GRITSbot (Pickem et al., 2015) is the open-source¹¹ swarm

⁵Colias open source website: <https://github.com/MonaRobot/Colias>

⁶Colias commercial website: <http://www.visomorph.com/>

⁷Mona open source website: <https://github.com/MonaRobot>

⁸Mona commercial website: <https://ice9robotics.co.uk/>

⁹Elisa-3 website: <https://www.gctronic.com/doc/index.php/Elisa-3>

¹⁰K-Team website: <https://www.k-team.com/>

¹¹GRITSbot website: <https://www.wevolver.com/wevolver.staff/gritsbot/master/blob/Overview.md>

robotic platform used in the Robotarium¹² at Georgia Tech, Atlanta. The Robotarium provides remote access to a large team of robots. Scholars can upload code to run experiments remotely to collect data. Features like automatic registration of robots with a server, autonomous charging, wireless code upload to the robots, and automatic sensor calibration makes the Robotarium attractive for remote research experiments. All these platforms use wheels for their locomotion and a set of different sensors, including distance and light sensors.

The e-puck robot (Mondada et al., 2009), and its successor e-puck2, are designed as educational and research robots to make it easy to program and control the robots' behaviors. It uses diverse sensors, e.g., infrared proximity sensors, a CMOS camera, and a microphone. The e-puck is available open-source¹³ or commercially at GcTronic¹⁴. The Xpuck is an extension of the e-puck in terms of aggregate raw processing power (as used in modern mobile system-on-chip devices) of two teraflops. Thus, higher-individual robot computation can be achieved, e.g., image processing using the ArUco Marker tracking (Jones et al., 2018). The Thymio II robot (Riedo et al., 2013) targets the understanding of programming and robotic concepts using a wide range of sensors, including temperature, infrared distance, accelerometer, and microphone. Programming can be done in Blockly using visual or text-based programming. The Thymio II is available both open-source and commercially at Thymio¹⁵. A recent platform for open-source¹⁶ swarm robotics education and research purposes is called Pheeno (Wilson et al., 2016). The user can adapt the platform with custom modules in three degrees of freedom. To interact with the environment it uses IR sensors. The Spiderino platform (Jdeed et al., 2017) is a six-legged open-source¹⁷ robot with spider-like locomotion. It is based on a hexpod toy that is enhanced by a PCB including an Arduino microcontroller, a WLAN module, and several reflective infrared sensors.

The goal of the I-Swarm (Intelligent Small-World Autonomous Robots for Micro-manipulation) project is the development of micro robots to form a swarm. The robot has a small size of only $3 \times 3 \times 3 \text{ mm}^3$, is solar powered without battery, performs locomotion via vibration, and communication via infrared transceivers (Seyfried et al., 2004). It has been developed with the goal to build a swarm of 1,000 robots (Seyfried et al., 2004). Prototypes of this robot can be seen in the technical museum in Munich, Germany.

The idea behind the open-source swarm robotics platform Zooids¹⁸ is different: It handles both the interaction and the display, and thus offers a new class of human-computer interfaces. The swarm is controlled via light patterns projected using an overhead projector (Le Goc et al., 2016). The APIS¹⁹

(Adaptable Platform for Interactive Swarm) comprises several components: the swarm robotic platforms, the infrastructure and test environment for the swarm, and the software infrastructure and simulation (Dhanaraj et al., 2019). The focus are experiments related to human-swarm interaction. For these interactions, additionally to sensors, the platform is equipped with an OLED display and a buzzer. The Wanda (Kettler et al., 2012) swarm robotic platform has a special assembly that could be used, e.g., to clean up the environment with a swarm. In addition, the authors implemented a whole tool chain especially for these robots from design and simulation to deployment.

The Droplet (Klingner et al., 2014) is another swarm robotic platform for teaching and research. It is a spherical robot which is able to organize into complex shapes with its neighbors by using vibration locomotion. It charges and communicates via a powered floor that is equipped with alternating stripes of positive charge and ground. It is available as an open-source project²⁰. The Swarm-bots (Mondada et al., 2002; Groß et al., 2006) can configure themselves to different geometric 3D shapes. The robots are constructed by a number of simpler, insect-like robots, which are built of relatively cheap components (the design is open-source²¹). A swarm of these robots is capable of self-assembling and self-organizing to adapt to its environment. With this assembling capability the swarm is able to transport objects that would be too heavy for the individual robots. The successor of the Swarm-bots project is the Swarmanoid project, which represented the very first attempt to study the integrated design, development, and control of a heterogeneous open-source²² swarm robotics system. The swarm in the Swarmanoid project covers autonomous robots of three types (each with an additional set of sensors): eye-bots (UAVs that can attach to an indoor ceiling), hand-bots (UGVs capable of climbing), and foot-bots (UGVs capable of self-assembling) (Dorigo et al., 2013). The Termes robots (Petersen et al., 2011) collaborate without communication or GPS localization to create large structures using modular blocks. The underlying concept is stigmergy and is inspired by the way termites build their nests in nature. The Termes robots are block-carrying climbing robots that can create these structures in unstructured environments. Symbrion and Replicator (Kernbach et al., 2008) are two sister projects, that develop autonomous platforms for usage in swarms. They can be operated either individually or form special patterns by physically connecting to each other. The main goal of these projects is to create a road-map of how to achieve the evolvability of robot organisms. PolyBots (Duff et al., 2001) are self-reconfigurable robots. Various types of locomotion capabilities and object manipulation modules are interchangeable that allow to form a number of shapes, e.g., an earthworm type to slither through obstacles, or a spider to stride over hilly terrain. These robots find their application if the environment is unknown or if robots need to perform multiple tasks. Further modular robots that allow self-configuration with similar robotic technologies include

¹²Robotarium website: <https://www.robotarium.gatech.edu/>

¹³e-puck open source website: <http://www.e-puck.org/>

¹⁴e-puck commercial website: <https://www.gcTronic.com/e-puck.php>

¹⁵Thymio website: <https://www.thymio.org>

¹⁶Pheeno website: <https://discourse.ros.org/t/pheeno-a-low-cost-ros-compatible-swarm-robotic-platform/2698>

¹⁷Spiderino website: <https://spiderino.nes.aau.at>

¹⁸Zooids website: <https://github.com/ShapeLab/SwarmUI>

¹⁹APIS website: <https://github.com/wvu-irl/reu-swarm-ros> (software only).

²⁰Droplet website: <https://code.google.com/archive/p/cu-droplet/>

²¹Swarm-bots website: https://www.ercim.eu/publication/Ercim_News/enw53/nolfi.html

²²Swarmanoid website: <https://cordis.europa.eu/project/id/022888>

M-TRAN (Murata et al., 2002), M-TRAN II (Kurokawa et al., 2003) and M-TRAN III (Kurokawa et al., 2008) (available as open-source²³ project), ATRON (Brandt et al., 2007) (available as open-source²⁴), CONRO (Castano et al., 2002), Sambot (Wei et al., 2010), Molecube (Zykov et al., 2007), but to name a few²⁵.

3.1.2. Aerial

There are several miniature and micro UAVs which form a good basis for a swarm system of inexpensive robots for research and education. An overview of such small-scale UAVs can be found in the publications by Cai et al. (2014) and Swetha et al. (2018). Although multiple off-the-shelf Micro Air Vehicles (MAVs) exist and are quite popular in the games industry, and in businesses for video- and photography, they typically have closed flight controllers that do not allow to develop custom algorithms (e.g., Qualcomm Flight Pro²⁶, DJI M100²⁷). One UAV designed specifically for usage in swarms is the MAV presented by Roberts et al. (2007). The MAVs are equipped with three rate gyroscopes, three accelerometers, one ultrasonic sensor, and four infrared sensors. It has been developed within the Swarmanoid project (Dorigo et al., 2013). A very distinct research platform is the Distributed Flight Array (Oung and D'Andrea, 2011). Each UAV makes up a module of a larger array and has a single rotor only. The modules self-assemble into a multi-rotor system where all vehicles must cooperate for coordinated flight. To facilitate this, they exchange information among each other and adjust local parameters. With the Crazyflies, available both open-source and commercially at bitcraze²⁸, a swarm of UAVs can be realized indoors. They use multiple sensors, e.g., accelerometer, gyroscope, magnetometer, and a high precision pressure sensor (Preiss et al., 2017). Their low weight of 27 g allows experiments with reduced danger for humans. The Crazyfly's localization relies on external tracking systems, such as OptiTrack²⁹. Another indoor swarm can be build with the FINken-III (Heckert, 2016)³⁰ and its predecessors. They use optical flow, infrared distance, and a tower of four sonar ranging sensors.

3.1.3. Aquatic

In the CoCoRo (Collective Cognitive Robotics) project (Schmickl et al., 2011) a swarm of 41 heterogeneous UUVs has been developed. There are three types of vehicles: A base station USV, an exploration UUV, and UUV for relaying information between the explorers and the base station. Communication is performed with sonar and electric fields. The main applications envisioned are environmental monitoring, measuring water pollution and effects of global warming. The UUV Monsun (Osterloh et al.,

2012) uses two types of communication: an acoustic underwater modem for information exchange and a camera to recognize and follow other swarm members. CORATAM (Control of Aquatic Drones for Maritime Tasks) (Christensen et al., 2015) is a project that develops swarms of USVs. Envisioned applications are environmental monitoring, sea life localization, and sea border patrolling. The platforms are available open-source³¹ and can execute swarm algorithms generated using evolutionary computation (Duarte et al., 2016).

3.1.4. Outer Space

For space exploration, NASA has developed Swarmies³² to collect material samples, such as water, ice, or useful minerals on Mars. This application is referred to as *in-situ* resource utilization (ISRU). Simultaneously, NASA launched a swarmathon³³ to entice students to develop swarm algorithms based on ant foraging. In an experiment 20 Swarmies could travel 42 km of linear distance in 8 h. The same distance was covered by Mars rover Opportunity in 11 years. Another innovative project was accepted by the NASA Innovative Advanced Concepts (NIAC) program. The objective is to increase Mars exploration using a swarm of Marsbees (Kang, 2018). These are robotic flapping wing flyer the size of a bumblebee. They are to self-explore the environment and use the Mars rover Opportunity as base and charging station. During the NIAC funding, a concept for the technical implementation of the flapping flyer using insect-like wings will be proposed.

3.2. Industrial Projects and Products

3.2.1. Terrestrial

One of the biggest challenges in agriculture is the increasing demand for food production (Tilman et al., 2011). SwarmFarm Robotics³⁴ is a company that provides farmers with swarms of agricultural UGVs—the SwarmBot3.0. They work cooperatively, but follow a centrally planned schedule. Before starting the mission, a given field is decomposed into smaller cells which are then allocated to the vehicles (Ball et al., 2015). The swarm's tasks are diverse, but involve planting, applying fertilizer, eliminating weeds and insects, irrigation, and harvesting. A similar project is addressed by the Fendt company with the UGV Xaver³⁵. Each Xaver is a battery-operated planting UGV that operates cloud-controlled and collaborates with the other UGVs in the swarm in terms of a centrally planned seeding plan (Blender et al., 2016). The series production of the robots started with the EU-project MARS (Mobile Agricultural Robot Swarms).

Within the GUARDIANS (Group of Unmanned Assistant Robots Deployed in Aggregative Navigation by Scent) project (Saez-Pons et al., 2010), a swarm of autonomous UGVs has been developed for emergency and rescue applications. This swarm can be used in dangerous situations where toxins are

²³M-TRAN website: <https://www.wevolver.com/wevolver.staff/m-tran>

²⁴ATRON website: <https://www.wevolver.com/wevolver.staff/modular.atron/master/blob/Overview.md>

²⁵For a full list, the reader is referred to https://en.wikipedia.org/wiki/Self-reconfiguring_modular_robot

²⁶Qualcomm Flight Pro website: <https://www.intrinsyc.com/qualcomm-flight-pro-development-kit/>

²⁷DJI M100 website: <https://www.dji.com/at/matrice100/info#specs>

²⁸bitcraze Crazyflies website: <https://www.bitcraze.io/crazyflie-2-1/>

²⁹Optitrack website: <https://optitrack.com/>

³⁰FINken website: <https://www.ci.ovgu.de/SwarmLab/Robots/FINKens.html>

³¹CORATAM website: <http://biomachineslab.com/projects/control-of-aquatic-drones-for-maritime-tasks-coratam/>

³²Swarmies website: <https://www.nasa.gov/content/meet-the-swarmies-robotics-answer-to-bugs>

³³Swarmathon website: <http://nasaswarmathon.com/>

³⁴SwarmFarm Robotics website: <http://www.swarmfarm.com/>

³⁵Xaver website: <https://www.fendt.com/int/xaver>

released that severely impair human senses. The robots warn of toxic chemicals, provide and maintain mobile communication links, infer localization information, and assist in searching. They can generate a formation and navigate while keeping this formation using so-called social potential fields. All tasks can be achieved without central control, and some of the behaviors can be performed without explicit communication between the robots.

Ocado (Telegraph, 2018) is an automated warehouse that uses a swarm of homogeneous cuboid UGVs. Grocery orders are assembled and dispatched using 1,100 collaborative robots. The robots collect ordered crates of food from stacks beneath a huge metal grid organized like a chess grid and deliver them to chutes, where human workers put the customer orders together. As the crates are organized in piles, the robots assist each other to lift out the ones standing in the way. They are operated in bursts to ensure breaks for charging. The robots are controlled centrally by a cloud server. Data is transmitted between the robots and the cloud using cellular technology. The cloud server handles the vast amount of data to coordinate the robots using machine learning approaches. The biggest player for robot swarms in warehouses is Amazon using the Kiva robot system (Brown, 2018). It uses up to 100,000 robots worldwide to move shelf towers in its warehouses. The robots use motion sensors to recognize other robots or shelves in their way. To find their way to the human workers who assemble the customer orders, they use visual tags on the ground of the warehouse to localize and navigate using the A* algorithm. The dispatching is organized centrally and communicated to the robots using WLAN. Robots drive to a charging station automatically in case of low power. A very similar system is used, among others, by the retailer Alibaba (Pickering, 2017).

The SWILT (Swarm Intelligence Layer to Control Autonomous Agents) project (Khatmi et al., 2019)³⁶ takes another approach to modeling UxV. The project focuses on industrial plants in the semi-conductor industry with a high-product mix (about 1,500 different products), where the swarm is made up of lots, machines, and other equipment. The innovation in SWILT is to apply nature-inspired behaviors extracted from swarm intelligence algorithms to the individuals of the swarm instead of pre-calculating global schedules or routing tables. The main difference to traditional methods, such as linear optimization is that feasible, global solutions emerge from local behavior.

3.2.2. Aerial

Swarm missions in the air are typically covered by UAVs, and can be used for different applications. For example, military applications are represented by the OFFSET (OFFensive Swarm-Enabled Tactics) project (Chung, 2017). The main idea of this project is to enhance reconnaissance with UAVs and UGVs inside cities. The applied robots should identify threats using more than 100 different swarm tactics in a game-based environment. The United States Air Force (USAF) works on a swarm of 250 UAVs that is able to perform a 6-h mission for the reconnaissance of eight city blocks. Another swarm of UAVs in military applications

is funded by the Pentagon: the Perdix drone (Mizokami, 2017). A swarm of 103 Perdix UAVs is released from three F/A-18 Super Hornets. The swarm of Perdix drones is able to perform four different missions, including hovering over a target, or forming a 100-m-wide circle in the sky. Their swarms have no central control, no leader, adapt to UAVs entering or exiting the team, are not pre-programmed, make collective-decisions, and can fly in formation. Typical military applications include surveillance missions and targeted assassinations.

The SMAVNET (Swarming Micro Air Vehicle Network) project (Hauert et al., 2009) belongs to the application domain of emergency and rescue. A swarm of autonomous MAVs is developed to deploy and manage an *ad-hoc* WLAN network (Varga et al., 2015). The application is to connect and coordinate rescue teams. Another aim is the exploration of disaster sites, with the goal of localizing victims and directing rescuers toward them. In the project SWARMIX³⁷ they form a swarm of heterogeneous agents (humans, dogs, UAVs) that work cooperatively in a search and rescue mission (Flushing et al., 2014). Similar goals related to search and rescue were achieved in the CPSwarm project³⁸, although the focus was on developing a toolchain for CPS swarm design, modeling, simulation, and deployment.

For agriculture, the SAGA (Swarm Robotics for Agricultural Applications) project³⁹ targets a distributed monitoring and mapping scenario using a swarm of UAVs, which is a novelty in smart farming (Albani et al., 2019). The fitness of the swarm is measured as a trade-off between exploration and weed recognition time. An on-board vision system is used to detect weeds.

Nowadays, entertainment in terms of light shows is a very attractive application for swarms of UAVs. The UAVs are equipped with LED lights and perform different pattern formations creating a free-form display light show, typically accompanied by music. Most providers, like Spaxels⁴⁰, Flyfire⁴¹, Ehang⁴², Intel (Barrett, 2018), and Lucie micro drone⁴³ have central solutions: the “swarm” of up to around 1,000 UAVs is controlled centrally and follows pre-planned patterns.

3.2.3. Aquatic

Environmental monitoring is a common application for swarms in aquatic missions. Platypus (Jeradi et al., 2015) sells autonomous swarm robotic boats, so-called USVs, to measure and monitor water quality. They provide dense maps of defined bodies of water to give a comprehensive picture of the water quality including salinity and oxygen stratification. Different platforms are used depending on the scale and type of the body of water. The boats perform centrally planned collective exploration and interact with the human operator using team oriented planning (Farinelli et al., 2017). Another example is the

³⁷SWARMIX website: <http://www.swarmix.org>

³⁸CPSwarm website: <https://www.cpswarm.eu>

³⁹SAGA website: <http://laral.isc.cnr.it/saga/>

⁴⁰Spaxels website: <http://www.spaxels.at>

⁴¹Flyfire website: <http://senseable.mit.edu/flyfire>

⁴²Ehang website: <http://www.ehang.com>

⁴³Lucie micro drone website: <https://veritystudios.com>

³⁶SWILT website: <https://swilt.aau.at/>

Apium Data Diver⁴⁴. It is a prototype vehicle built for operations in swarms on surface and under water. It is able to dive to a maximum depth of 100 m and has multiple sensors on board including temperature, pressure, and GPS. Possible applications include oceanography, aquaculture, hydrographic survey, and defense. The Data Diver swarm can receive high level commands from a human operator for navigating to a target and forming specific patterns (MacCreedy, 2015). Further autonomous UUVs were developed by Hydromea⁴⁵. Their swarm of UUVs, the so-called Vertex swarm, is able to take water quality measurements at many locations simultaneously down to a depth of 300 m and create 3D data sets with high spatial and temporal resolution much faster than traditional methods. The small size of the UUVs allows for their application, e.g., under ice, in protected areas, underground water caverns, and storage tanks. The UUVs are able to localize in the swarm using acoustic triangulation. Using this positioning information, they form an underwater *ad-hoc* communication network (Schill et al., 2016). The main goal of the SWARMs (Smart Networking Underwater Robots in Cooperation Meshes) project (Real-Arce et al., 2016)⁴⁶ is to make underwater and surface vehicles more accessible and useful for maritime and offshore operations. The aim is to extensively use maritime vehicles instead of professional divers for the typically dangerous offshore operations. SWARMs mainly works on the design and development of a set of software and hardware components to incorporate them into the current generation of maritime vehicles. This helps to improve autonomy, cooperation, robustness, cost-effectiveness, and reliability. Exemplary applications of SWARMs comprise among others: corrosion prevention in offshore installations, monitoring of chemical pollution, and tracking of plumes. A major research focus lies on reliable underwater communication (Rodríguez-Molina et al., 2017) leveraging topology control (Li et al., 2017).

Another application in aquatic environments is military. The software kit CARACaS (Control Architecture for Robotic Agent Command and Sensing), initially developed by NASA for the Mars rover, has been adapted by the Office of Naval Research (ONR). This technology allows autonomous operation of US Navy boats where these USVs interact with each other (Smalley, 2016). These characteristics allow the swarm of USVs to choose their own routes, to intercept enemy vessels as a swarm, and to escort and protect naval assets. To support changes in the swarm, CARACaS allows to re-plan and distribute new task lists. The first successful demo in 2014 has been held on the James River in Virginia, where CARACaS was installed on multiple rigid-hulled inflatable boats. The main application has been demonstrated during the Safe Harbor mission (Hsu, 2016). The project subCULTron (Submarine Cultures Perform Long-Term Robotic Exploration of Unconventional Environmental Niches)⁴⁷ uses the results of the CoCoRo project (described in the previous section Research Platforms) to deploy and test a swarm

of UUVs in the Venetian Lagoon in Italy to evaluate the learning, self-regulation, and self-sustainability of the swarm.

3.2.4. Terrestrial/Aerial/Aquatic

The project ROBORDER (Autonomous Swarm of Heterogeneous Robots for Border Surveillance)⁴⁸ employs an autonomous swarm of heterogeneous robots (UGV, UAV, USV) equipped with multimodal sensors for sea and land border surveillance. Their aim is to detect and identify criminal activities in a vast heterogeneity of threats. The main objective of this project is to incorporate multimodal, statically networked sensors in a swarm of robots.

The project BugWright2⁴⁹ focuses on service and maintenance of large ships. This includes hull cleaning and inspection. Typically, this induces high costs. Therefore, the project's objective lies in the deployment of different cooperating UxV swarms (UGV, UAV, UUV) for a detailed multi-robot visual and acoustic inspection of the hull structure, detecting corrosion patches, and cleaning the surface where necessary.

Sentien Robotics⁵⁰ develops UGVs and UAVs to serve the needs of surveillance, environmental monitoring, infrastructure inspection, and national security. The company develops scalable swarm intelligence software, sensor data processing algorithms, and robot hardware. It furthermore develops a system for automatic launching and recovering of multiple UAVs (Borko, 2016).

3.2.5. Outer Space

Two swarms of satellites are currently in Earth orbit for space exploration: Swarm (Agency, 2004) and Cluster II (Escoubet et al., 2001). Swarm has been launched in 2013 and consists of three identical satellites, each 9 m long, and placed into two different polar orbits: two side by side at an altitude of 450 km and a third at an altitude of 530 km. Their primary task is to study the Earth's magnetic field. Cluster II has been launched in 2000 and consists of four identical, cylindrical (2.9×1.3 m) spacecraft flying in a tetrahedral formation. Their task is to study the impact of the Sun's activity on the Earth's space environment. This is the first time a mission is able to provide three-dimensional data on the influence of the solar wind on the Earth's magnetosphere.

3.3. Discussion

In the above sections we provided an overview of currently available industrial projects and products in the area of swarm robotics as well as research platforms for swarm robotics. This provides researchers and engineers in the domain of swarm robotics a comprehensive overview of current work, existing products, ongoing projects, and available research platforms. From this overview it can be seen that swarm robotic applications are still rare nowadays. Often, the swarm size depends on the number of robots that companies or research agencies have in stock, and are not always selected according to the desired swarm behavior. Although research has been going on for

⁴⁴Apium Data Diver website: <http://apium.com/data-diver/>

⁴⁵Hydromea website: <http://hydromea.com/>

⁴⁶SWARMs website: <http://www.swarms.eu/>

⁴⁷subCULTron website: <http://www.subcultron.eu/>

⁴⁸ROBORDER website: <https://roborder.eu/>

⁴⁹BugWright2 website: <http://dream.georgiatech-metz.fr/?q=node/108>, project start: January 2020.

⁵⁰Sentien Robotics website: <http://sentienrobotics.com/>

several decades, a breakthrough of swarm robotics, especially for industrial applications, has not yet occurred.

This is because there are still several open issues. First of all, the dependability of the robot swarm is a concern. Natural swarms work with the assumption that individual swarm members might fail. In engineered swarms, high reliability and availability is desired in order to provide a working system. Failure of individual swarm members can increase the operational cost and can lead to safety issues, especially for UAVs. Swarm behaviors with their emergent characteristics executed by autonomous robots relying on distributed information cannot give the required guarantees on safety, security, and availability. Therefore, many industrial projects still rely on centralized control, e.g., in the agriculture and warehouse applications of section 3.2.1, or the entertainment applications in section 3.2.2. In these projects, the term swarm is solely used to imply the high number of agents. The implementations neglect the principal idea of swarm robotics which is distributed decision making that leads to self-organized behavior. Even though the robots have the ability to identify their environment, gather data locally, and communicate this data with the rest of the swarm, they rely on a central unit. This central unit either pre-defines the behavior of each individual robot or, in more dynamic scenarios, processes the information received from the robots to control their behavior. The issue of safety and security is addressed in research projects like SAGA (Albani et al., 2019), SWILT (Khatmi et al., 2019), and CPSwarm (Bagnato et al., 2017). In these projects additional routines are defined that allow to stop individual swarm members or the entire self-organized swarm to prevent harm to humans or other machines. This is achieved by running multiple processes in parallel to react to certain events. For example, sensor data or emergency stop signals by a human operator can be processed in parallel to the behavior algorithm in order to immediately stop any movement of the swarm members. These emergency processes provide a deterministic behavior as opposed to the normal run time behavior. The safety issues are less critical in aquatic environments. The USVs and UUVs described in section 3.2.3 are able to perform fully autonomous exploration and localization. Compared to terrestrial or aerial environments, the possibility of harming humans is relatively low. Though, harm to the environment or animals is not taken into account.

Another issue is the communication within the swarm and between the swarm and a central unit commanding and controlling station. For a swarm to work fully autonomously, it should provide its own means of communication. This is achieved by *ad-hoc* WLAN networks typically employed in emergency and rescue scenarios (see section 3.2.2). Such communication networks have a limited range and are less stable since they can break down when individual robots fail or move out of range. An infrastructure-based network, such as a cellular network, can provide more stable communication but it requires the installation of base stations. While this is typically available for terrestrial or areal environments, this does not work for space or aquatic missions. Especially in aquatic missions, commonly used radio communication does not work due to the high attenuation of water (Rodríguez-Molina

et al., 2017). Therefore, robot swarms in such environments must use less researched technologies, such as sound or electric field communication that have lower throughput (see section 3.1.3). Besides the technological limitations, an important issue when communicating is security. This is of special interest in military applications (section 3.2.2). First, information exchanged between robots could be sensitive and should not be disclosed to hostile parties. Second, the behavior of a swarm could be influenced based on the information the swarm members receive. This means that the behavior of a swarm could be influenced by altering the messages exchanged between robots or injecting false information in the swarm. Therefore, the communication in the swarm needs to be encrypted and swarm members must authenticate against each other in order to provide reliable behavior. This is especially important when a central station sends commands to control the swarm.

Compared to the industrial projects and products, the swarm robotic research shows swarm behaviors close to the natural swarm inspiration that relies on distributed control. Generally, the research platforms' hardware design technologically follows the inspiration from nature as it is small and cheap. They typically use simple and reduced microcontrollers, such as the Arduino. In terms of other hardware characteristics, they are quite diverse. Especially the UGVs show different types of locomotion, e.g., the Kilobots (Rubenstein et al., 2014a) use vibration, the Spiderino (Jdeed et al., 2017) is a six-legged robot, and the AMiR (Arvin et al., 2009) and its successors use wheels. Furthermore, they show different types of communication, e.g., the Kilobots (Rubenstein et al., 2014a) use reflecting light, the I-Swarm (Seyfried et al., 2004) uses infrared, and the Khepera IV (Soares et al., 2016) uses WLAN. Additionally, different types of power sources are used, e.g., batteries (which are most common) and solar cells for the I-Swarm (Seyfried et al., 2004). Besides the diverse actuators, they offer a number of different sensors that can be used for different swarm behaviors to interact with the environment. As outlined by the authors, these platforms are mainly used for research. So, most circuit designs are published open-source online. Partially, these platforms are specifically dedicated for educational use. For example, the Thymio (Riedo et al., 2013) is offered online with a lot of diverse educational material⁵¹ and the Spiderino (Jdeed et al., 2017) is offered as part of workshops to pupils in the eduLab (Pitschmann, 2019).

With all these available components on the swarm robotic research platforms, it is possible to test and evaluate swarm intelligence algorithms. This is not restricted to the software side but includes also the hardware needed to interact with the environment. This allows to draw first conclusions on the emergence of swarm behaviors and required hardware in laboratory environments. Nevertheless, the step from these research platforms to a significant number of applications or industrial products has not been achieved yet.

Nevertheless, there is already a paradigm shift in the industry. Several companies in different domains envision self-organizing solutions to the increasingly high complexity of their production

⁵¹Thymio's Official Website: <https://www.thymio.org/de/>

plants and activities in dynamic environments. Such industry projects are, e.g., SWILT, ROBORDER, and BugWright2. This establishes an opportunity for researchers in swarm robotics and swarm intelligence to get their ideas out of the lab and into a specific application. The main issue here is to map the swarm members onto the components of an existing system and to engineer their behavior with respect to the target application. In the SWILT project, swarm members are machines, products, or lots. They must fulfill certain conditions, e.g., having computational resources to exhibit local intelligence. Nevertheless, real-world scenarios typically go beyond swarm robotics and fall into the area of swarms of cyber-physical systems. The most promising applications are in domains where it is impossible or too dangerous for humans to enter, the environment is unknown, or the real-time requirements are too restrictive to pre-compute globally optimal solutions. Specific examples could be the exploration of the deep sea, space, or celestial bodies, environmental monitoring, smart traffic concepts, or nano medicine. These visionary applications are further detailed by Schrantz et al. (2020). To implement solutions in such environments, new methods, technologies, and visions are required in order to shift swarm intelligence from swarm robotic research platforms to swarm robotic products.

4. CONCLUSION

Research on swarm algorithms is a relatively young topic. Despite the large number of swarm algorithms, the transition to industry and industrial production, not to mention daily use, has not been made successfully. Nevertheless, several steps toward swarm applications have already been taken. The main objective of this paper is to motivate future research and engineering activities by providing a comprehensive list of existing platforms, projects and products as a starting point for applied research in swarm robotics.

This paper classifies basic swarm behaviors and presents a comprehensive overview of current research platforms and industrial applications. While this demonstrates the possibility of integrating basic swarm behaviors in current applications, it also shows that many applications of swarm robotics cannot fully exploit the advantages offered by distributed swarm architectures due to systems with only few agents or central control. Swarm algorithms build upon self-organized swarm behaviors, e.g., observed in natural swarm systems, such as insect colonies or

flocks of birds that are able to handle extremely diverse and dynamic environments. The same holds for robot swarms. They are meant to operate in the physical world, which typically faces continual dynamic changes and must cope with events and external conditions that are hard to predict or model. Besides huge potential for applications in areas like logistics, agriculture, and inspection, one suitable working environment for swarms are places that are unsuitable for humans, including places that are hard to reach, dangerous, or dirty. Applications in these environments could help to better observe, understand and exploit the advantages of swarm behaviors: adaptability, robustness, and scalability.

In addition to industrial applications, we have also surveyed different research hardware platforms dedicated to swarm robotic experiments. On the one hand, this overview allows to choose an appropriate research platform for implementing and testing swarm algorithms in laboratory environments. On the other hand, it shows that there is a huge potential in research to transform these platforms from pure prototyping platforms to productive, industrial robotic systems that are able to perform in the real world. This might require to shift from the current simplified robot models and controls to a trade-off between simplicity of design and capability of solving complex tasks in a reliable way, e.g., from reduced resource consumption to a more intensive usage of sensor data and information sharing.

AUTHOR CONTRIBUTIONS

MSc: main part of research in industrial products and projects and research platforms including discussion, extension of new swarm behaviors, and overall organization of contribution and format. MSe: main part of swarm behaviors, support in research on products and projects and discussion. MU: research on swarm behaviors and adaption with new behaviors. WE: support in the discussion, abstract, and conclusion.

ACKNOWLEDGMENTS

We thank Arthur Pitman and Andreas Kercek for proofreading the text. The research leading to these results has received funding from the European Union Horizon 2020 research and innovation program under grant agreement No. 731946, CPSSwarm project.

REFERENCES

- Agency, E. S. (2004). *Swarm—The Earth's Magnetic Field and Environment Explorers Technical and Programmatic Annex*. Technical report, European Space Agency.
- Albani, D., Manoni, T., Arik, A., Nardi, D., and Trianni, V. (2019). "Field coverage for weed mapping: toward experiments with a UAV swarm," in *Proceedings of the 11th EAI International Conference on Bio-Inspired Information and Communications Technologies* (Cham), 1–16. doi: 10.1007/978-3-030-24202-2_10
- Alkilabi, M. H. M., Narayan, A., and Tuci, E. (2017). Cooperative object transport with a swarm of e-puck robots: robustness and scalability of evolved collective strategies. *Swarm Intell.* 11, 185–209. doi: 10.1007/s11721-017-0135-8
- Arbuckle, D. J., and Requicha, A. A. G. (2010). Self-assembly and self-repair of arbitrary shapes by a swarm of reactive robots: algorithms and simulations. *Auton. Robots* 28, 197–211. doi: 10.1007/s10514-009-9162-7
- Arvin, F., Espinosa, J., Bird, B., West, A., Watson, S., and Lennox, B. (2019). Mona: an affordable open-source mobile robot for education and research. *J. Intell. Robot. Syst.* 94, 761–775. doi: 10.1007/s10846-018-0866-9
- Arvin, F., Murray, J., Zhang, C., and Yue, S. (2014). Colias: an autonomous micro robot for swarm robotic applications. *Int. J. Adv. Robot. Syst.* 11:113. doi: 10.5772/58730

- Arvin, F., Samsudin, K., and Ramli, A. R. (2009). Development of a miniature robot for swarm robotic application. *Int. J. Comput. Electr. Eng.* 1:436. doi: 10.7763/IJCEE.2009.V1.67
- Arvin, F., Samsudin, K., Ramli, A. R., and Bekravi, M. (2011). Imitation of honeybee aggregation with collective behavior of swarm robots. *Int. J. Comput. Intell. Syst.* 4, 739–748. doi: 10.1080/18756891.2011.9727825
- Arvin, F., Watson, S., Turgut, A. E., Espinosa, J., Krajník, T., and Lennox, B. (2017). Perpetual robot swarm: long-term autonomy of mobile robots using on-the-fly inductive charging. *J. Intell. Robot. Syst.* 92, 1–18. doi: 10.1007/s10846-017-0673-8
- Bagnato, A., Biró, R. K., Bonino, D., Pastrone, C., Elmenreich, W., Reiners, R., et al. (2017). “Designing swarms of cyber-physical systems: the H2020 CPSwarm project: invited paper,” in *Proceedings of the Computing Frontiers Conference* (New York, NY: Association for Computing Machinery), 305–312. doi: 10.1145/3075564.3077628
- Ball, D., Ross, P., English, A., Patten, T., Upcroft, B., Fitch, R., et al. (2015). *Robotics for Sustainable Broad-Acre Agriculture*. Springer, 439–453. doi: 10.1007/978-3-319-07488-7_30
- Barciś, A., Barciś, M., and Bettstetter, C. (2019). “Robots that sync and swarm: a proof of concept in ROS 2,” in *Proceedings of the International Symposium on Multi-Robot and Multi-Agent Systems* (New Brunswick, NJ: IEEE), 98–104.
- Barrett, B. (2018). *Intel Lights Up the Night with 500 ‘Shooting Star’ Drones*. Available online at: <https://www.wired.com/story/more-inclusive-fourth-of-july-drone-show/> (accessed August 2, 2019).
- Bezzo, N., Cruz, P. J., Sorrentino, F., and Fierro, R. (2014). Decentralized identification and control of networks of coupled mobile platforms through adaptive synchronization of chaos. *Phys D Nonlin Phenom.* 267, 94–103. doi: 10.1016/j.physd.2013.08.012
- Binitha, S., and Sathya, S. S. (2012). A survey of bio inspired optimization algorithms. *Int. J. Soft Comput. Eng.* 2, 137–151.
- Blender, T., Buchner, T., Fernandez, B., Pichlmaier, B., and Schlegel, C. (2016). “Managing a mobile agricultural robot swarm for a seeding task,” in *Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society* (Florence), 6879–6886. doi: 10.1109/IECON.2016.7793638
- Blum, C., and Li, X. (2008). “Swarm intelligence in optimization,” in *Swarm Intelligence: Introduction and Applications*, eds C. Blum and D. Merkle (Berlin; Heidelberg: Springer), 43–85. doi: 10.1007/978-3-540-74089-6
- Bonabeau, E., Dorigo, M., and Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
- Borko, B. (2016). *System for Automatic Takeoff and Landing by Interception of Small UAVs*. US Patent 9,505,493.
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* 7, 1–41. doi: 10.1007/s11721-012-0075-2
- Brandt, D., Christensen, D. J., and Lund, H. H. (2007). “ATRON robots: versatility from self-reconfigurable modules,” in *Proceedings of the International Conference on Mechatronics and Automation* (Harbin), 26–32. doi: 10.1109/ICMA.2007.4303511
- Brown, A. (2018). *Rise of the Machines? Amazon’s Army of More Than 100,000 Warehouse Robots Still Can’t Replace Humans Because They Lack ‘Common Sense’*. Available online at: <http://www.dailymail.co.uk/sciencetech/article-5808319/Amazon-100-000-warehouse-robots-company-insists-replace-humans.html> (accessed August 2, 2019).
- Buck, J. (1988). Synchronous rhythmic flashing of fireflies. II. *Q. Rev. Biol.* 63, 265–289. doi: 10.1086/415929
- Cai, G., Dias, J., and Seneviratne, L. (2014). A survey of small-scale unmanned aerial vehicles: recent advances and future development trends. *Unmann. Syst.* 2, 175–199. doi: 10.1142/S2301385014300017
- Camazine, S., Franks, N. R., Sneyd, J., Bonabeau, E., Deneubourg, J.-L., and Theraulaz, G. (2001). *Self-Organization in Biological Systems*. Princeton University Press.
- Caprari, G., Balmer, P., Piguet, R., and Siegwart, R. (1998). “The autonomous micro robot ‘Alice’: a platform for scientific and commercial applications,” in *Proceedings of the International Symposium on Micromechatronics and Human Science. Creation of New Industry* (Cat. No. 98TH8388) (Nagoya: IEEE), 231–235.
- Castano, A., Behar, A., and Will, P. M. (2002). The conro modules for reconfigurable robots. *IEEE/ASME Trans. Mechatr.* 7, 403–409. doi: 10.1109/TMECH.2002.806233
- Cheng, J., Cheng, W., and Nagpal, R. (2005). “Robust and self-repairing formation control for swarms of mobile agents,” in *Proceedings of the 20th National Conference on Artificial Intelligence*, Vol. 1.
- Christensen, A., O’Grady, R., and Dorigo, M. (2009). From fireflies to fault-tolerant swarms of robots. *IEEE Trans. Evol. Comput.* 13, 754–766. doi: 10.1109/TEVC.2009.2017516
- Christensen, A. L., Oliveira, S., Postolache, O., de Oliveira, M. J., Sargento, S., Santana, P., et al. (2015). “Design of communication and control for swarms of aquatic surface drones,” in *Proceedings of the International Conference on Agents and Artificial Intelligence*, 548–555. doi: 10.5220/0005281705480555
- Chung, T. H. (2017). *OFFensive Swarm-Enabled Tactics (OFFSET)*. Available online at: https://www.darpa.mil/attachments/OFFSET_ProposersDay.pdf (accessed April 2, 2019).
- Dai, Y.-S., Hinchey, M., Madhusoodan, M., Rash, J., and Zou, X. (2006). “A prototype model for self-healing and self-reproduction in swarm robotics system,” in *Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing* (Indianapolis, IN: IEEE), 3–10. doi: 10.1109/DASC.2006.10
- Dhanaraj, N., Hewitt, N., Edmonds-Estes, C., Jarman, R., Seo, J., Gunner, H., et al. (2019). “Adaptable platform for interactive swarm robotics (apis): a human-swarm interaction research testbed,” in *Proceedings of the 19th International Conference on Advanced Robotics* (Belo Horizonte), 720–726. doi: 10.1109/ICAR46387.2019.8981628
- Dorigo, M., and Birattari, M. (2007). Swarm intelligence. *Scholarpedia* 2:1462. doi: 10.4249/scholarpedia.1462
- Dorigo, M., Floreano, D., Gambardella, L. M., Mondada, F., Nolfi, S., Baaboura, T., et al. (2013). Swarmanoid: a novel concept for the study of heterogeneous robotic swarms. *IEEE Robot. Autom. Mag.* 20, 60–71. doi: 10.1109/MRA.2013.2252996
- Duarte, M., Costa, V., Gomes, J., Rodrigues, T., Silva, F., Oliveira, S. M., et al. (2016). Evolution of collective behaviors for a real swarm of aquatic surface robots. *PLoS ONE* 11:e151834. doi: 10.1371/journal.pone.0151834
- Duff, D., Yim, M., and Roufas, K. (2001). “Evolution of polybot: a modular reconfigurable robot,” in *Proceedings of the Harmonic Drive Intelligent Symposium* (Nagano).
- Elisa-3 (2014). *Elisa-3 Robots–38 Units With Obstacle Avoidance*. Available online at: <https://www.youtube.com/watch?v=WDxflFhpm1g> (accessed January 30, 2020).
- Escoubet, C., Fehringier, M., and Goldstein, M. (2001). Introduction—the cluster mission. *Proc. Ann. Geophys.* 19, 1197–1200. doi: 10.5194/angeo-19-1197-2001
- Farinelli, A., Raeissi, M. M., Marchi, N., Brooks, N., and Scerri, P. (2017). Interacting with team oriented plans in multi-robot systems. *Auton. Agents Multi Agent Syst.* 31, 332–361. doi: 10.1007/s10458-016-9344-6
- Floreano, D., and Mattiussi, C. (2008). *Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies*. MIT Press.
- Flushing, E. F., Gambardella, L. M., and Di Caro, G. A. (2014). “A mathematical programming approach to collaborative missions with heterogeneous teams,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Chicago, IL), 396–403. doi: 10.1109/IROS.2014.6942590
- Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (2000). Probabilistic approach to collaborative multi-robot localization. *Auton. Robots* 8, 325–344. doi: 10.1023/A:1008937911390
- Garnier, S., Gautrais, J., and Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm Intell.* 1, 3–31. doi: 10.1007/s11721-007-0004-y
- Garnier, S., Jost, C., Gautrais, J., Asadpour, M., Caprari, G., Jeanson, R., et al. (2008). The embodiment of cockroach aggregation behavior in a group of micro-robots. *Artif. Life* 14, 387–408. doi: 10.1162/artl.2008.14.4.14400
- Giusti, A., Nagi, J., Gambardella, L., and Di Caro, G. A. (2012). “Cooperative sensing and recognition by a swarm of mobile robots,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura: IEEE), 551–558. doi: 10.1109/IROS.2012.6385982
- Goss, S., Aron, S., Deneubourg, J. L., and Pasteels, J. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften* 76, 597–581. doi: 10.1007/BF00462870
- Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2006). Autonomous self-assembly in swarm-bots. *IEEE Trans. Robot.* 22, 1115–1130. doi: 10.1109/TRO.2006.882919

- Gross, R., and Dorigo, M. (2008). Self-assembly at the macroscopic scale. *Proc. IEEE* 96, 1490–1508. doi: 10.1109/JPROC.2008.927352
- Hamann, H., and Schmickl, T. (2012). Modelling the swarm: analysing biological and engineered swarm systems. *Math. Comput. Modell. Dyn. Syst.* 18, 1–12. doi: 10.1080/13873954.2011.601426
- Hartbauer, M., Kratzer, S., Steiner, K., and Römer, H. (2005). Mechanisms for synchrony and alternation in song interactions of the bushcricket *Mecopoda elongata* (Tettigoniidae: Orthoptera). *J. Compar. Physiol. A* 191, 175–188. doi: 10.1007/s00359-004-0586-4
- Hartbauer, M., and Römer, H. (2006). “Decentralized microrobot swarm communication via coupled oscillators,” in *Proceedings of the IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006, Vol. 2006 (IEEE)*, 585–590.
- Hassanien, A. E., and Alamry, E. (2015). *Swarm Intelligence: Principles, Advances, and Applications*. CRC Press.
- Hauert, S., Zufferey, J.-C., and Floreano, D. (2009). Evolved swarming without positioning information: an application in aerial communication relay. *Auton. Robots* 26, 21–32. doi: 10.1007/s10514-008-9104-9
- Heckert, A. (2016). *Entwicklung eines dynamischen Modells und Parameterschätzung für den Finken 3 Quadkopter* (Bachelor's thesis), Fakultät für Elektrotechnik und Informationstechnik, Otto-von-Guericke-Universität Magdeburg.
- Héder, M. (2017). From NASA to EU: the evolution of the TRL scale in public sector innovation. *Innov. J.* 22:3.
- Hölldobler, B., and Wilson, E. O. (1994). *Journey to the Ants: A Story of Scientific Exploration*. Harvard University Press.
- Howard, A., Matarić, M. J., and Sukhatme, G. S. (2003). “Localization for mobile robot teams: a distributed MLE approach,” in *Experimental Robotics VIII* (Berlin; Heidelberg: Springer Berlin Heidelberg), 146–155. doi: 10.1007/3-540-36268-1_12
- Hsu, J. (2016). *U.S. Navy's Drone Boat Swarm Practices Harbor Defense*. Available online at: <https://spectrum.ieee.org/automaton/robotics/military-robots/navy-drone-boat-swarm-practices-harbor-defense> (accessed September 27, 2019).
- Igoshin, O. A., Mogilner, A., Welch, R. D., Kaiser, D., and Oster, G. (2001). Pattern formation and traveling waves in myxobacteria: theory and modeling. *Proc. Natl. Acad. Sci. U.S.A.* 98, 14913–14918. doi: 10.1073/pnas.221579598
- Jdeed, M., Zhevzhky, S., Steinkellner, F., and Elmenreich, W. (2017). “Spiderino: a low-cost robot for swarm research and educational purposes,” in *Proceedings of the 13th Workshop on Intelligent Solutions in Embedded Systems* (Hamburg), 35–39. doi: 10.1109/WISES.2017.7986929
- Jeradi, A., Raeissi, M. M., Farinelli, A., Brooks, N., and Scerri, P. (2015). “Focused exploration for cooperative robotic watercraft,” in *Proceedings of the International Workshop on Artificial Intelligence and Robotics*, 83–93.
- Jones, S., Studley, M., Hauert, S., and Winfield, A. F. T. (2018). A two teraflop swarm. *Front. Robot. AI* 5:11. doi: 10.3389/frobt.2018.00011
- Kang, C.-K. (2018). *Marsbee-Swarm of Flapping Wing Flyers for Enhanced Mars Exploration*. Available online at: https://www.nasa.gov/directorates/spacetechniac/2018_Phase_I_Phase_II/Marsbee_Swarm_of_Flapping_Wing_Flyers_for_Enhanced_Mars_Exploration (accessed September 5, 2019).
- Kernbach, S., Häbe, D., Kernbach, O., Thenius, R., Radspieler, G., Kimura, T., et al. (2013). Adaptive collective decision-making in limited robot swarms without communication. *Int. J. Robot. Res.* 32, 35–55. doi: 10.1177/0278364912468636
- Kernbach, S., Meister, E., Schlachter, F., Jebens, K., Szymanski, M., Liedke, J., et al. (2008). “Symbiotic robot organisms: REPLICATOR and SYMBRION projects,” in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems* (Gaithersburg, MD), 62–69. doi: 10.1145/1774674.1774685
- Kettler, A. (2012). *Cleaning Up With a Swarm of Robots*. Available online at: https://www.youtube.com/watch?v=FqMP_AIkDj8 (accessed February 5, 2020).
- Kettler, A., Szymanski, M., and Wörn, H. (2012). “The wanda robot and its development system for swarm algorithms,” in *Advances in Autonomous Mini Robots*, eds R. Ulrich, J. Sitte, and F. Werner (Berlin; Heidelberg: Springer), 133–146. doi: 10.1007/978-3-642-27482-4_15
- Khatmi, E., Elmenreich, W., Wogatai, K., Schranz, M., Umlauf, M., Laure, W., et al. (2019). “Swarm intelligence layer to control autonomous agents (SWILT),” in *Proceedings of the STAF 2019 1st Research Project Showcase Workshop* (Eindhoven), 91–96.
- King, D., and Breedon, P. (2010). Towards cooperative robotic swarm recognition: object classification and validity. *Key Eng. Mater.* 450, 320–324. doi: 10.4028/www.scientific.net/KEM.450.320
- Klingner, J., Kanakia, A., Farrow, N., Reishus, D., and Correll, N. (2014). “A stick-slip omnidirectional powertrain for low-cost swarm robotics: mechanism, calibration, and control,” in *IEEE International Conference on Intelligent Robots and Systems* (Chicago, IL), 846–851. doi: 10.1109/IROS.2014.6942658
- Kornienko, S., Kornienko, O., Constantinescu, C., Pradier, M., and Levi, P. (2005). “Cognitive micro-agents: individual and collective perception in microrobotic swarm,” in *Proceedings of the Workshop on Agents in Real-Time and Dynamic Environments* (Edinburgh), 33–42.
- Krause, J., Cordeiro, J., Parpinelli, R. S., and Lopes, H. S. (2013). “A survey of swarm algorithms applied to discrete optimization,” in *Swarm Intelligence and Bio-Inspired Computation*, eds X. S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, and M. Karamanoglu (Elsevier), 169–191. doi: 10.1016/B978-0-12-405163-8.00007-7
- Kurazume, R., and Hirose, S. (2000). An experimental study of a cooperative positioning system. *Auton. Robots* 8, 43–52. doi: 10.1023/A:1008988801987
- Kurokawa, H., Kamimura, A., Yoshida, E., Tomita, K., Kokaji, S., and Murata, S. (2003). “M-TRAN II: metamorphosis from a four-legged walker to a caterpillar,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3, 2454–2459.
- Kurokawa, H., Tomita, K., Kamimura, A., Kokaji, S., Hasuo, T., and Murata, S. (2008). Distributed self-reconfiguration of M-TRAN III modular robotic system. *Int. J. Robot. Res.* 27, 373–386. doi: 10.1177/0278364907085560
- Le Goc, M., Kim, L. H., Parsaei, A., Fekete, J.-D., Dragicevic, P., and Follmer, S. (2016). “Zoooids: building blocks for swarm user interfaces,” in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo), 97–109. doi: 10.1145/2984511.2984547
- Li, N., Cürüklü, B., Bastos, J., Sucasas, V., Fernandez, J. A. S., and Rodriguez, J. (2017). A probabilistic and highly efficient topology control algorithm for underwater cooperating AUV networks. *Sensors* 17:E1022. doi: 10.3390/s17051022
- Lipson, H., and Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. *Nature* 406, 974–978. doi: 10.1038/35023115
- MacCready, T. (2015). “Multiscale vorticity from a swarm of drifters,” in *Proceedings of the IEEE/OES Eleventh Current, Waves and Turbulence Measurement* (St. Petersburg, FL), 1–6. doi: 10.1109/CWTM.2015.7098096
- Mazdin, P., and Rinner, B. (2019). *Coordination of Mobile Agents for Simultaneous Coverage. Volume 11873 of Lecture Notes in Computer Science*. Cham: Springer International Publishing, 170–185. doi: 10.1007/978-3-030-33792-6_11
- McLurkin, J., Lynch, A. J., Rixner, S., Barr, T. W., Chou, A., Foster, K., et al. (2013). “A low-cost multi-robot system for research, teaching, and outreach,” in *Distributed Autonomous Robotic Systems*, eds M. Alcherio, M. Francesco, C. Nikolaus, M. Grégory, E. Magnus, H. M. Ani, P. E. Lynne, and S. Kasper (Berlin; Heidelberg: Springer), 597–609. doi: 10.1007/978-3-642-32723-0_43
- Mermoud, G., and Evans, W. C. (2010). *Aggregation-mediated Collective Perception and Action in a Group of Miniature Robots*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
- Mizokami, K. (2017). *The Pentagon's Autonomous Swarming Drones Are the Most Unsettling Thing You'll See Today*. Available online at: <https://www.popularmechanics.com/military/aviation/a24675/pentagon-autonomous-swarming-drones/> (accessed April 2, 2018).
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Ciani, C., Klapotcz, A., et al. (2009). “The e-puck, a robot designed for education in engineering,” in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, 59–65.
- Mondada, F., Pettinaro, G. C., Kwee, I. W., Guignard, A., Gambardella, L. M., Floreano, D., et al. (2002). *SWARM-BOT: A Swarm of Autonomous Mobile Robots With Self-assembling Capabilities*. Technical report, ETH-Zürich.
- Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., and Kokaji, S. (2002). M-TRAN: self-reconfigurable modular robotic system. *IEEE/ASME Trans. Mechatr.* 7, 431–441. doi: 10.1109/TMECH.2002.806220
- Néda, Z., Ravasz, E., Brechet, Y., Vicsek, T., and Barabási, A.-L. (2000). The sound of many hands clapping. *Nature* 403, 849–850. doi: 10.1038/35002660
- O’Keeffe, K. P., Hong, H., and Strogatz, S. H. (2017). Oscillators that sync and swarm. *Nat. Commun.* 8:1504. doi: 10.1038/s41467-017-01190-3
- Olfati-Saber, R., and Jalalkamali, P. (2012). Coupled distributed estimation and control for mobile sensor networks. *IEEE Trans. Autom. Control* 57, 2609–2614. doi: 10.1109/TAC.2012.2190184

- Osterloh, C., Pionteck, T., and Maehle, E. (2012). "MONSUN II: a small and inexpensive AUV for underwater swarms," in *Proceedings of the 7th German Conference on Robotics* (Munich), 1–6.
- Oung, R. (2013). *The Distributed Flight Array: Summary*. Available online at: <https://www.youtube.com/watch?v=fcradVE9uts> (accessed January 30, 2020).
- Oung, R., and D'Andrea, R. (2011). The distributed flight array. *Mechatronics* 21, 908–917. doi: 10.1016/j.mechatronics.2010.08.003
- Parpinelli, R. S., and Lopes, H. S. (2011). New inspirations in swarm intelligence: a survey. *Int. J. Bioinsp. Comput.* 3, 1–16. doi: 10.1504/IJBIC.2011.038700
- Perez-Diaz, F., Zillmer, R., and Groß, R. (2015). "Firefly-inspired synchronization in swarms of mobile agents," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems* (Istanbul), 279–286.
- Peskin, C. S. (1975). *Mathematical Aspects of Heart Physiology*. Courant Inst. Math.
- Petersen, K. H., Nagpal, R., and Werfel, J. K. (2011). "Termes: an autonomous robotic system for three-dimensional collective construction," in *Robotics: Science and Systems VII*. doi: 10.15607/RSS.2011.VII.035
- Pickem, D., Glotfelter, P., Wang, L., Mote, M., Ames, A. D., Feron, E., et al. (2017). *The Robotarium: A Remotely Accessible Swarm Robotics Research Testbed*. Available online at: <https://robhub.org/the-robotarium-a-remotely-accessible-swarm-robotics-research-testbed/> (accessed January 30, 2020).
- Pickem, D., Lee, M., and Egerstedt, M. (2015). "The GRITSBot in its natural habitat—a multi-robot testbed," in *Proceedings of the IEEE International Conference on Robotics and Automation* (Seattle, WA), 4062–4067. doi: 10.1109/ICRA.2015.7139767
- Pickering, J. (2017). *Take a Look Inside Alibaba's Smart Warehouse Where Robots do 70% of the Work*. Available online at: <https://www.businessinsider.com/inside-alibaba-smart-warehouse-robots-70-per-cent-work-technology-logistics-2017-9?IR=T> (accessed August 2, 2019).
- Pinciroli, C., and Beltrame, G. (2016). Swarm-oriented programming of distributed robot networks. *Computer* 49, 32–41. doi: 10.1109/MC.2016.376
- Pitschmann, D. (2019). *Spiderino—Robotikplattform Unterstützt im Schwarm MINT Ausbildung*. Available online at: <https://www.lakeside-scitec.com/news/news-articles/spiderino-robotikplattform-unterstuetzt-im-schwarm-mint-ausbildung/> (accessed September 30, 2019).
- Preiss, J. A., Hönig, W., Sukhatme, G. S., and Ayanian, N. (2017). "Crazyswarm: a large nano-quadcopter swarm," in *Proceedings of the International Conference on Robotics and Automation*, 3299–3304. Software available online at: <https://github.com/USC-ACTLab/crazyswarm>
- Ratnieks, F. L. W., and Anderson, C. (1999). Task partitioning in insect societies. *Insect. Soc.* 46, 95–108. doi: 10.1007/s000400050119
- Real-Arce, D. A., Morales, T., Barrera, C., Hernández, J., and Llinás, O. (2016). "Smart and networking underwater robots in cooperation meshes: the swarms ECSEL: H2020 project," in *Proceedings of the International Workshop on Marine Technology*, 19.
- Rekleitis, I., Dudek, G., and Milios, E. (2001). Multi-robot collaboration for robust exploration. *Ann. Math. Artif. Intell.* 31, 7–40. doi: 10.1023/A:1016636024246
- Riedo, F., Chevalier, M., Magnenat, S., and Mondada, F. (2013). "Thymio II, a robot that grows wiser with children," in *Proceedings of the IEEE Workshop on Advanced Robotics and Its Social Impacts* (Tokyo), 187–193. doi: 10.1109/ARSO.2013.6705527
- Roberts, J. F., Stirling, T. S., Zufferey, J., and Floreano, D. (2007). "Quadrotor using minimal sensing for autonomous indoor flight," in *Proceedings of the European Micro Air Vehicle Conference and Flight Competition* (Toulouse).
- Rodríguez-Molina, J., Bilbao, S., Martínez, B., Frasheri, M., and Cürüklü, B. (2017). An optimized, data distribution service-based solution for reliable data exchange among autonomous underwater vehicles. *Sensors* 17:E1802. doi: 10.3390/s17081802
- Rossi, F., Bandyopadhyay, S., Wolf, M., and Pavone, M. (2018). Review of multi-agent algorithms for collective behavior: a structural taxonomy. *IFAC Pap Online* 51, 112–117. doi: 10.1016/j.ifacol.2018.07.097
- Rubenstein, M., Ahler, C., Hoff, N., Cabrera, A., and Nagpal, R. (2014a). Kilobot: a low cost robot with scalable operations designed for collective behaviors. *Robot. Auton. Syst.* 62, 966–975. doi: 10.1016/j.robot.2013.08.006
- Rubenstein, M., Cornejo, A., and Nagpal, R. (2014b). Programmable self-assembly in a thousand-robot swarm. *Science* 345, 795–799. doi: 10.1126/science.1254295
- Rubenstein, M., and Shen, W. M. (2008). "A scalable and distributed model for self-organization and self-healing," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems* 2, 1157–1160.
- Saez-Pons, J., Alboul, L., Penders, J., and Nomdedeu, L. (2010). Multi-robot team formation control in the GUARDIANS project. *Ind. Robot Int. J.* 37, 372–383. doi: 10.1108/01439911011044831
- Schill, F., Bahr, A., and Martinoli, A. (2016). "Vertex: a new distributed underwater robotic platform for environmental monitoring," in *International Symposium on Distributed Autonomous Robotic Systems* (Cham: Springer International Publishing), 1–14.
- Schmickl, T. (2015). *Collective Cognitive Robots—The Year of CoCoRo*. Available online at: <http://zool33.uni-graz.at/artlife/cocoro> (accessed January 30, 2020).
- Schmickl, T., Möslinger, C., and Crailsheim, K. (2007). *Collective Perception in a Robot Swarm*. Berlin; Heidelberg: Springer.
- Schmickl, T., Thenius, R., Moslinger, C., Timmis, J., Tyrrell, A., Read, M., et al. (2011). "CoCoRo—the self-aware underwater swarm," in *Proceedings of the 5th IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops* (Ann Arbor, MI), 120–126. doi: 10.1109/SASOW.2011.11
- Schranz, M., Di Caro, G. A., Schmickl, T., Elmenreich, W., Arvin, F., Şekercioglu, A., et al. (2020). *Swarm Intelligence and Cyber-Physical Systems: Concepts, Challenges and Future Trends*. Elsevier.
- Schranz, M., and Rinner, B. (2015). "Resource-aware state estimation in visual sensor networks with dynamic clustering," in *Proceedings of the 4th International Conference on Sensor Networks—Volume 1: SENSORNETS* (Angers: INSTICC, SciTePress), 15–24.
- Seyfried, J., Szymanski, M., Bender, N., Estaña, R., Thiel, M., and Wörn, H. (2004). "The I-SWARM project: intelligent small world autonomous robots for micro-manipulation," in *Proceedings of the International Workshop on Swarm Robotics* (Berlin; Heidelberg: Springer), 70–83. doi: 10.1007/978-3-540-30552-1_7
- Sharkey, A. J. (2007). Swarm robotics and minimalism. *Connect. Sci.* 19, 245–260. doi: 10.1080/09540090701584970
- Shimizu, H., Sawada, Y., and Sugiyama, T. (1993). Minimum tissue size required for hydra regeneration. *Dev. Biol.* 155, 287–296. doi: 10.1006/dbio.1993.1028
- Smalley, D. (2016). *Autonomous Swarmboats: New Missions, Safe Harbors*. Available online at: <https://www.onr.navy.mil/en/Media-Center/Press-Releases/2016/Autonomous-Swarmboats> (accessed May 5, 2018).
- Soares, J. M., Navarro, I., and Martinoli, A. (2016). "The Khepera IV mobile robot: performance evaluation, sensory data and software toolbox," in *Proceedings of the 2nd Iberian Robotics Conference* (Cham: Springer International Publishing), 767–781. doi: 10.1007/978-3-319-27146-0_59
- Stegagno, P., Massidda, C., and Bühlhoff, H. H. (2014). "Object recognition in swarm systems: preliminary results," in *Proceedings of the Workshop on the Centrality of Decentralization in Multi-Robot Systems: Holy Grail or False Idol?* 1–3.
- Suthakorn, J., Cushing, A. B., and Chirikjian, G. S. (2003). An autonomous self-replicating robotic system. *IEEE/ASME Int. Conf. Adv. Intell. Mechatr.* 1, 137–142. doi: 10.1109/AIM.2003.1225085
- Swetha, S., Anandan, R., and Kalaivani, K. (2018). An investigation of micro aerial vehicles (μ AV). *Int. J. Eng. Technol.* 7, 174–177. doi: 10.14419/ijet.v7i2.31.13434
- Telegraph, T. (2018). *The Science Behind the Swarm of Robots Picking Your Grocery Order*. Available online at: <https://www.telegraph.co.uk/technology/2018/06/04/science-behind-swarm-robots-picking-grocery-order/> (accessed August 2, 2018).
- Thrun, S., Burgard, W., and Fox, D. (2000). "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in *Proceedings of the IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, Vol. 1 (San Francisco, CA: IEEE), 321–328.
- Thrun, S., and Liu, Y. (2005). "Multi-robot SLAM with sparse extended information filters," in *Springer Tracts in Advanced Robotics*, Vol. 15, eds D. Paolo and C. Raja (Berlin; Heidelberg: Springer), 254–266. doi: 10.1007/11008941_27
- Tilman, D., Balzer, C., Hill, J., and Belfort, B. L. (2011). Global food demand and the sustainable intensification of agriculture. *Proc. Natl. Acad. Sci. U.S.A.* 108, 20260–20264. doi: 10.1073/pnas.1116437108
- Timmis, J., Andrews, P., and Hart, E. (2010). On artificial immune systems and swarm intelligence. *Swarm Intell.* 4, 247–273. doi: 10.1007/s11721-010-0045-5

- Timmis, J., Ismail, A., Bjerknes, J., and Winfield, A. (2016). An immune-inspired swarm aggregation algorithm for self-healing swarm robotic systems. *Biosystems* 146, 60–76. doi: 10.1016/j.biosystems.2016.04.001
- Trianni, V., and Nolfi, S. (2009). Self-organizing sync in a robotic swarm: a dynamical system view. *IEEE Trans. Evol. Comput.* 13, 722–741. doi: 10.1109/TEVC.2009.2015577
- Varga, M., Basiri, M., Heitz, G., and Floreano, D. (2015). “Distributed formation control of fixed wing micro aerial vehicles for area coverage,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Hamburg), 669–674. doi: 10.1109/IROS.2015.7353444
- Vashev, E., Sterritt, R., Rouff, C., and Hinchey, M. (2012). Swarm technology at NASA: building resilient systems. *IT Prof.* 14, 36–42. doi: 10.1109/MITP.2012.18
- Vitanza, A., Rossetti, P., Mondada, F., and Trianni, V. (2019). Robot swarms as an educational tool: the Thymio’s way. *Int. J. Adv. Robot. Syst.* 16:13. doi: 10.1177/1729881418825186
- von Neumann, J. (1966). *Theory of Self-Reproducing Automata*. University of Illinois Press.
- Wallace, H. (1981). *Vertebrate Limb Regeneration*. Wiley New York.
- Wei, H., Cai, Y., Li, H., Li, D., and Wang, T. (2010). “Sambot: a self-assembly modular robot for swarm robot,” in *Proceedings of the IEEE International Conference on Robotics and Automation* (Anchorage, AK: IEEE), 66–71.
- Wilson, S., Buffin, A., Pratt, S. C., and Berman, S. (2018). Multi-robot replication of ant collective towing behaviours. *R. Soc. Open Sci.* 5:180409. doi: 10.1098/rsos.180409
- Wilson, S., Gameros, R., Sheely, M., Lin, M., Dover, K., Gevorkyan, R., et al. (2016). Pheno, a versatile swarm robotic research and education platform. *IEEE Robot. Autom. Lett.* 1, 884–891. doi: 10.1109/LRA.2016.2524987
- Wyss Institute (2017). *Programmable Robot Swarms*. Available online at: <https://wyss.harvard.edu/technology/programmable-robot-swarms/> (accessed November 6, 2019).
- Yang, X.-S., Cui, Z., Xiao, R., Gandomi, A. H., and Karamanoglu, M. (2013). *Swarm Intelligence and Bio-inspired Computation: Theory and Applications*. Newnes.
- Yang, X.-S., Deb, S., Zhao, Y.-X., Fong, S., and He, X. (2017). Swarm intelligence: past, present and future. *Soft Comput.* 22, 5923–5933. doi: 10.1007/s00500-017-2810-5
- Ye, Y., Boies, S., Liu, J., and Yi, X. (2002). “Collective perception in massive, open, and heterogeneous multi-agent environment,” in *Proceedings of the International Conference on Autonomous Agents* (Bologna), 1175–1182.
- Yim, M., Ying, Z., and Duff, D. (2002). Modular robots. *IEEE Spectr.* 39, 30–34. doi: 10.1109/6.981854
- Zykov, V., Mytilinaios, E., Desnoyer, M., and Lipson, H. (2007). Evolved and designed self-reproducing modular robotics. *IEEE Trans. Robot.* 23, 308–319. doi: 10.1109/TRO.2007.894685

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Schrantz, Umlauf, Sende and Elmenreich. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Guiding the Self-Organization of Cyber-Physical Systems

Carlos Gershenson^{1,2,3*}

¹ Departamento de Ciencias de la Computación, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Mexico City, Mexico, ² Centro de Ciencias de la Complejidad, Universidad Nacional Autónoma de México, Mexico City, Mexico, ³ ITMO University, St Petersburg, Russia

Self-organization offers a promising approach for designing adaptive systems. Given the inherent complexity of most cyber-physical systems, adaptivity is desired, as predictability is limited. Here I summarize different concepts and approaches that can facilitate self-organization in cyber-physical systems, and thus be exploited for design. Then I mention real-world examples of systems where self-organization has managed to provide solutions that outperform classical approaches, in particular related to urban mobility. Finally, I identify when a centralized, distributed, or self-organizing control is more appropriate.

Keywords: complexity, self-organization, information, adaptation, robustness, antifragility

OPEN ACCESS

Edited by:

Heiko Hamann,
University of Lübeck, Germany

Reviewed by:

Payam Zahadat,
University of Graz, Austria
Mark Post,
University of York, United Kingdom
Gabriele Valentini,
Arizona State University,
United States

*Correspondence:

Carlos Gershenson
cgg@unam.mx

Specialty section:

This article was submitted to
Computational Intelligence in
Robotics,
a section of the journal
Frontiers in Robotics and AI

Received: 16 November 2019

Accepted: 09 March 2020

Published: 03 April 2020

Citation:

Gershenson C (2020) Guiding the
Self-Organization of Cyber-Physical
Systems. *Front. Robot. AI* 7:41.
doi: 10.3389/frobt.2020.00041

1. INTRODUCTION

We are submerged in **complexity**. And this complexity is increasing. But what is complexity? There are dozens of definitions and measures in the literature (Lloyd, 2001; Gershenson and Heylighen, 2005), but not a definite one. Well, life is not properly defined either, and it is not a hindrance for biology. Still, to have an idea of what we refer to, let us go to its etymological root. Complexity comes from the Latin *plexus*, which means entwined. In other words, something complex is difficult to separate. This is because the interactions among its components are relevant (Gershenson, 2013b). Relevant because they co-determine the future of the system. Thus, if we do not consider such interactions, but study components in isolation, we will not be able to understand the system properly. Also, interactions can generate novel information, not present in initial nor boundary conditions. This novel information limits predictability (Gershenson, 2013a) and is the source of computational irreducibility (Wolfram, 2002), i.e., there is no shortcut to know the future: one must go through all intermediate steps, because the information produced in the process is required to reach/compute the future.

A recent collaborative effort produced this definition: “Complexity science, also called complex systems science, studies how a large collection of components—locally interacting with each other at small scales—can spontaneously self-organize to exhibit non-trivial global structures and behaviors at larger scales, often without external intervention, central authorities or leaders. The properties of the collection may not be understood or predicted from the full knowledge of its constituents alone. Such a collection is called a complex system and it requires new mathematical frameworks and scientific methodologies for its investigation.” (De Domenico et al., 2019).

One of the core concepts explained in De Domenico et al. (2019) is **self-organization**: “Interactions between components of a complex system may produce a global pattern or behavior. This is often described as self-organization, as there is no central or external controller. Rather, the “control” of a self-organizing system is distributed across components and integrated through their interactions. Self-organization may produce physical/functional structures like crystalline

patterns of materials and morphologies of living organisms, or dynamic/informational behaviors like shoaling behaviors of fish and electrical pulses propagating in animal muscles. As the system becomes more organized by this process, new interaction patterns may emerge over time, potentially leading to the production of greater complexity.” Common examples of self-organizing systems include flocks of birds, schools of fishes, insect swarms, herds, crowds, and other collective phenomena (Camazine et al., 2003; Vicsek and Zafeiris, 2012), although self-organization is not restricted to living systems (Nicolis and Prigogine, 1977; Haken, 1988; Gershenson and Heylighen, 2003; Prokopenko et al., 2009).

There are many cases where self-organization has been used as an approach in **engineering** (Di Marzo Serugendo et al., 2004; De Wolf et al., 2005; Zambonelli and Rana, 2005; Mamei et al., 2006; Helbing et al., 2007; Dressler, 2008; Müller-Schloer et al., 2011; Rohden et al., 2012; Brambilla et al., 2013; Rubenstein et al., 2014; Vászárhelyi et al., 2018). In these cases, we can *describe* a system as self-organizing when elements interact to achieve dynamically a global function or behavior (Gershenson, 2007). In other words, instead of designing directly a solution, one regulates the potential interactions among elements. This is useful in *non-stationary* problems: when the situation changes, then the system adapts by itself. Since interactions in complex systems produce novel information, it is common that this information will change a complex problem. Not only its state, but also its state space. Thus, self-organization can be useful to face complexity by providing general adaptation mechanisms. Several methodologies using self-organization have been proposed (see Frei and Di Marzo Serugendo, 2011 for an overview), although the approach has not been widely applied.

In a parallel effort, **guided self-organization** attempts to combine seemingly opposed processes: design to define and regulate the properties and behavior of a system (one tells the system what to do), and self-organization that implies certain autonomy and adaptability (the system follows its own dynamics) (Prokopenko, 2009, 2014; Ay et al., 2012; Polani et al., 2013). Guided self-organization can be understood as “the steering of the self-organizing dynamics of a system toward a desired configuration” (Gershenson, 2012).

In this paper, I compile concepts and approaches useful for designing self-organizing systems in the physical realm. I illustrate these with case studies from urban mobility before discussing implications. A diagram of the paper structure is shown in **Figure 1**.

2. CONCEPTS

Several concepts are useful to design and guide self-organizing systems. In this section, a non-exhaustive list is presented.

2.1. Adaptation

Adaptation can be defined as a change in an agent or system as a response to a state of its environment that will help the agent or system to fulfill its goals (Gershenson, 2007). Living systems naturally adapt to changes in their environment, and artificial

systems can benefit from exhibiting adaptation (Holland, 1975; Steels and Brooks, 1995; Bedau et al., 2013).

If problems are **stationary**, i.e., do not change, then it is worthwhile attempting to predict the future of a system to control it. However, for **non-stationary** problems, predictability by definition is limited. Novel information generated by interactions in complex systems can lead to non-stationarity. In this case, adaptation is desirable to complement the unpredictable aspects of a problem (Gershenson, 2013a). And self-organization offers a method for building adaptive systems.

For example, city traffic is changing constantly: every time a red light switches to green, the number of waiting vehicles is different. Thus, the timing of the traffic lights should also change to prevent idling. Traditional adaptive traffic light control methods (e.g., Sydney, Dublin, Singapore) use sensors to shift phases depending on recent average demands. This is usually better than not having adaptation, where the best possible option would be to take average measurements, set fixed phases, and perhaps change the programs a few times per day. However, if traffic lights can adapt at the same timescale as the traffic demand does, i.e., every cycle, then the performance would be much improved (Goel et al., 2017).

Adaptation implies *flexibility* and can take place at different timescales: *learning* is relatively fast, *development* occurs during the lifetime of an individual, and *evolution* acts across generations.

2.2. Robustness

A system is **robust** if it continues to function in the face of perturbations (Wagner, 2005), and in general any type of change. As with adaptation, robustness is prevalent in living systems and desirable in artificial ones (Jen, 2005).

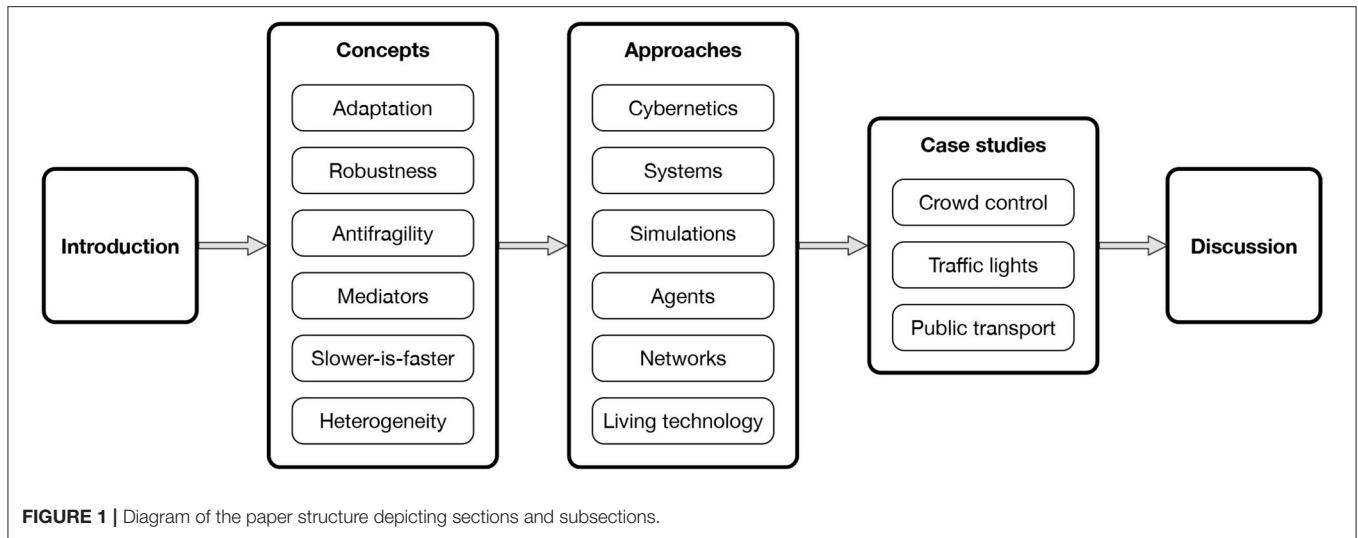
Robustness and adaptability are complementary: a system has to be robust enough to survive while it adapts, and adaptation can favor robustness.

For example, the Internet is quite robust. The TCP/IP protocol was designed to resist nuclear warfare. If any server goes down, other servers will manage to transmit packages, unless the network becomes disconnected. At the structural level (which servers are linked, which pages are linked), self-organization has led to a scale-free topology (Barabási et al., 2000), which is also robust to random failures (although fragile to directed attacks Caldarelli, 2007). This is because only few nodes have several connections, so most probably a random failure will affect a non-important node. However, directed attacks can aim for the hubs.

Robust systems are more prone to be *scalable* than fragile ones. Adding new components or functionality to a system can be seen as a type of perturbation, so in this sense robustness becomes a requirement for scalability.

2.3. Antifragility

A fragile system is damaged by perturbations. A robust system is unaffected by perturbations. An **antifragile** system *benefits* from perturbations (Taleb, 2012). Particular examples of systems that benefit from noise had been already identified (Atlan, 1974), and the concept of antifragility can be seen as a generalization.



For example, the immune system is antifragile. Children who grow up in extremely sanitized conditions are not exposed to pathogens (perturbations), so their immune systems do not develop, leading to stronger infections and allergies in adulthood. Certainly, children should not be infected intentionally, but being exposed to a “normal” amount of pathogens and falling ill now and then is helpful for training the immune system.

We have recently proposed a measure of antifragility (Pineda et al., 2019), which is positive when perturbations improve the performance of a system, negative when perturbations decrease the performance (fragility), and zero when perturbations do not affect the performance (robustness). An important aspect is that there is no “optimal” antifragility independent of an environment. A system should be as antifragile as its environment varies (this is related with **requisite variety**, discussed in section 3.1).

2.4. Mediators

Interactions can be classified as positive, neutral, or negative, depending on the effect they have on the goals of a system (Gershenson, 2007, 2011b).

A **mediator** arbitrates among the elements of a system, to minimize conflict, interferences and frictions (negative interactions); and to maximize cooperation and synergy (positive interactions) (Michod, 2003; Heylighen, 2006; Gershenson, 2007).

Negative interactions, by definition, are those that prevent or damage the functionality, performance, goals, or behavior of a system. Positive interactions would benefit, facilitate, or promote them. Neutral interactions do not affect them. For example, actions that generate a cost but fail to provide a benefit for a society can be said to generate friction, e.g., aggression. If the benefit provided by actions is greater than the cost, one can say that they are synergistic, e.g., politeness. If the cost and benefit balance out, the interactions would be neutral, e.g., tolerance.

Traffic rules can be seen as examples of mediators. They aim at reducing conflict in urban mobility. Without these rules, we

would need to decide constantly on which side of the streets to drive, how to give way, make turns, etc. Even when rules and norms vary from country to country, and in some cases from city to city, when everybody follows the same set of rules (mediators), conflicts tend to be reduced.

Money is another example. It mediates transactions that are much facilitated compared to bartering.

Designing mediators can be useful for regulating systems where the elements cannot be modified. Still, mediators can change the interactions between elements, leading to different systemic behavior and properties (see case study in section 4.1).

2.5. Slower-Is-Faster Effect

Probably this effect was first described about 20 years ago while modeling crowd dynamics (Helbing et al., 2000a,b). If people trying to evacuate a room are panicked (trying to exit faster), then they create friction (negative interactions) that leads to a “turbulent” flow that is slower than if people exit calmly (neutral interactions), thus with a “laminar” flow. The same effect has been studied in vehicular traffic, logistics, public transport, social dynamics, ecological systems, and adaptive processes (Gershenson and Helbing, 2015).

In general, the slower-is-faster effect occurs when a system performs worse as its components try to do better. This implies that a balance between doing “too few” and doing “too much” is necessary. However, in many cases this balance is dynamic, as with antifragility. For example, the optimal speed for highway traffic (that maximizes flow) depends on the vehicular density. For this reason, systems that present a slower-is-faster effect, require constant adaptation, that can be achieved through self-organization.

The slower-is-faster effect may refer to any variable, not only speed. For example, growth or profits are not necessarily maximized in the long term with a short-term maximization strategy. Managing natural resources, such as fisheries, requires this understanding: if all resources are depleted, then in the near future there will be no profits. Maximizing profits requires

a careful balance between short-term action and long-term planning. As with the case of highway traffic, usually this balance is non-stationary.

2.6. Heterogeneity

Most of our models of complex systems are homogeneous: all components have the same properties. This simplification is useful when we face computational limitations. However, increasing processing power and data availability have allowed us to make more realistic models, where different elements of a system have varying properties.

Perhaps the most studied heterogeneity in complex systems is the one of network topologies (Albert and Barabási, 2002; Newman et al., 2006; Gershenson and Prokopenko, 2011; Barabási, 2016) (see section 3.5). Many networks are heterogeneous, with few elements having lots of connections and many elements having few connections. This leads to important differences with homogeneous, regular networks, where all elements have the same number of connections. Apart from the robustness already mentioned, heterogeneous networks can also transmit information faster (they have shorter average path lengths) (Aldana, 2003).

More recently, temporal heterogeneity has been also studied (Cocho et al., 2015; Morales et al., 2018), i.e., systems where different components change at different rates. In a similar way to structural heterogeneity, few elements change slower than most elements. This heterogeneity seems to lead to a balance where slow elements are robust and fast elements are adaptable. In homogeneous systems, this balance is achieved only in phase transitions, which can be characterized as “critical” (Balleza et al., 2008). However, heterogeneity seems to expand the balance beyond criticality, making it easier to search an unknown parameter space, simply because different components diversify any search procedure (Martínez-Arévalo et al., in preparation).

3. APPROACHES

How to implement the properties related to self-organization in cyber-physical systems? The concept of self-organizing systems originated within cybernetics (Ashby, 1947, 1962; von Foerster, 1960; Heylighen et al., 1993), where useful approaches were already developed.

3.1. Cybernetics

Ashby not only coined the term “self-organizing system,” but he also proposed the law of **requisite variety** (Ashby, 1956; Heylighen and Joslyn, 2001; Bar-Yam, 2004; Gershenson, 2015). Variety can be understood as the possible number of states that a system can have. This law states that an *active* controller must have at least as much variety as the system it is trying to control. For example, if we want a robot at a manufacturing plant to deal with seven different types of boxes, then it should be able to distinguish and make the appropriate decisions to handle each type of box. A common problem is that complexity explodes variety and vice versa. Therefore, traditional (non-adaptive) approaches become limited. To handle the variety of a system, we can either reduce its variety (using mediators), or

increase the variety of the controller, but then the latter will imply an increase in the complexity of the controller as well.

Everything else being equal, the variety of *non-stationary* domains will be greater or equal than those of *stationary* ones, as their change usually implies a greater number of potential states. Therefore, **adaptive** controllers and **antifragile** mechanisms have to consider this increased variety.

Active controllers are related with feedforward and feedback (positive or negative) control. Feedback occurs in response to a signal or perturbation, so it can be seen as a type of **adaptation** (Gershenson, 2007). Negative feedback reduces the effect of the perturbation, trying to reach stability, while positive feedback amplifies perturbations, leading to greater change. Feedforward control might be preferred, as it acts on a perturbation or signal before it can affect the controlled. However, this requires *anticipation*, and since complexity implies a limited predictability due to novel information being generated by relevant interactions (non-stationarity), this type of control will also be limited.

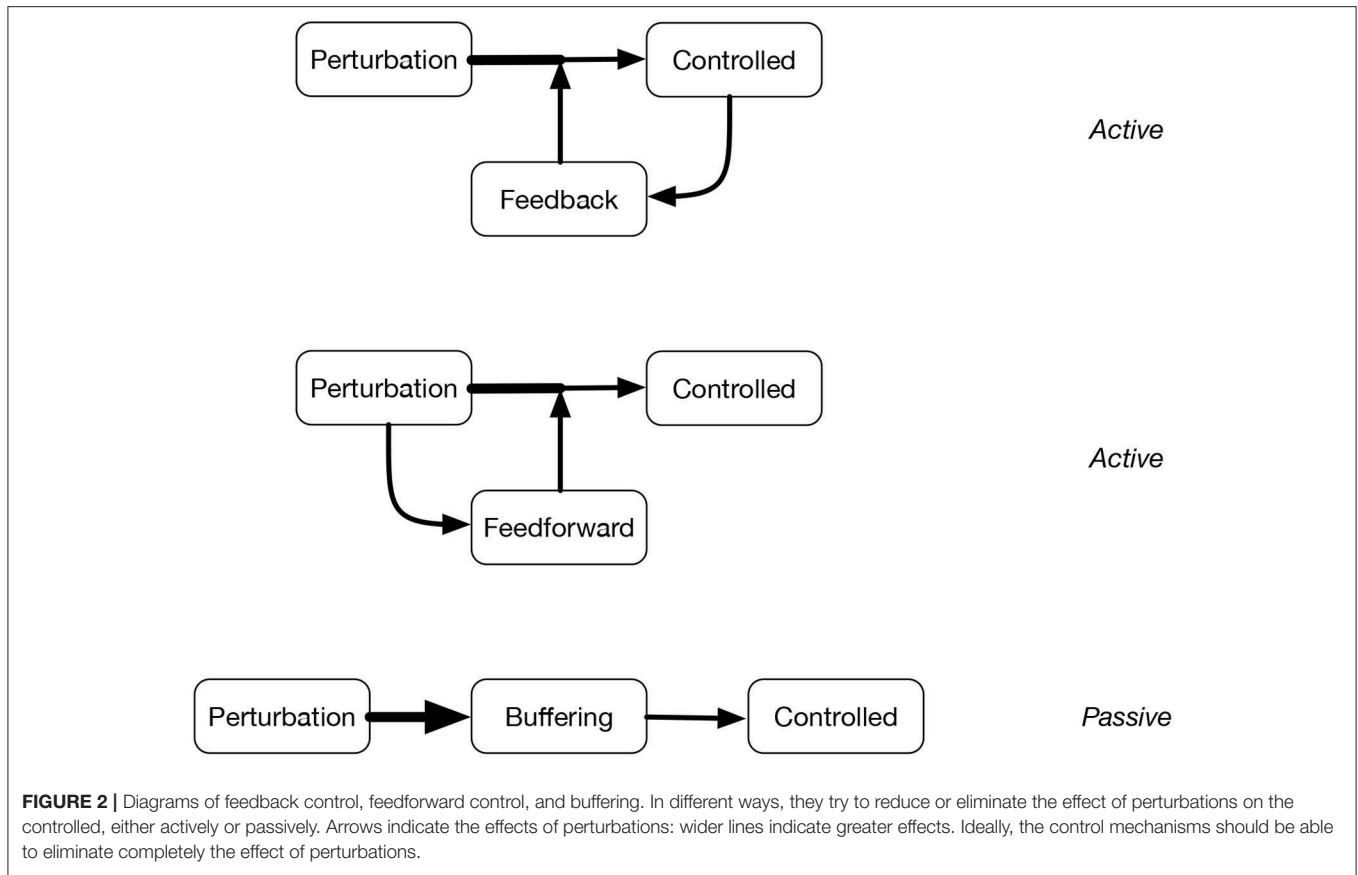
Complementary to active controllers, *passive* controllers were also studied in cybernetics, related to buffering. Passive control can increase the **robustness** of systems, since it prevents perturbations from affecting the controlled. **Figure 2** illustrates active and passive controllers.

There is an interesting relationship between variety and **heterogeneity**. Heterogeneous systems by definition have more variety, so in principle they should be able to control more situations than similar homogeneous systems. However, they might be less robust and more complicated to design and understand. For example, “if there is a system of ten agents each able to solve ten tasks, a homogeneous system will be able to solve ten tasks robustly (if we do not consider combinations as new tasks). A fully heterogeneous system would be able to solve a hundred tasks, but it would be fragile if one agent failed.” (Gershenson, 2007, p. 53). In this case, the homogeneous system would be **robust**, because if one agent fails, others can perform the same function. Still, the **variety** of the system would be restricted to ten tasks. The heterogeneous system would have a tenfold variety, but if a single agent fails, then no other agent would be able to take over the task, and the system would fail as well. Thus, a balance between homogeneity and heterogeneity should also give us a balance between **robustness** and **adaptability** (Langton, 1990; Kauffman, 1993).

3.2. Systems

Contemporary and overlapped with cybernetics, systems theory has also permeated into all disciplines (von Bertalanffy, 1968). The word “system” comes from the ancient Greek *σύστημα* (*sýstema*), which means a whole made of several parts. It is a useful abstraction that can be applied to describe several phenomena at different scales. Moreover, it can be the basis for understanding how elements interact to generate behavior or properties at the system level, and how these properties regulate or constrain the behavior or properties of the elements.

Cybernetics and systems theory naturally merge in cyber-physical systems, where control and communication are required in the understanding and engineering of systems composed



of “bits and atoms,” i.e., digital information is entwined with physical mechanisms.

In a similar way, cyber-social systems are those that merge digital technology and social interactions. The “human factor” increases the **variety** of such systems, and our “creativity” limits even more their predictability.

3.3. Simulations

We can consider computers as telescopes of **complexity** (Pagels, 1989). In other words, without computers, our cognitive abilities are limited to studying models considering not many more than two or three variables. To explore models with thousands or millions of variables, **computer simulations** are necessary (Gershenson, 2007) because of computational irreducibility (Wolfram, 2002). Complexity implies that new information is generated by interactions, so there is no “shortcut” to the future and all intermediate steps are necessary (Wuensche and Lesser, 1992). This limits inherently the predictability of systems (Gershenson, 2013a).

Simulations do not replace other approaches, but their usefulness can be seen in the spreading of computational methods to all disciplines.

Also, simulations allow us to contrast theories in a synthetic way (Steels, 1993). The inductive method validates theories through observation of phenomena. The synthetic method builds artificial systems based on a theory, and then this is validated observing the performance of the artificial system (Simon, 1996).

Since one can contrast different theories using computer simulations, it can be said that computational social sciences are “hardening” the social sciences (Axelrod, 1997; Lazer et al., 2009).

3.4. Agents

Agent-based modeling (Bonabeau, 2002; Schweitzer, 2003; Epstein, 2006; Wilensky and Rand, 2015) has been a useful approach to describe complex systems. An agent can be defined as an entity that *acts* on its environment (Gershenson, 2007). As such, they can be used to model *active* controllers.

Agents have been used to model cognitive systems of different flavors, including rational (Wooldridge and Jennings, 1995), adaptive (Maes, 1994), social (Epstein and Axtell, 1996; Gershenson, 2001), and economic (Arthur, 1999; Challet et al., 2013).

Considering elements of a complex systems as agents, with states, goals, and rules allows us to study how changes at one scale lead to effects at another scale. The effects can go in both directions: changes in agents leading to changes in the system and vice versa. Moreover, systems can also be described as (higher scale) agents.

Another advantage of agent-based modeling is that such models are closer to common language than previous modeling approaches based in e.g., differential equations. Therefore, people do not require a strong mathematical background to develop models using a multi-agent approach.

3.5. Networks

Another approach that is becoming more and more popular as data availability and computing power increase is **network science** (Newman, 2003; Newman et al., 2006; Barabási, 2016). Networks have the benefit of being able to represent naturally elements (nodes) and interactions (links). The relationship between the structure and function of networks has been an intense area of study, where self-organization can play a relevant role (Gershenson, 2012).

Different organizations of the same elements can lead to radically different functionalities. A classical example is different arrangements (allotropes) of carbon atoms, which can lead to charcoal, diamond, graphite, graphene, nanotubes, buckyballs, etc. The components are the same, but changing their organization (structure) leads to radically different properties (function) of these materials.

The **robustness** of systems can be promoted through different mechanisms (Gershenson, 2012), such as redundancy (having several copies of the same element), degeneracy (having different elements perform the same function), modularity (short-range links stronger than long-range ones), and scale-free-like (heterogeneous) topologies (few elements with several links, several elements with few links).

3.6. Living Technology

Ethology—the study of animal behavior—has been taken as an inspiration to build **adaptive** systems (Beer, 1990; Maes, 1994; Steels and Brooks, 1995) and to study complex artificial systems (Rahwan et al., 2019). Animals have evolved to survive in complex environments, so adaptive strategies and self-organizing

mechanisms found in nature have been used in cyber-physical systems. In this sense, **living technology** (Bedau et al., 2009; Gershenson et al., 2018) takes the advantageous properties of living systems and applies them in socio-technical systems, from protocells (Rasmussen et al., 2008) to cities (Gershenson, 2013c).

Living technology has been defined as technology that exhibits the properties of living systems, such as **adaptation**, learning, evolvability, **robustness**, and self-organization. First-order living technology is actually alive, either manipulating existing living systems (Gibson et al., 2010; Kriegman et al., 2020) or (eventually) building them from scratch (Rasmussen et al., 2008; Čejková et al., 2017). Second-order living technology uses living systems as components to achieve the desired properties found in living systems (Benyus, 1997; Liu and Tsui, 2006).

4. CASE STUDIES

In this section, I illustrate the previous concepts and approaches with case studies we have worked with in recent years, related to urban mobility. Particular concepts are highlighted, although approaches are implicitly used.

4.1. Crowd Control

More than a hundred million people use the hundred busiest metro systems in the world every day, a number that is growing fast as the urban population is increasing and cities develop. In the Mexico City Metro and other **cyber-social systems**, people would normally push each other, not letting passengers exit trains, collapsing the systems. How to regulate passenger behavior, when a selfish approach might seem to bring individual

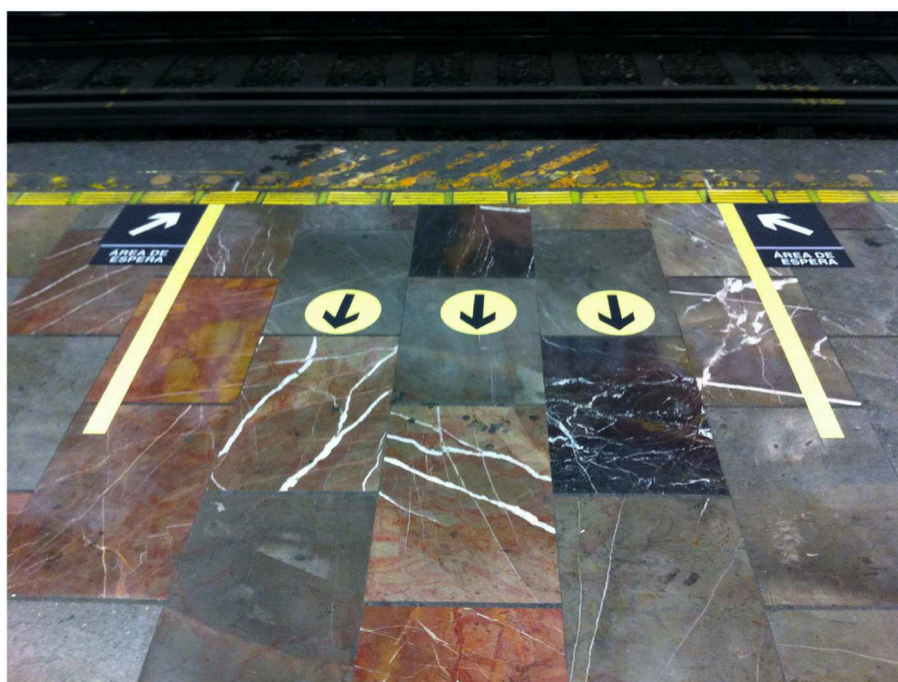


FIGURE 3 | Signs installed to mediate passenger boarding and descent in Mexico City Metro. Reproduced from Carreón et al. (2017) under the Creative Commons CC BY license <https://doi.org/10.1371/journal.pone.0190100.g015>.

benefit but lead to collective inefficiency? One can think of different **mediators**, but they can be costly to try in real systems. To explore alternatives, we first used **simulations** of a model of crowd dynamics (Helbing et al., 2000a) and then implemented a pilot study in the Balderas station of the Mexico City Metro on December, 2016 (Carreón et al., 2017). The pilot was a success and it has since been extended to several other busy stations.

The intervention consisted of “simple” signs that indicate passengers roughly where the train doors will be, asking them to leave free space for exiting passengers, as shown in **Figure 3**. What we did not expect nor suggest was that people would queue (**Figure 4**), and that these queues could even go upstairs as people respected them.

This intervention managed to change the behavior of the passengers and thus the crowd, without changing the elements of the system (where could we get different “educated” passengers from?). The signs **mediated** interactions between people. This is an example of a **passive** control, where interactions are regulated “simply” providing useful information. The mediators managed to change the *structure* of the crowd, leading to a more efficient *function*.

4.2. Traffic Light Coordination

The coordination of traffic lights is an EXP-complete problem, meaning that in theory it takes exponentially more time to find a solution as more intersections are added to a street network. Also, the precise number of vehicles changes every cycle, so in practice the problem changes faster than it can be optimized. An **active** controller should **adapt** as fast as the controlled changes

(**requisite temporal variety**), and for that sensors are required to provide relevant information to the controller.

With this in mind, we have proposed self-organizing algorithms that can coordinate traffic flows and adapt to constant changes in the demand as fast as it changes (Gershenson, 2005; Zapotecatl et al., 2017), achieving close-to-optimal performance (Gershenson and Rosenblueth, 2012). The main idea behind the algorithms is that streets with a higher demand get a preference. This is implemented by counting how many vehicles are approaching or waiting behind red lights, and when the integral over time of this counter reaches a threshold, then the green light is requested. Thus, busier directions will wait less for a green light. This increases the probability that vehicles will aggregate behind red lights with few cars, leading to the formation of platoons. As platoons reach a certain size, they can request a green light before they even reach an intersection (because they quickly reach the threshold), so vehicles do not need to stop, unless there are other vehicles or pedestrians crossing. Platoons are easier to coordinate than individual vehicles, as they leave spaces between them that other platoons can use without interference. When densities are high, the preference is given to the street that has more space after the intersection, preventing gridlocks.

It is difficult to compare the performance of self-organizing traffic lights, as there are no benchmarks in traffic light coordination. However, they are close to optimal. We can define optimality by calculating the maximum performance (measured in terms of velocity or flow) of isolated intersections for different densities. If a **system** with several intersections performs as efficient at every intersection, we can say that the coordination

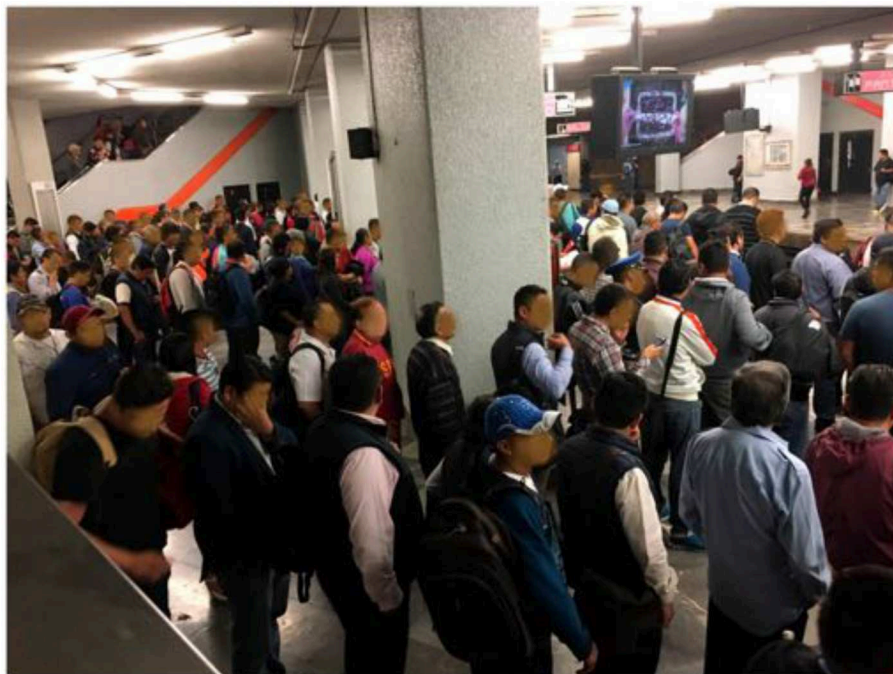
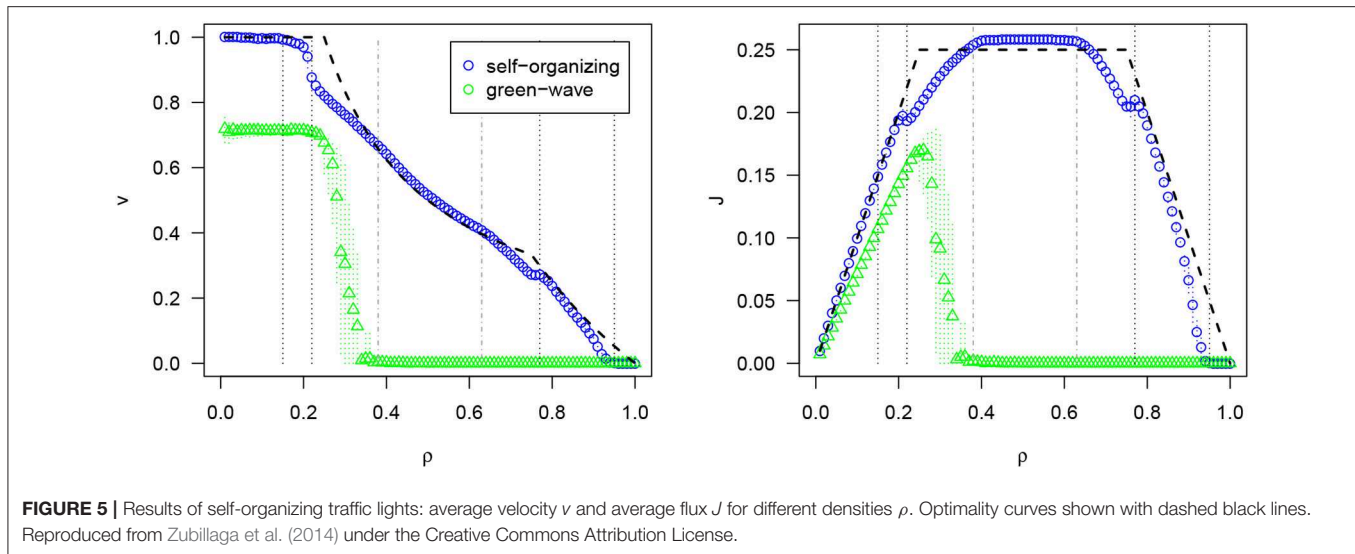


FIGURE 4 | Passengers queuing waiting for a train in Mexico City Metro during rush hour, San Lázaro metro station. Reproduced from Carreón et al. (2017) under the Creative Commons CCBY license <https://doi.org/10.1371/journal.pone.0190100.g016>.



is optimal. **Figure 5** shows a comparison of the self-organizing approach and a traditional top-down control method known as the “green wave” that attempts to offset phases according to the expected speed of vehicles. However, demands change constantly and this method cannot adapt, leading to gridlocks even at medium densities. The self-organizing method achieves optimality for low densities (no vehicle stops) and medium densities (all intersections are used at maximum capacity: there are always vehicles crossing all intersections. Topologically it is not possible to improve this). For other densities, the performance is close to the optimality curves (for details, see Gershenson and Rosenblueth, 2012).

More recently, we have found that self-organizing traffic lights would improve traffic more than if all vehicles were autonomous but with traditional traffic lights. Nevertheless, autonomous vehicles and self-organizing traffic lights are even better (Zapotecatl, 2019).

By distributing control locally, the **requisite variety** of the traffic light coordination can be tackled **robustly** as conditions change, while the formation of platoons **self-organizes** the traffic flows and assists the coordination of intersection controllers at the city scale. In this way, the traffic lights are **mediators** of vehicles, but the vehicles are also **mediators** of traffic lights. We have made simulations with up to ten thousand intersections achieving efficient or optimal coordination, so this solution is certainly scalable.

As there are so many variables involved in this system, **simulations** are necessary to explore and test potential solutions. It is natural to represent the topology of a city as a **network**, where nodes are intersections and links are streets connecting them. Vehicles and traffic lights can be usefully described as **agents**, since they act on their environment. It is worth noting that then traffic lights become part of the environment of vehicles, while vehicles are part of the environment of traffic lights.

4.3. Public Transport Regulation

In theory, passengers in public transport are served optimally when vehicle headway—the time between arrivals at a station—is equal. However, as we have shown, an equal headway configuration is unstable by nature (Gershenson and Pineda, 2009), since delays become amplified by positive feedbacks. Thus, many efforts have been made by transportation engineers to prevent the “equal headway instability,” also known as the “bus bunching problem.”

To keep equal headways, all vehicles—trains, trams, buses—must wait the same time at each station. This time can vary from station to station, but it must be fixed or some vehicles will go faster than others, leading to unequal headways and potentially to the collapse of the **system**. Since the precise number of passengers varies each time a vehicle reaches a station, and thus the required waiting time, then either vehicles will require a margin and be idle, or they will depart before servicing all passengers when these are more than expected.

We proposed a self-organizing algorithm inspired by ant colony communication (Gershenson, 2011a; Carreón et al., 2017), so this can be seen as an example of **living technology**. Some ant species communicate via their environment, a phenomenon known as stigmergy (Theraulaz and Bonabeau, 1999). When they find a food source, they return to their nest leaving a pheromone trail. This indicates the food location to other ants. When they find the food, they can reinforce the trail while returning to their nest. Since pheromones evaporate, once the food is finished, ants stop reinforcing the trail, and they start exploring again. In the case of our algorithm, vehicles can be seen as ants, and we wanted a pheromone-like environmental signal to be used to indicate when the last vehicle had passed. However, pheromones reduce their concentration, while we needed an increasing signal, so we defined “antipheromones” that are secreted by the environment, increase their concentration in time, and are erased by vehicles as they pass.

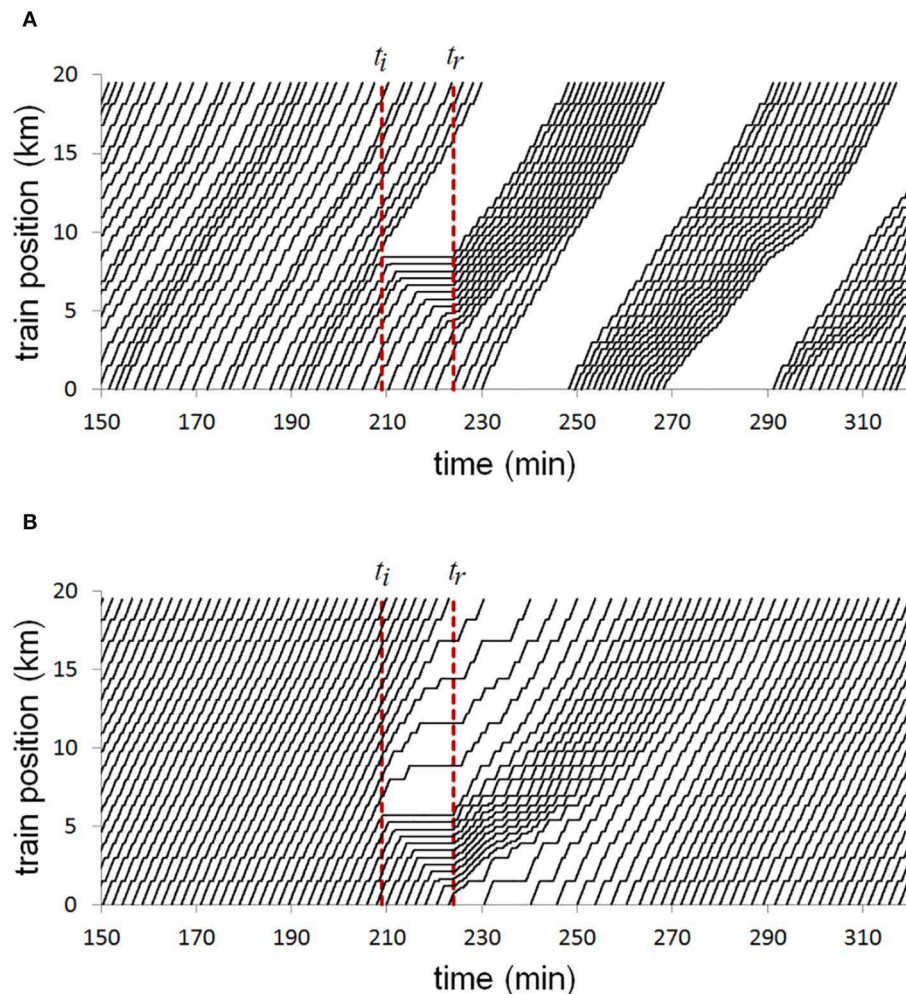


FIGURE 6 | Diagram showing positions of trains at different times. Trains move upwards in distance and rightwards in time. There is an interruption of service at t_i , and it is reestablished 15 min later at t_r . **(A)** Current control method. **(B)** Self-organizing method. Reproduced from Carreón et al. (2017) under the Creative Commons CC BY license <https://doi.org/10.1371/journal.pone.0190100.g014>.

In our algorithm, each vehicle “simply” tries to keep equal distance to the vehicles in front and behind (using antipheromones as **mediators**), but is flexible enough to serve passengers at stations and at the same time prevent idling. Equal headways are not maintained, but the system does not collapse. Rather, its performance is even better than the case with equal headways, i.e., it is supraoptimal. This is because of the **slower-is-faster effect**: It is true that passengers minimize their waiting time at stations with equal headways (as expected by theory). But their total travel time is not independent of the equal headways, so idling will increase their total travel time. With the self-organizing algorithm, passengers wait more at stations, but once they board a vehicle, they will reach their destination faster, as there is no idling. Again, **adaptation** takes place at the scales at which the system changes. We can say that this approach is **antifragile**, as supraoptimality is achieved precisely because of the “noise.” (**heterogeneity**)

of arriving passengers. If all stations had always the same demand (homogeneous), then the self-organizing algorithm would perform as good as the theoretical optimum, i.e., less than supraoptimal.

Figure 6 shows results from a simulation of Line 1 of the Mexico City Metro. On the top panel, the trajectories of trains using the current regulation method is depicted. There is a 15 min interruption of the service, and it can be seen that the system does not recover. In reality, the system does recover, but it requires human intervention and can take one, two, or more hours, depending on the passenger demand. On the bottom panel of **Figure 6**, the trajectories of a similar scenario are shown, but using our self-organizing method. It can be seen that even before the service is reestablished, the vehicles try to maintain equal headways with their neighbors, delaying vehicles ahead of the station where service was interrupted. Once service is reestablished, since the intervals between trains did not collapse,

trains can quickly **adapt** and respond to the delayed service, recovering a desired configuration in less than half an hour.

5. DISCUSSION

We cannot reduce the complexity of several systems we have to deal with. Novel information produced by interactions leads to changes, making problems non-stationary. For example, in the case of traffic lights, one cannot try to optimize intersections in isolation and expect the system to be coordinated. Since the “output” of one intersection becomes the “input” of the next one downstream, this information should be constantly updated by sensors and taken into consideration by controllers.

Self-organization has been used in a broad variety of cyber-physical systems. It allows systems to adapt at the scales at which the problem they are solving changes in a robust fashion. In addition to the case studies mentioned in the previous section, dynamic road pricing in Singapore and variable parking cost in San Francisco are examples of self-organization being used to regulate urban mobility. We can see that the same principles apply in other cyber-physical and cyber-social systems, from telecommunications (Amoretti and Gershenson, 2016) to organizations (Gershenson, 2008).

As in the case of crowd control, there are many systems where we cannot change the components. Still, we can try to mediate interactions to control the function of the system. We will not change politicians. But perhaps we can regulate their interactions to improve politics. We cannot change teachers. But maybe novel mediators can improve education. Businesspeople will not change. But probably promoting certain interactions and restricting others can improve economies. It can take lots of energy to turn charcoal into diamond, but it can be done. They are made of the same atoms. “Only” their organization is different.

A relevant step toward adopting self-organizing controllers is to give up the desire to control completely our systems. This implies accepting that predictability is limited by complexity, and that **adaptation** should complement this inherent uncertainty, even if we do not know how systems will adapt. As complexity limits our predictability, systems require certain autonomy to make the “right decisions.” Even if we use traditional approaches, we do not have full control of our systems, as they are constantly entering unexpected situations. We would like to be able to be sure that our systems will never fail, but they will. We can have formal proofs but these are also limited, since they assume idealized/closed/predefined situations. Self-organizing systems can do the same as traditional engineered systems and more, as they can deal with more realistic/open/variable situations. We just have to (systematically and cautiously) try and see, constantly adapting (Gershenson, 2007). Even if a solution already worked, it does not assure that it will continue working (as conditions change) or that it can be applied in the same way in a different context.

The best solution depends on the context/environment/problem. In some cases, centralized control will be good,

TABLE 1 | Different control approaches are more appropriate for different causalities, complexities, and diversities.

Control	Causality	Complexity	Diversity
Centralized	Top-down	Low	Homogeneous or heterogeneous
Distributed	Bottom-up	Medium	Homogeneous
Self-organizing	Multiscale	High	Homogeneous or heterogeneous

TABLE 2 | Different control types are more related to certain concepts, approaches, and aspects.

Control	Concepts	Approaches	Aspects
Active	Adaptation/antifragility	Agents	Functional
Passive	Robustness/heterogeneity	Networks	Structural

in others distributed is more appropriate, in yet others self-organizing. As shown in **Table 1**, **centralized** control is appropriate when causality should be top-down. Because of the law of requisite variety, systems with a high variety/complexity will require a controller with a high variety/complexity, so the centralized approach becomes less viable. **Distributed** control can deal with a greater complexity, but it is still limited, because the integration of the distributed solutions is not necessarily trivial. This limits distributed control to homogeneous systems: since information flow across the system is restricted, the local solutions assume that each local problem is similar. As illustrated in the traffic lights example, **self-organizing** control can deal with top-down and bottom-up causality (multiscale), as components can interact in a distributed fashion to change system properties (bottom-up), but then the system properties can mediate (top-down) to regulate the behavior of components. Self-organization can be scalable, adaptive, robust, and can deal with a high complexity and homogenous or heterogeneous problems. It is not that one approach is better than others, but they are more appropriate for different problems. Centralized control is easier to implement and understand, but is useful for low complexity/variety problems. Distributed control can deal with a greater complexity, but only for homogeneous, separable systems. Self-organizing systems might be more difficult to design and test, but they can handle greater complexity/variety/diversity.

How the control is organized is certainly relevant, but also whether the control is active or passive. As shown in **Table 2**, **active** control is more related with adaptation and antifragility, as these concepts imply constant change in the **function** of the controller. An agent-based approach is natural here, as it is straightforward to describe actions with agents, since these are entities that act on their environment. On the other hand, **passive** control is more related with robustness and heterogeneity, as these are intrinsic properties of systems and their **structure** (independently on whether there is change or not in the environment). A network description is useful in this case, as the relationships between elements can describe the organization of a system. Note that these are not exclusive, e.g., one can certainly use both active and passive controllers, or combine

agents represented as networks, or study how structure and function affect each other. Also, the concepts and approaches not mentioned here apply to both control cases. Moreover, the relationship between structure and function is far from trivial and has been an open area of research (Heylighen, 1999), since structure defines function but also function can change structure. In many cases, we design structure for a desired function, but also we can design function for a desired structure (Dorigo et al., 2004; Werfel et al., 2014).

As the complexity of our cyber-physical systems increases, and also our understanding of it, we will see more self-organizing approaches. Perhaps names will differ, but the concepts presented here are required to control cyber-physical and cyber-social systems by guiding their self-organization.

REFERENCES

- Albert, R., and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74, 47–97. doi: 10.1103/RevModPhys.74.47
- Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Physica D* 185, 45–66. doi: 10.1016/S0167-2789(03)00174-X
- Amoretti, M. and Gershenson, C. (2016). Measuring the complexity of adaptive peer-to-peer systems. *Peer Peer Netw. Appl.* 9, 1031–1046. doi: 10.1126/science.284.5411.107
- Arthur, W. B. (1999). Complexity and the economy. *Science* 284, 107–109. doi: 10.1126/science.284.5411.107
- Ashby, W. R. (1947). Principles of the self-organizing dynamic system. *J. Gen. Psychol.* 37, 125–128. doi: 10.1080/00221309.1947.9918144
- Ashby, W. R. (1956). *An Introduction to Cybernetics*. London: Chapman & Hall. doi: 10.5962/bhl.title.5851
- Ashby, W. R. (1962). “Principles of the self-organizing system,” in *Principles of Self-Organization*, eds H. V. Foerster and G. W. Zopf Jr. (Oxford: Pergamon), 255–278.
- Atlan, H. (1974). On a formal definition of organization. *J. Theor. Biol.* 45, 295–304. doi: 10.1016/0022-5193(74)90115-5
- Axelrod, R. (1997). “Advancing the art of simulation in the social sciences,” in *Simulating Social Phenomena, Lecture Notes in Economics and Mathematical Systems*, eds R. Conte, R. Hegselmann, and P. Terna (Berlin; Heidelberg: Springer), 21–40.
- Ay, N., Der, R., and Prokopenko, M. (2012). Guided self-organization: perception–action loops of embodied systems. *Theory Biosci.* 131, 125–127. doi: 10.1007/s12064-011-0140-1
- Balleza, E., Alvarez-Buylla, E. R., Chaos, A., Kauffman, S., Shmulevich, I., and Aldana, M. (2008). Critical dynamics in genetic regulatory networks: examples from four kingdoms. *PLoS ONE* 3:e2456. doi: 10.1371/journal.pone.0002456
- Barabási, A.-L. (2016). *Network Science*. Cambridge: Cambridge University Press.
- Barabási, A.-L., Albert, R., and Jeong, H. (2000). Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A* 281, 69–77. doi: 10.1016/S0378-4371(00)00018-2
- Bar-Yam, Y. (2004). Multiscale variety in complex systems. *Complexity* 9, 37–45. doi: 10.1002/cplx.20014
- Bedau, M. A., McCaskill, J. S., Packard, N. H., Parke, E. C., and Rasmussen, S. R. (2013). Introduction to recent developments in living technology. *Artif. Life* 19, 291–298. doi: 10.1162/ARTL_e_00121
- Bedau, M. A., McCaskill, J. S., Packard, N. H., and Rasmussen, S. (2009). Living technology: exploiting life’s principles in technology. *Artif. Life* 16, 89–97. doi: 10.1162/artl.2009.16.1.16103
- Beer, R. D. (1990). *Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology*. San Diego, CA: Academic Press.
- Benyus, J. M. (1997). *Biomimicry: Innovation Inspired by Nature*. New York, NY: William Morrow.
- Bonabeau, E. (2002). Agent-based modeling: methods and techniques for simulating human systems. *Proc. Natl. Acad. Sci. U.S.A.* 99(Suppl. 3):7280–7287. doi: 10.1073/pnas.082080899

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

FUNDING

This work was partially supported by UNAM’s PAPIIT projects IN107919 and IV100120.

ACKNOWLEDGMENTS

I appreciate useful comments from János Kertész, special issue editors, and reviewers.

- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* 7, 1–41. doi: 10.1007/s11721-012-0075-2
- Caldarelli, G. (2007). *Scale-Free Networks*. Oxford: Oxford University Press.
- Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2003). *Self-Organization in Biological Systems*. Princeton, NJ: Princeton University Press.
- Carreón, G., Gershenson, C., and Pineda, L. A. (2017). Improving public transportation systems with self-organization: a headway-based model and regulation of passenger alighting and boarding. *PLoS ONE* 12:e0190100. doi: 10.1371/journal.pone.0190100
- Čejková, J., Banno, T., Hanczyc, M. M., and Štěpánek, F. (2017). Droplets as liquid robots. *Artif. Life* 23, 528–549. doi: 10.1162/ARTL_a_00243
- Challet, D., Marsili, M., and Zhang, Y.-C. (2013). *Minority Games: Interacting Agents in Financial Markets*. Number 9780199686698 in OUP Catalogue. Oxford: Oxford University Press.
- Cocho, G., Flores, J., Gershenson, C., Pineda, C., and Sánchez, S. (2015). Rank diversity of languages: generic behavior in computational linguistics. *PLoS ONE* 10:e0121898. doi: 10.1371/journal.pone.0121898
- De Domenico, M., Camargo, C., Gershenson, C., Goldsmith, D., Jeschonnek, S., Kay, L., et al. (2019). *Complexity Explained: A Grassroot Collaborative Initiative to Create a Set of Essential Concepts of Complex Systems*. Available online at: <https://complexityexplained.github.io>
- De Wolf, T., Samaey, G., and Holvoet, T. (2005). “Engineering self-organising emergent systems with simulation-based scientific analysis,” in *Proceedings of the International Workshop on Engineering Self-Organising Applications (Utrecht)*, 46–160.
- Di Marzo Serugendo, G., Karageorgos, A., Rana, O. F., and Zambonelli, F., editors (2004). *Engineering Self-Organising Systems, Nature-Inspired Approaches to Software Engineering, volume 2977 of Lecture Notes in Computer Science*. Berlin; Heidelberg: Springer.
- Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T. H., Baldassarre, G., et al. (2004). Evolving self-organizing behaviors for a swarm-bot. *Auton. Robots* 17, 223–245. doi: 10.1023/B:AURO.0000033973.24945.f3
- Dressler, F. (2008). “Bio-inspired networking - self-organizing networked embedded systems,” in *Organic Computing*, ed R. P. Würtz (Berlin; Heidelberg: New York, NY: Springer), 285–302.
- Epstein, J. (2006). *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton Studies in Complexity. Princeton, NJ: Princeton University Press.
- Epstein, J. M., and Axtell, R. L. (1996). *Growing Artificial Societies: Social Science from the Bottom Up*. Washington, DC; Cambridge, MA: Brookings Institution Press; MIT Press.
- Frei, R., and Di Marzo Serugendo, G. (2011). Advances in complexity engineering. *Int. J. Bio-Inspired Comput.* 3, 199–212. doi: 10.1504/IJBIC.2011.041144
- Gershenson, C. (2001). *Artificial societies of intelligent agents* (B.E. thesis). Fundaci3n Arturo Rosenblueth, Mexico.
- Gershenson, C. (2005). Self-organizing traffic lights. *Comp. Syst.* 16, 29–53.

- Gershenson, C. (2007). *Design and Control of Self-organizing Systems*. Mexico: CopIt ArXives. Available online at: <http://tinyurl.com/DCSOS2007>
- Gershenson, C. (2008). Towards self-organizing bureaucracies. *Int. J. Public Informat. Syst.* 2008, 1–24.
- Gershenson, C. (2011a). Self-organization leads to supraoptimal performance in public transportation systems. *PLoS ONE* 6:e21469. doi: 10.1371/journal.pone.0021469
- Gershenson, C. (2011b). The sigma profile: a formal tool to study organization and its evolution at multiple scales. *Complexity* 16, 37–44. doi: 10.1002/cplx.20350
- Gershenson, C. (2012). Guiding the self-organization of random Boolean networks. *Theory Biosci.* 131, 181–191. doi: 10.1007/s12064-011-0144-x
- Gershenson, C. (2013a). “Facing complexity: prediction vs. adaptation,” in *Complexity Perspectives on Language, Communication and Society*, eds A. Massip and A. Bastardas (Berlin; Heidelberg: Springer), 3–14.
- Gershenson, C. (2013b). The implications of interactions for science and philosophy. *Found. Sci.* 18, 781–790. doi: 10.1007/s10699-012-9305-8
- Gershenson, C. (2013c). Living in living cities. *Artif. Life* 19, 401–420. doi: 10.1162/ARTL_a_00112
- Gershenson, C. (2015). Requisite variety, autopoiesis, and self-organization. *Kybernetes* 44, 866–873. doi: 10.1108/K-01-2015-0001
- Gershenson, C., and Helbing, D. (2015). When slower is faster. *Complexity* 21, 9–15. doi: 10.1002/cplx.21736
- Gershenson, C., and Heylighen, F. (2003). “When can we call a system self-organizing?” in *Advances in Artificial Life, 7th European Conference, ECAL 2003 LNAI 2801*, eds W. Banzhaf, T. Christaller, P. Dittrich, J. T. Kim, and J. Ziegler (Berlin: Springer), 606–614.
- Gershenson, C., and Heylighen, F. (2005). “Chap. 3: How can we think the complex?” in *Managing Organizational Complexity: Philosophy, Theory and Application*, ed K. Richardson (Greenwich, CT: Information Age Publishing), 47–61.
- Gershenson, C., and Pineda, L. A. (2009). Why does public transport not arrive on time? The pervasiveness of equal headway instability. *PLoS ONE* 4:e7292. doi: 10.1371/journal.pone.0007292
- Gershenson, C., and Prokopenko, M. (2011). Complex networks. *Artif. Life* 17, 259–261. doi: 10.1162/artl_e_00037
- Gershenson, C., and Rosenbluth, D. A. (2012). Self-organizing traffic lights at multiple-street intersections. *Complexity* 17, 23–39. doi: 10.1002/cplx.20392
- Gershenson, C., Trianni, V., Werfel, J., and Sayama, H. (2018). “Self-organization and artificial life: a review,” in *The 2018 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, eds T. Ikegami, N. Virgo, O. Witkowski, M. Oka, R. Suzuki, and H. Iizuka (Tokyo: MIT Press), 510–517.
- Gibson, D. G., Glass, J. I., Lartigue, C., Noskov, V. N., Chuang, R.-Y., Algire, M. A., et al. (2010). Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* 329, 52–56. doi: 10.1126/science.1190719
- Goel, S., Bush, S. F., and Gershenson, C. (2017). Self-organization in traffic lights: evolution of signal control with advances in sensors and communications. *arXiv [preprint] arXiv:1708.07188*.
- Haken, H. (1988). *Information and Self-organization: A Macroscopic Approach to Complex Systems*. Berlin: Springer-Verlag.
- Helbing, D., Farkas, I., and Vicsek, T. (2000a). Simulating dynamical features of escape panic. *Nature* 407, 487–490. doi: 10.1038/35035023
- Helbing, D., Farkas, I. J., and Vicsek, T. (2000b). Freezing by heating in a driven mesoscopic system. *Phys. Rev. Lett.* 84, 1240–1243. doi: 10.1103/PhysRevLett.84.1240
- Helbing, D., Siegmeier, J., and Lämmer, S. (2007). Self-organized network flows. *Netw. Heterogeneous Media* 2, 193–210. doi: 10.3934/nhm.2007.2.193
- Heylighen, F. (1999). “The growth of structural and functional complexity during evolution,” in *The Evolution of Complexity Volume 8 of Einstein Meets Magritte*, eds F. Heylighen, J. Bollen, and A. Riegler (Dordrecht: Kluwer), 17–44.
- Heylighen, F. (2006). “Mediator evolution: a general scenario for the origin of dynamical hierarchies,” in *Worldviews, Science and Us*, eds D. Aerts, B. D’Hooghe, and N. Note (Singapore: World Scientific), 45–48.
- Heylighen, F., and Joslyn, C. (2001). “Cybernetics and second order cybernetics,” in *Encyclopedia of Physical Science and Technology, 3rd Edn.*, Vol. 4, ed R. A. Meyers (New York, NY: Academic Press), 155–170.
- Heylighen, F., Joslyn, C., and Turchin, V. (1993). *Principia Cybernetica Web*. Available online at: <http://pcp.vub.ac.be>
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: The University of Michigan Press.
- Jen, E. (ed.). (2005). *Robust Design: A Repertoire of Biological, Ecological, and Engineering Case Studies*. Oxford: Santa Fe Institute Studies on the Sciences of Complexity, Oxford University Press.
- Kauffman, S. A. (1993). *The Origins of Order*. Oxford: Oxford University Press.
- Kriegman, S., Blackiston, D., Levin, M., and Bongard, J. (2020). A scalable pipeline for designing reconfigurable organisms. *Proc. Natl. Acad. Sci. U.S.A.* 117, 1853–1859. doi: 10.1073/pnas.1910837117
- Langton, C. G. (1990). Computation at the edge of chaos: phase transitions and emergent computation. *Physica D* 42, 12–37. doi: 10.1016/0167-2789(90)90064-V
- Lazer, D., Pentland, A. S., Adamic, L., Aral, S., Barabasi, A. L., Brewer, D., et al. (2009). Life in the network: the coming age of computational social science. *Science* 323:721. doi: 10.1126/science.1167742
- Liu, J., and Tsui, K. C. (2006). Toward nature-inspired computing. *Commun. ACM* 49, 59–64. doi: 10.1145/1164394.1164395
- Lloyd, S. (2001). *Measures of Complexity: A Non-exhaustive List*. Cambridge, MA: Department of Mechanical Engineering, Massachusetts Institute of Technology.
- Maes, P. (1994). Modeling adaptive autonomous agents. *Artif. Life* 1, 135–162. doi: 10.1162/artl.1993.1.1_2.135
- Mamei, M., Menezes, R., Toksdorf, R., and Zambonelli, F. (2006). Case studies for self-organization in computer science. *J. Syst. Architect.* 52, 443–460. doi: 10.1016/j.sysarc.2006.02.002
- Michod, R. E. (2003). “Cooperation and conflict mediation during the origin of multicellularity,” in *Genetic and Cultural Evolution of Cooperation*, Ch. 16, ed P. Hammerstein (Cambridge, MA: MIT Press), 261–307.
- Morales, J. A., Colman, E., Sánchez, S., Sánchez-Puig, F., Pineda, C., Iniguez, G., et al. (2018). Rank dynamics of word usage at multiple scales. *Front. Phys.* 6:45. doi: 10.3389/fphys.2018.00045
- Müller-Schloer, C., Schmeck, H., and Ungerer, T. (eds.). (2011). *Organic Computing — A Paradigm Shift for Complex Systems*. Autonomic Systems. Basel: Springer.
- Newman, M., Barabási, A.-L., and Watts, D. J., editors (2006). *The Structure and Dynamics of Networks*. Princeton Studies in Complexity. Princeton, NJ: Princeton University Press.
- Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Rev.* 45, 167–256. doi: 10.1137/S003614450342480
- Nicolis, G., and Prigogine, I. (1977). *Self-Organization in Non-Equilibrium Systems: From Dissipative Structures to Order Through Fluctuations*. Chichester: Wiley.
- Pagels, H. R. (1989). *The Dreams of Reason: The Computer and the Rise of the Sciences of Complexity*. New York, NY: Bantam Books.
- Pineda, O. K., Kim, H., and Gershenson, C. (2019). A novel antifragility measure based on satisfaction and its application to random and biological Boolean networks. *Complexity* 2019:10. doi: 10.1155/2019/3728621
- Polani, D., Prokopenko, M., and Yeager, L. S. (2013). Information and self-organization of behavior. *Adv. Comp. Syst.* 16:1303001. doi: 10.1142/S021952591303001X
- Prokopenko, M. (2009). Guided self-organization. *HFSP J.* 3, 287–289. doi: 10.2976/1.3233933
- Prokopenko, M. (ed.). (2014). *Guided Self-Organization: Inception, volume 9 of Emergence, Complexity and Computation*. Berlin; Heidelberg: Springer.
- Prokopenko, M., Boschetti, F., and Ryan, A. (2009). An information-theoretic primer on complexity, self-organisation and emergence. *Complexity* 15, 11–28. doi: 10.1002/cplx.20249
- Rahwan, I., Cebrian, M., Obradovich, N., Bongard, J., Bonnefon, J.-F., Breazeal, C., et al. (2019). Machine behaviour. *Nature* 568, 477–486. doi: 10.1038/s41586-019-1138-y
- Rasmussen, S., Bedau, M. A., Chen, L., Deamer, D., Krakauer, D. C., Packard, N. H., et al. (eds.). (2008). *Protocells: Bridging Nonliving and Living Matter*. Cambridge, MA: MIT Press.
- Rohden, M., Sorge, A., Timme, M., and Witthaut, D. (2012). Self-organized synchronization in decentralized power grids. *Phys. Rev. Lett.* 109:064101. doi: 10.1103/PhysRevLett.109.064101

- Rubenstein, M., Cornejo, A., and Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science* 345, 795–799. doi: 10.1126/science.1254295
- Schweitzer, F. (2003). *Brownian Agents and Active Particles. Collective Dynamics in the Natural and Social Sciences*. Springer Series in Synergetics. Berlin: Springer.
- Simon, H. A. (1996). *The Sciences of the Artificial, 3rd Edn*. Cambridge, MA: MIT Press.
- Steels, L. (1993). “Building agents out of autonomous behavior systems,” in *The Artificial Life Route to Artificial Intelligence: Building Embodied Situated Agents*, eds L. Steels and R. A. Brooks (New Haven, CT: Lawrence Erlbaum), 102–137.
- Steels, L., and Brooks, R. (1995). *The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents*. New York, NY: Lawrence Erlbaum Associates.
- Taleb, N. N. (2012). *Antifragile: Things That Gain From Disorder*. London: Random House.
- Theraulaz, G., and Bonabeau, E. (1999). A brief history of stigmergy. *Artif. Life* 5, 97–116. doi: 10.1162/106454699568700
- Vásárhelyi, G., Virágh, C., Somorjai, G., Nepusz, T., Eiben, A. E., and Vicsek, T. (2018). Optimized flocking of autonomous drones in confined environments. *Sci. Robot.* 3:20. doi: 10.1126/scirobotics.aat3536
- Vicsek, T., and Zafeiris, A. (2012). Collective motion. *Phys. Rep.* 517, 71–140. doi: 10.1016/j.physrep.2012.03.004
- von Bertalanffy, L. (1968). *General System Theory: Foundations, Development, Applications*. New York, NY: George Braziller.
- von Foerster, H. (1960). “On self-organizing systems and their environments,” in *Self-Organizing Systems*, eds M. C. Yovitts and S. Cameron (New York, NY: Pergamon), 31–50.
- Wagner, A. (2005). *Robustness and Evolvability in Living Systems*. Princeton, NJ: Princeton University Press.
- Werfel, J., Petersen, K., and Nagpal, R. (2014). Designing collective behavior in a termite-inspired robot construction team. *Science* 343, 754–758. doi: 10.1126/science.1245842
- Wilensky, U., and Rand, W. (2015). *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo*. Cambridge, MA: MIT Press.
- Wolfram, S. (2002). *A New Kind of Science*. Champaign, IL: Wolfram Media.
- Wooldridge, M., and Jennings, N. R. (1995). Intelligent agents: theory and practice. *Knowl. Eng. Rev.* 10, 115–152. doi: 10.1017/S0269888900008122
- Wuensche, A., and Lesser, M. (1992). *The Global Dynamics of Cellular Automata; An Atlas of Basin of Attraction Fields of One-Dimensional Cellular Automata*. Reading, MA: Santa Fe Institute Studies in the Sciences of Complexity. Addison-Wesley.
- Zambonelli, F., and Rana, O. F. (2005). Self-organization in distributed systems engineering: introduction to the special issue. *IEEE Trans. Syst. Man Cybernet. Part A* 35, 313–315. doi: 10.1109/TSMCA.2006.846372
- Zapotecatl, J. L. (2019). *Algoritmos de control de semáforos auto-organizados para incrementar el flujo de tránsito vehicular basados en autómatas celulares* (Ph.D. thesis). Universidad Nacional Autónoma de México, Mexico.
- Zapotecatl, J. L., Rosenblueth, D. A., and Gershenson, C. (2017). Deliberative self-organizing traffic lights with elementary cellular automata. *Complexity* 2017:7691370. doi: 10.1155/2017/7691370
- Zubillaga, D., Cruz, G., Aguilar, L. D., Zapotécatl, J., Fernández, N., Aguilar, J., et al. (2014). Measuring the complexity of self-organizing traffic lights. *Entropy* 16, 2384–2407. doi: 10.3390/e16052384

Conflict of Interest: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer PZ declared a past co-authorship with one of the authors CG to the handling editor.

Copyright © 2020 Gershenson. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Scalable and Robust Fabrication, Operation, and Control of Compliant Modular Robots

Nialah Jenae Wilson^{1*†}, Steven Ceron^{1†}, Logan Horowitz² and Kirstin Petersen²

¹ Collective Embodied Intelligence Lab, Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, United States,

² Collective Embodied Intelligence Lab, Electrical and Computer Engineering, Cornell University, Ithaca, NY, United States

OPEN ACCESS

Edited by:

Carlo Pinciroli,
Worcester Polytechnic Institute,
United States

Reviewed by:

Giovanni Beltrame,
École Polytechnique de Montréal,
Canada
Alan Gregory Millard,
University of Lincoln, United Kingdom

*Correspondence:

Nialah Jenae Wilson
njw68@cornell.edu

[†] These authors have contributed
equally to this work

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 12 November 2019

Accepted: 13 March 2020

Published: 07 April 2020

Citation:

Wilson NJ, Ceron S, Horowitz L and
Petersen K (2020) Scalable and
Robust Fabrication, Operation, and
Control of Compliant Modular Robots.
Front. Robot. AI 7:44.
doi: 10.3389/frobt.2020.00044

A major goal of autonomous robot collectives is to robustly perform complex tasks in unstructured environments by leveraging hardware redundancy and the emergent ability to adapt to perturbations. In such collectives, large numbers is a major contributor to system-level robustness. Designing robot collectives, however, requires more than isolated development of hardware and software that supports large scales. Rather, to support scalability, we must also incorporate robust constituents and weigh interrelated design choices that span fabrication, operation, and control with an explicit focus on achieving system-level robustness. Following this philosophy, we present the first iteration of a new framework toward a scalable and robust, planar, modular robot collective capable of gradient tracking in cluttered environments. To support co-design, our framework consists of hardware, low-level motion primitives, and control algorithms validated through a kinematic simulation environment. We discuss how modules made primarily of flexible printed circuit boards enable inexpensive, rapid, low-precision manufacturing; safe interactions between modules and their environment; and large-scale lattice structures beyond what manufacturing tolerances allow using rigid parts. To support redundancy, our proposed modules have on-board processing, sensing, and communication. To lower wear and consequently maintenance, modules have no internally moving parts, and instead move collaboratively via switchable magnets on their perimeter. These magnets can be in any of three states enabling a large range of module configurations and motion primitives, in turn supporting higher system adaptability. We introduce and compare several controllers that can plan in the collective's configuration space without restricting motion to a discrete occupancy grid as has been done in many past planners. We show how we can incentively redundant connections to prevent single-module failures from causing collective-wide failure, explore bad configurations which impede progress as a result of the motion constraints, and discuss an alternative "naive" planner with improved performance in both clutter-free and cluttered environments. This dedicated focus on system-level robustness over all parts of a complete design cycle, advances the state-of-the-art robots capable of long-term exploration.

Keywords: self-reconfigurable, modular robots, soft robots, robot kinematics, simulation environment, path planning

1. INTRODUCTION

Modular self-reconfigurable robots are composed of active modules capable of rearranging their connection topology to adapt to dynamic environments, changing task settings, and partial failures (Yim et al., 2007). It is desirable to increase the number of modules to increase the potential for adaptability and redundancy, however, scaling up the collective size poses several challenges (Brunete et al., 2017). Controllers must be capable of efficiently exploring the configuration space and providing introspection to cope with internal and external changes. The module hardware must be inexpensive and fast to produce, work reliably, and require little maintenance. Consequently, isolated efforts to develop scalable control and hardware do not necessarily result in system-level robustness. Rather, to facilitate large numbers of robots in the first place, we argue for the importance of incorporating robustness into all levels of design, and demonstrate how this approach leads to tightly co-dependent parameters across hardware and software. In this paper, we discuss our design approach, an early hardware prototype, and custom controllers. Our focus is explicitly on enabling long-term robustness of an autonomous, self-reconfigurable, modular robot through a hardware-software design cycle, with the idea that we can build on such a robust platform in the future to achieve more advanced behaviors.

Figure 1A provides an overview of the measures we have taken to ensure system-level robustness, and how many of these design decisions carry over between fabrication, operation, and control. Related to the *design* itself, system robustness is mediated by (1) the simultaneous development of hardware and software; (2) ease of iterations, e.g., through realistic simulation environments that let the designer focus on high level behaviors, as well as simple hardware that supports easy extensions; and (3) open access to permit a wide range of users and inputs. To support inexpensive, fast, and therefore *scalable fabrication* we focus on (1) simple designs with minimal components; (2) mechanical compliance to permit higher manufacturing tolerances; and (3) manufacturing rigs to support non-expert labor. These design parameters correlate with those of *scalable operation*, e.g., because (1) compliance lets modules interact safely with each other and with external objects; (2) compliance permits large scale connectivity despite poor manufacturing tolerances; and (3) hardware simplicity limits the risk of failure. Other operation-specific considerations include the ability of modules to operate, sense, and perceive independently from others; the ability to stay connected without continuous use of power; the ability of modules to move in a multitude of ways to overcome partial failures; and the potential to lower mechanical wear by omitting internally moving parts. All of these design choices warrant custom *controllers* and to support system robustness, we focus on (1) reactive (over deterministic) behaviors that could adapt to dynamic perturbations; (2) naive and simple control schemes that scale well with the number of robots; (3) minimum energy expenditure through efficient path planners; (4) connection redundancy to avoid single module failures from causing complete collective failure; and (5) enabling

a large configuration space that facilitates system adaptability to unforeseen perturbations.

More specifically, we introduce a novel planar, modular robot composed of compliant modules moving in unison. We refer to the robotic modules as “DONUTs” (Deformable Self-Organizing Nomadic Units) for their visual kinship (**Figures 1B–D**). To support simple and fast manufacturing, DONUT modules are composed of a single flexible printed circuit board (PCB) wrapped in a loop and populated with sensors, actuators, processors, and room for batteries. To mitigate wear, the DONUTs have no moving parts; rather, they move as a collective by activating and deactivating Simplified Electro-Permanent Magnets (SEPs) on their perimeter. These magnets can be polarized in either direction or turned off to enable a very large configuration space and consequently collective adaptability (**Figure 1C**). Furthermore, they do not require continued application of power to maintain polarization which saves energy. To lower fabrication cost and risk of errors, we minimize the number of components, e.g., by making double use of the PCB as a chassis and the SEPs for communication. The passive compliance introduced with the flexible PCB permits large lattice configurations despite rapid, imprecise manufacturing. The compliance and low driving voltages also enable the modules to interact safely with each other and with surrounding objects.

We further develop DONUT-specific coordination schemes, low-level primitives for module operation, as well as an open source simulation environment to support controller development. We refrain from imposing artificial constraints on module motion beyond what the hardware is capable of. This means that the modules operate in a grid-free environment and can achieve a much larger set of connection topologies to adapt to the task at hand. Toward real-world operation, we furthermore focus on reactive configurations, rather than predetermined shape transitions as is common for modular robots. Specifically, in a simulated energy harvesting scenario, we investigate how such modules may perform gradient tracking toward a light source in clutter-free and cluttered environments (**Figure 1D**). We choose this specific task, because it supports reactive and scalable behavior and because it highlights the benefits of grid-free operation. To evaluate the performance, in terms of path efficiency, of our controllers, we compare them against paths generated by an all-knowing Oracle planner in clutter-free environments. We explore a locally optimal A* search-based controller and how we may incentively redundant connection topologies for more error tolerant operation. We compare this to a “naive” iterative control scheme that scales better with the number of modules in both computation and memory, finding comparable performance. We further discuss particular connection topologies which may impede progress due to the hardware-specific motion constraints, and show how these may be circumvented using the naive controller. We also allude to how energy expenditure is used across modules in the collective, which is an interesting area for future work. In this paper, we focus only on centralized coordination, however, all of the methods may be adapted for decentralized coordination at the expense of communication.

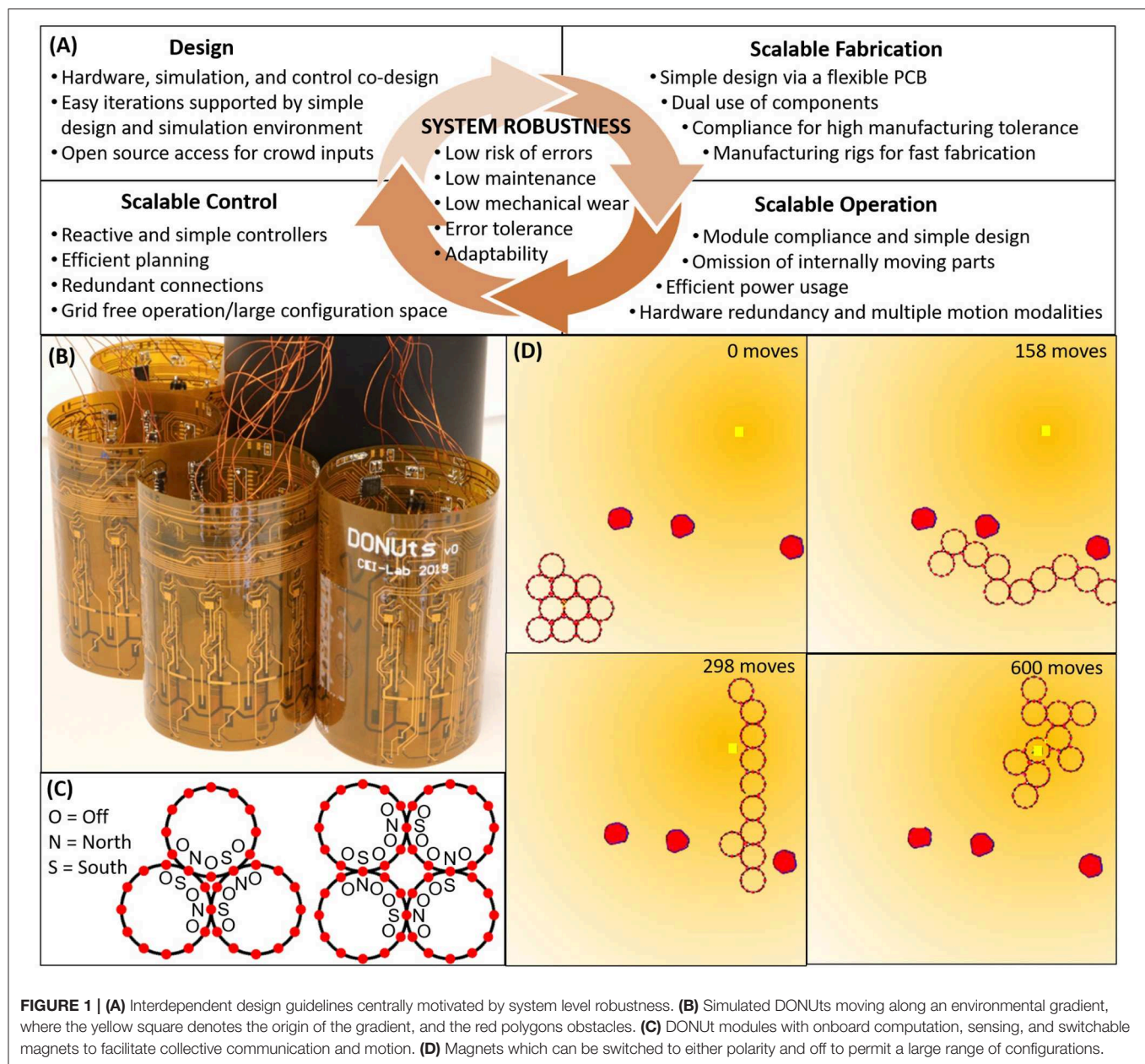


FIGURE 1 | (A) Interdependent design guidelines centrally motivated by system level robustness. (B) Simulated DONUTs moving along an environmental gradient, where the yellow square denotes the origin of the gradient, and the red polygons obstacles. (C) DONUT modules with onboard computation, sensing, and switchable magnets to facilitate collective communication and motion. (D) Magnets which can be switched to either polarity and off to permit a large range of configurations.

Although more work is needed to demonstrate full-scale practical collective operation, the work in this paper illustrates the highly interdependent design choices that lay the foundation for a scalable and robust modular robot. The following sections detail (1) related work of both controllers and hardware; (2) a hardware prototype composed of compliant modules with individual computation, communication, sensing, and collective motion; (3) an inexpensive, quick manufacturing process for both modules and components, based on pre-populated, flexible PCB and a rapid SEP winding mechanism; (4) a characterization of module deformation, mobility, sensing, and communication; (5) an open source kinematic simulation framework for the DONUTs informed by low-level motion primitives and experimentally obtained sensor performance characteristics; and

(6) a comparative study of two controllers for efficient and error tolerant gradient tracking with the DONUTs in environments without an occupancy grid.

2. RELATED WORK

The framework described in this paper combines and builds on findings from many sources spanning both hardware and coordination. In the following sections, we describe these in turn.

2.1. Modular Robot Platforms

Past research on hardware for modular self-reconfigurable robots includes design of inexpensive and durable mechanisms for actuation, docking, communication, and power distribution

(Brunete et al., 2017). Low maintenance requirements are especially important for this class of robots, as they scale linearly with the number of modules required. Module cost and fabrication time are equally important factors, but are somewhat mitigated by the fact that unit price decreases significantly with mass fabrication. Additionally, the module weight and stiffness determines both structural stability and how many modules can be moved at once.

The majority of modular robots consist of rigid components assembled into either a fixed form factor (Jorgensen et al., 2004; Goldstein et al., 2005; Daudelin et al., 2018; Zhu and El Baz, 2019), or into modules which can actively deform to produce motion (Rus and Vona, 2001; Ishiguro et al., 2006; Karagozler et al., 2007; Li et al., 2019). Recently, merging with soft robotics, pneumatically-driven modules with infinite degrees of (passive) freedom have also been shown (Lee et al., 2016; Vergara et al., 2017). These have the benefit of overcoming small manufacturing defects that otherwise scale poorly in large lattice-structures. The most successful demonstrations of these robots currently rely on traditional electro-mechanical actuators for reconfiguration, such as DC motors (Daudelin et al., 2018). However, researchers are also exploring designs that require fewer components and (1) have no internal moving parts which are prone to wear (Goldstein et al., 2005; Vergara et al., 2017; Zhu and El Baz, 2019), (2) rely solely on collective motion over individual module mobility (Goldstein et al., 2005; Li et al., 2019), and (3) exploit non-mechanical latches, such as switchable magnets (Goldstein et al., 2005; Gilpin and Rus, 2010; Zhu and El Baz, 2019), electrostatics (Karagozler et al., 2007), and meltable plastic and alloys (Neubert et al., 2014; Swisler and Rubenstein, 2018). Note that the last two options are superior for connection strength, but require high voltage generation or power usage, respectively. The DONUTs are intended for rapid reconfiguration, standardize operation, and will not experience high tensile force, therefore we base our design on switchable magnets.

Currently, the closest “relatives” of the DONUTs are the Caroms (Goldstein et al., 2005) and the Nonoperable (Gilpin and Rus, 2010), both planar modular robots. In the former, round, rigid modules can move in six discrete steps around each other using switchable magnets. This is still an active research platform, especially in terms of controllers, power, connectors, and communication (Campbell et al., 2005; Kirby et al., 2007; Naz et al., 2018; Piranda and Bourgeois, 2018). The DONUTs rely on a similar means of locomotion, but are compliant, simpler to manufacture, and have the potential to be teacherless. The Nonoperable are small form-factor cubes with switchable magnets used both for inter-module docking, power transfer, and communication; module movement comes from external forces. They involve a quick manufacturing procedure by wrapping a flexible PCB around a rigid frame, enabling deflections to overcome manufacturing defects.

It is worth noting our specific choice of an SEP docking mechanism. In the Caroms and Nonoperable, the switchable magnets were electromagnets and electro-permanent magnets, respectively. The former has high power consumption when on, and the latter can only be switched off or on in one polarity. To limit power consumption and to permit a wider range of

configurations (**Figure 1C**), we instead leverage SEPs (Zhu and El Baz, 2019) which can switch polarities and be turned off. We further explore different SEP designs to lower module weight and enable stand-alone operation.

In summary, the design of the DONUTs combines many of these past findings, including: (1) passive module compliance to overcome manufacturing defects, (2) collective motion via switchable magnets to decrease mechanical wear, and (3) a very simple fabrication process to improve system scalability.

2.2. Coordination of Modular Robots

Path planners for modular, lattice-based robots typically focus on shape transition, i.e., how to plan admissible and energy efficient paths for all modules from one configuration to another (Pamecha et al., 1997; Walter et al., 2005). Past literature on reactive reconfiguration to reach a goal in a cluttered environment is much more sparse, but has been shown with slime mold-inspired, crystalline, and prismatic modules (Kubica et al., 2001; Rus and Vona, 2001; Butler et al., 2004; Ishiguro et al., 2006; Li et al., 2019), through coupled oscillators, traditional path planners, and cellular automata, respectively. All of these were based on distributed controllers and hardware with active degrees of deformation to help the modules move. In contrast, the DONUT modules briefly presented in Ceron et al. (2019a), have only passive compliance. Although this passive compliance is not currently part of our simulation framework, the presented control algorithms are only dependent on the connection topology and sensed objects, not the actual robot morphology, and could therefore work on the real hardware. We reason further about the benefits of module deformability and strain sensing in Ceron et al. (2019b).

The majority of research has focused on distributed controllers, e.g., through agent automata and globally imposed, or module-generated, gradients (Butler et al., 2004; Stoy and Nagpal, 2004). Centralized path planners for optimal shape transition become computationally intractable as the number of modules grow. This is typically overcome through careful preplanning (Walter et al., 2002; Daudelin et al., 2018) or sub-optimal planners dealing with hierarchical layers of modules (Bhat et al., 2006). The planning is further simplified through discrete occupancy grids (triangles, squares/cubes, and hexagons/rhombic dodecahedrons). Controllers for the Caroms, for example, typically discretize the world into hexagonal cells (Walter et al., 2005; Bhat et al., 2006). Although this approach is convenient mathematically, it also artificially limits the set of achievable configurations, which becomes especially critical in modules that are dependent on others to move.

Here, we explore centralized control schemes which adds no constraints on the module configuration beyond what the hardware is capable of. Similar to Ishiguro et al. (2006) and Li et al. (2019) we do not divide the world into a fixed occupancy grid, however, each module does have a finite set of connection points. Also, similar to many past controllers, path admissibility is ensured through a globally connected topology and consecutive movement of modules. Centralized controllers can suffer from a single point of failure and requires the need of a global sensor (or global communication), however, the

algorithms we present rely on knowledge and plans which could be computed locally to overcome such weaknesses, at the expense of added synchronized communication.

3. MODULE DESIGN AND CHARACTERIZATION

We start by describing the SEPs, as they dominate the module infrastructure, power consumption, weight, and assembly time. We then detail the remainder of the hardware (**Figure 2A**), characterize the module ability to move, deform, communicate, and sense, and end with a discussion on scalability. As previously mentioned, our design considerations are based on enabling long term, stand-alone operation.

3.1. Docking Mechanism

SEPs consist of a low coercivity magnet wound with a copper coil and finished with ferrous end caps to induce and guide the magnetic field, respectively. By sending a high current pulse through the coil we can orient all the dipoles in the core to change its overall polarity; by applying a pulse of lower current magnitude, we can effectively turn off the magnet (**Figure 3A** inset). SEPs are advantageous for modular robots because (1) they have no internally moving parts which limit wear, (2) they remain polarized without continued supply of power which lowers maintenance, (3) they can be used for both movement and communication which minimizes the number of components, and (4) they can switch between opposite polarizations and off which supports a large range of module configurations. This section details our first SEP design and how it relates to the rest of the module design.

The SEP design considerations include part accessibility, the magnet geometry and coercivity, the geometry of the end cap, the number of turns in the coil, the wire gauge, and the amount of energy that can be transferred to the coil, which in turn is dependent on the supply voltage, series resistance, and pulse duration. These considerations come with trade-offs: a stronger SEP will facilitate better bonding strength and require fewer on-board SEPs needed for actuation, whereas a weaker SEP weighs less and can therefore work on a lower weight module that is easier to move. Our SEP design is based on a careful balance of these parameters. Small-scale, off-the-shelf, low coercivity magnets are rare and therefore the availability of these dominated our design. We decided on a magnet made of Alnico grade 5, with a length of 3/8" and a diameter of 1/8", available from Magnet Kingdom. The end caps are made of steel and manually cut to the dimensions $4 \times 4 \times 1.5$ mm.

A high energy pulse, and therefore a high supply voltage, is needed to flip the dipoles in the Alnico magnet. To keep the modules light weight, small, and mobile, we target a single cell Lithium Polymer on-board battery with a 3.7 V output. To activate the SEPs, we boost the battery voltage from 3.7 to 26 V, using an AP3012KTR boost converter with $\sim 80\%$ efficiency. To avoid damaging the battery, we first charge a capacitor bank $C = 1\text{mF}$ slowly (over 75 ms), and then discharge rapidly from this bank into the coil. We choose ceramic capacitors to

provide a low R_{ESR} . There are four $22\mu\text{F}$ capacitors in parallel placed next to each SEP. Our circuit design is modular, such that the capacitor bank can be discharged into any combination of SEPs simultaneously depending on the actuation sequence desired (**Figure 2B**). The maximum charge, Q , that can be sourced from the bank is given by: $Q = CV = 27.5\text{mC}$. We used this circuit to help us find the remaining parameters of the SEPs experimentally.

Knowing the magnet material and dimensions as well as the available power, we next focus on the coil. Specifically, the achievable SEP pull force is directly dependent on the amount of current we can push through the coil, which in turn is dependent on the number of coil turns (or inductance) and the resistance in the coil:

$$I = V/R_{ESR}(1 - e^{-tR_{ESR}/L}) \quad (1)$$

where V is the SEP supply voltage, R_{ESR} is the series resistance in the supply R_C , plus that in the coil R_L , L is the inductance of the coil, and t is the time since the charge started. Thicker, longer wires however produce diminishing returns due to (1) the maximum steady state magnetization strength of the Alnico rod, (2) the limited power available, and (3) the fact that the copper adds to the weight of the module which in turn increases the necessary pull force to produce motion.

Through a number of experiments to evaluate weight vs. magnet strength, we decided to settle for 40 AWE (American Wire Gauge) copper wire. **Figure 3A** shows how the number of turns with this wire affects the SEP pull force. F_{max} was measured between an SEP charged with the circuit described above and a steel bar, using a micro load cell rated 0–780 g from Fidgets Inc. As expected, F_{max} increases with an increasing number of turns, until R_L starts to limit I . With 100 turns, we found $F_{max} = 1.11 \pm 0.15\text{N}$. We then measured how the pull force was affected by the number of times an SEP (40 AWE, 100 turns) was charged after being fully polarized in the opposite direction, shown in **Figure 3B**. We found that the SEP reaches maximum pull force after being charged approximately 5 times. These SEPs weigh 0.95 g, with the coil and end caps contributing 0.15 and 0.30 g, respectively. With 12 SEPs located around the perimeter of a 46 mm diameter module, the weight of a full module is around 20.9 g without a battery and 25.4 g with one; i.e., the 12 SEPs make up approximately 45% of the full module weight. Based on the experiments above, we use five consecutive capacitor bank charges to flip the polarity of a SEP and 1 to simply turn it off. As part of future work, we hope to perform a model-based analysis to find more optimal parameters, with the aim of increasing the SEP strength, while decreasing the total module weight.

3.2. Actuation

To move the modules, we make two assumptions: First, the moving module moves only itself. Second, the module it is moving around is connected to many other modules keeping it relatively stationary. To move on a neighboring module, the moving module has to first inform the other about its desired move and polarity. If the module is transitional between two modules it needs to ask its neighbor to pass the message along to the following module. We target several types of motion

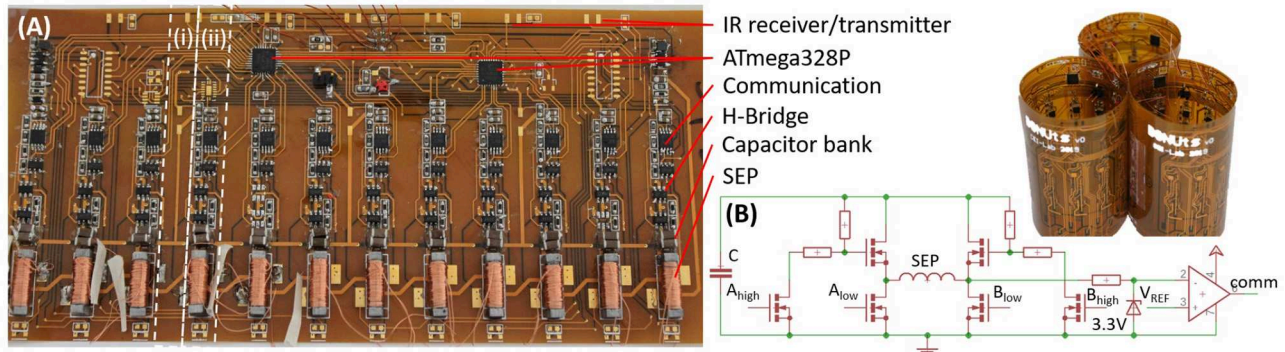


FIGURE 2 | (A) A module PCB (unwrapped) and three modules (wrapped). The white dashed lines indicate (i) flexible and (ii) rigid regions due to the placement of components; IR sensors and amplifiers are not mounted in this photo, similarly, a few external wires appear only for initial debugging and powering purposes. **(B)** SEP driver and communication circuit.

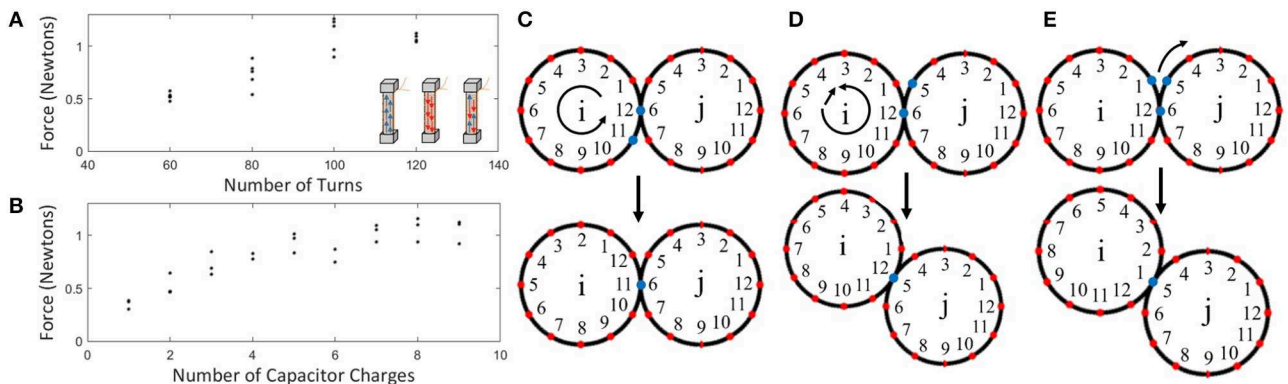


FIGURE 3 | (A) SEP pull force when placed against a steel bar vs. the number of coil turns (five trials each). All measurements were taken by polarizing the SEP in one orientation with a strong magnet, then depositing a constant charge to flip the polarization. The insets show how the SEPs work conceptually, by changing the orientation of dipoles in the Alnico-material. **(B)** Pull force for an SEP with 100 turns, 40 AWG wire, and 4 min end caps. Each measurement was done by first orienting all dipoles in one direction (using 11 charges), and then depositing a number of charges in the opposite direction. **(C–E)** Conceptual movement by module *i* (red illustrates the position of the SEPs, blue those related to the move). **(C)** Rotating-motion, i.e., counterclockwise rotation of *i*. **(D)** Rotating-translating motion, i.e., counterclockwise rotation and translation by *i* to the adjacent SEP on *j*. **(E)** Gear-like motion, i.e., clockwise rotation by *i* along the perimeter of *j*.

including on-axis rotation, rotation-translation, and gear-like rotation, as shown in **Figures 3C–E**, respectively. We imagine that the latter two modes are useful for general motion, and that the former is of use if a particular module sensor is broken, or if the collective wants to take more measurements from slightly different angles. We anticipate that a combination of these motion abilities will support system-level robustness.

We found that rotation around the module axis is possible through the following sequence: (1) [S-O-N-N; N-S-S-N], (2) [S-O-O-N; N-S-S-N], (3) [S-N-O-N; N-S-S-N]; and (4) [S-N-S-N; N-S-S-N], where N, S, and O corresponds to north, south, and off, respectively. When enabling these types of motions we used a total of 11 capacitor bank charges to make sure that an SEP was polarized to the desired state. We further found that rotation-translation is feasible by conducting the following sequence of polarity switches: (1) [X-N-S-X; X-S-S-X], (2) [X-N-N-X; X-S-S-X], (3) [X-S-N-X; X-S-S-X], where X corresponds

to any state. In step 1, the modules are connected between locations 2 in the array, and in step 3 between locations three in the array. This type of motion requires a total of 10 capacitor bank charges. We conducted a reliability test of the translation motion, and found that the module was able to successfully move 48 out of 50 times when no external forces were applied (**Figure 4A**). It should be noted that in these two experiments we used external power for experimental ease, but both tests were performed with the weight of a Li-Po battery on-board. It is also important to note that these moves require only one module to activate its SEPs. Therefore, although modules must first agree on the upcoming move with their neighbor, a movement does not require synchronous behavior.

Finally, we found that the current hardware only facilitates gear-like rotation in two scenarios: either when external compressive loads are applied as shown in **Figure 4B**, or when approximately half the weight is removed from the modules.

More generally, we found that friction is an important factor in determining the motion that a module is capable of and that it is dominated by acceleration due to fast SEP switching. The acceleration, in turn, is determined by module inertia. According to Seiner's theorem, the inertia required for the module to move about a point on its perimeter is: $I = I_o + mr^2$, where I_o is the inertia for the module to rotate about its own axis, and m and r are the module mass and radius, respectively. Therefore, if given the chance, the module will spin around its own axis, rather than travel along the perimeter of another. In future work, we hope to enable gear-like motion via the following approaches: (1) an in-depth study and optimization of SEP and module parameters, (2) synchronized SEP switching in neighboring modules, and (3) addition of friction tape along the module perimeters to make on-axis rotation more energy consuming than translational motion.

3.3. Passive Deformation

The passive module deformation is useful both to enable large-scale configurations beyond what manufacturing tolerances would allow with rigid modules, and to permit modules to interact safely with each other modules and their environment. For completeness, we here characterize the deformation modules are capable of.

The components on the flexible PCB are spaced to produce rigid zones, and flexible zones in between the SEPs (**Figure 2A**). This means that when a static external load is applied to a DONUt module, it deforms by an amount proportional to the load. Beyond guiding the magnetic field, the SEP end caps also function as a mechanical stop which prevents pinching that could permanently deform the PCB. Therefore, when the load is released, the module reverts back to its original shape, exhibiting spring-like behavior. The effective spring constant of a DONUt module was experimentally obtained by applying increment amounts of weight on a flat surface lying on top of a sideways module. The constant is calculated from Hooke's law: $F = -k_s \Delta x$. The term k_s refers to the effective spring constant and Δx is the change in length when a force, F , is applied. We found $k_s = 28.01 \pm 2.85 \text{ N/m}$ (**Figure 4C**).

The looped PCB, of course, does not behave like a perfect spring, and the change in deformation between increment weights decreases slightly with increasing load. This is due to an increasing effect of the rigid zones on the deformation of the module as they are pressed closer to each other at the right- and left-most edges of the module, corresponding to the areas of highest curvature. It should be noted that if the load was dynamic with non-negligible momentum, impact, or vibrations, then the geometric response of the module would be quite different, and it is possible that this effect may be exploited in future work.

3.4. Computation

Our choice of controller, ATmega328, coincides with those of the Arduous platforms which are very popular in the do-it-yourself community, again aligning with our philosophy of lowering the barriers to entry for diverse researchers and developers to help increase system robustness. To provide a sufficient number of control pins, each DONUt module has two ATmega328 microprocessors running on their internal 8 MHz RC oscillator,

with 2 KB SRAM and 32 KB EEPROM. The first processor controls SEPs 1–7 and three IR transceivers; the second, SEPs 8–12, one IR transceiver, and all SEP communication channels. The two processors communicate via UART. The software for low-level control of all peripherals take up just 2.3% of the SRAM and 6.5% of the EEPROM, leaving the majority of static and dynamic memory for the controllers described in section 5.

3.5. Communication

As previously mentioned, we simplify the design by making double use of the SEPs for actuation and module-to-module communication. Restricting the communication range is a commonly used method to avoid bandwidth problems as many asynchronous modules try to communicate (Rubenstein et al., 2014). When two SEPs located on separate modules are in contact, they can communicate locally as follows. The capacitor bank is first charged to maximum capacity. Bits are then transmitted using electromagnetic induction; i.e., the transmitter encodes bits in pulses of current, which are received by the neighboring SEP via induced current in the coil. The receiver then decodes these (weaker) pulses into bits. The current communication protocol is able to send a packet of 4B at a rate of 5 kbps on a single capacitor bank charge.

We developed our own protocol to facilitate communication with bits encoded in pulse length: A “1” is approximately twice the length of a “0” (**Figure 5A** top). This encoding simplifies synchronization because we can treat any bit like a clock signal, and use a simple schmitt-trigger coupled to a timer input comparator on the processor to decode the package. A transmission is started with a “1,” and bits are sent from least to most significant. The main limitation in baud rate is the time it takes to charge the capacitor bank. **Figure 5A** bottom shows the decrease in transmission voltage as a (worst case) package of all “1”s is sent, and the capacitor bank discharges.

To test communication reliability, we cycled through a transmission of all possible characters between two SEPs. We found the error rate to be 1 flipped bit per 1,000 bits. This issue may be addressed by adding in one or more parity bits for a slight decrease in throughput.

3.6. Sensing

Sensors allow the modules to interact intelligently with their environment. Although we focus on simple IR sensors for gradient tracking and obstacle detection, it is relatively easy to modify the module design to fit different sensors because it only involves a slight re-routing of the PCB.

Currently, each module is equipped with four infrared emitters (OP140A) and four receivers (LTR-301) operating at 935–940 nm (**Figure 5B**). For full spatial coverage while keeping the number of components small, these eight components are spaced equally around the perimeter of the module, and have a radial emission angle of 40° and a relative sensitivity around 20° , respectively. The outputs from the receivers are multiplexed into the analog to digital converters (ADC) on the processors. To measure the distance to an object for instance, we turn on the relevant emitter and multiplexed channel, and subsequently read the ADC value. We experimentally tested the distance sensors

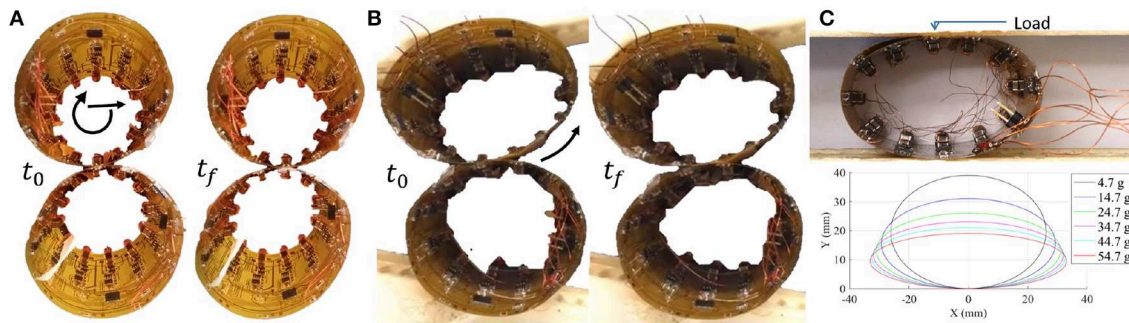


FIGURE 4 | (A) Module moving about another module, with motions as described in **Figure 3C**. **(B)** Motion of compressed modules, with rotation similar to that described in **Figure 3D**. **(C)** Characterization of module deformation, conducted by placing weights on a module.

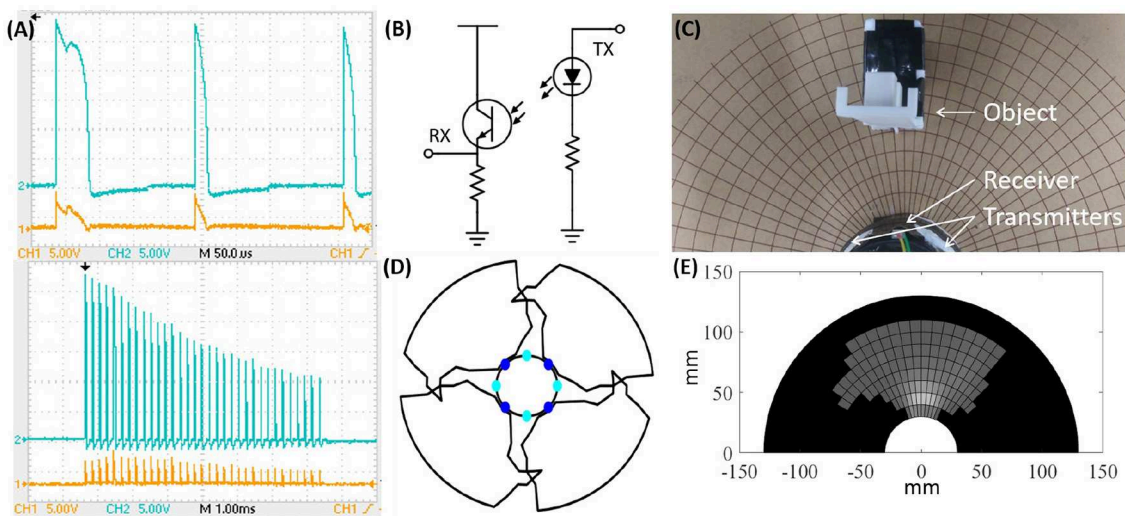


FIGURE 5 | (A) Top: Communication packet between SEPs. Received (orange) and transmitted (blue) message of a start bit, “1,” followed by two “0” s. Bottom: Decrease in voltage over the capacitor bank as a package of all “1” s is sent. **(B)** IR transceiver circuit. **(C)** Experimental setup to measure module field of view. **(D)** Top view of the experimentally measured sensor coverage in a module. **(E)** IR intensity map (bright values correspond to close objects and reach a maximum of 498 bits; dark values correspond to 0 and no measured signal).

using the setup in **Figure 5C**. **Figure 5D** shows a top view of the module coverage pattern and **Figure 5E** bottom shows the raw values from the sensor. Although every module will have a slightly different coverage dependent on the manual mounting of the sensors, even small objects should be visible before contact. In Ceron et al. (2019b), we discuss how to use this ability for object shape estimation.

3.7. Power

A DONUT module can fit up to three single cell 0.15 Ah Li-Po batteries from E-flite, weighing 4.5 g and measuring $45 \times 12 \times 8$ mm each. The module has the ability to measure its own battery level to support more intelligent collective behaviors as further discussed in section 5. The vast majority of energy spent in a module is on actuation and communication. As a rough estimate, a single battery should be able to support $E_{batt}/(0.5CV^2) = 6,000$ capacitor bank charges. Given that a single gear-like move

requires 11 capacitor bank charges, this corresponds to a full travel length of 12.9 m or 280 module diameters (with no communication). Beyond improved movement, future work will target integration of solar cells to support longer term operation.

3.8. Scalability

As argued in the introduction, a focus on individual module robustness supports large scale robot collectives, which in turn enables system-level robustness. Here, we discuss the current state of the modules in terms of cost, fabrication time, and maintenance, and how these may be improved to make large scale DONUT collectives feasible.

3.8.1. Cost

As we have yet to optimize for cost, a single module is priced around 587 USD. The biggest cost stems from the two-layer flexible PCB (468 USD quote from Advanced PCB), the 48

ceramic capacitors (46 USD), the 12 MOSFET drivers (17 USD), and the 12 SEPs (12 USD). The remaining 44 USD stems from components, such as processors, LEDs, resistors, etc.

There are several ways to lower the price of low-volume module fabrication. The cost can be reduced drastically by picking a cheaper PCB manufacturer (the lowest quote was just 90 USD, but had a longer lead time), or by taking advantage of the recent progress in Inkjet printable flexible PCB (Kawahara et al., 2013). The latter reports a drop in price to 10 USD per meter of film, which would leave the cost of the PCB to be negligible compared to the other parts in the module. There are also cheaper alternatives to the current capacitor banks; we could, e.g., use fewer, but larger OSCON capacitors similar to those used for flash in cameras. To give an idea of how the price scales with mass fabrication, the price of the current (non-optimized) component list drops from 119 to 52 USD/module when ordering for 1,000 modules. We aim to produce a second version of these modules with a price point around 50 USD, placing them in a similar range to the cheaper modular robots in literature (Brunete et al., 2017).

3.8.2. Fabrication

One of the key benefits of the DONUt module design, is the reliance on a single PCB which supports imprecise, rapid, and inexpensive manual assembly of both SEPs and wrapped PCB. The largest time sink for the fabrication is component soldering; currently, one PCB takes around 5 h to solder by hand. In the future, we hope to have the majority of the PCB pre-populated at a manufacturing house. To get a rough estimate of how this would trade off cost for lowered assembly time, we requested a quote from Advanced PCB which came to 30 USD/module for an order of 1,000 modules. We expect that this cost can be lowered with a more thorough search of vendors, a longer requested lead time, and the right choice of components. The current capacitor bank, for example, consists of many components in parallel; it would be beneficial to replace these with a few, larger capacitors.

If the PCB assembly is outsourced, that leaves the following steps for in-house assembly: (1) SEP manufacturing, (2) attachment of SEPs and batteries, and (3) flexing the PCB into a loop. Of these three, only the first two take any considerable amount of time. The process is as follows. First, the magnet is glued to the steel end caps with super glue; then the assembly is inserted into our winding rig shown in **Figure 6**. The gears in the rig are dimensioned such that a single turn of the red wheel by hand adds 100 turns to the coil. This entire process, including PCB mounting, takes at most 4 min per SEP, i.e., 48 min per module.

3.8.3. Maintenance

The maintenance requirements of modular robots stem from mechanical wear, the ability of a user to operate (start, stop, and program) all modules with a global command, the module battery life time, and the reliability of individual components. We address each factor in sequence. (1) DONUt modules have no internally moving mechanical parts that can wear with use, and have no loose wires or connectors that may break over time which tend to be one of the bigger problems in small electro-mechanical devices. (2) In future versions, we may explore better

parallel operation, enabling user control through a single IR source similar to past platforms (Rubenstein et al., 2014). (3) In the future we may optimize the maximum possible travel distance per module through integration of solar panels on the PCB. Although this type of power harvesting will be slow, it fits this particular style of robots well: only perimeter modules in large collectives are able to move which causes a spiraling migration pattern where the majority of modules at any one point in time remain stationary (further discussed in section 5). (4) Although more thorough tests are needed, we tested 50 moves in a row without any component faults.

4. SIMULATION ENVIRONMENT

We have developed an open-source simulation platform in Matlab® to support general access to development and testing of control schemes for the DONUtS modular self-reconfigurable robot (https://github.com/njw68/DONUtS_Simulation). The framework permits programmers to easily test large numbers of modules operating in varying degrees of clutter, and perform structured analysis of system resilience to signal noise and component failures. The simulation incorporates gear-like motion (**Figure 3D**), connections, sensing range, and message passing. Module compliance, friction, and inter-module forces are not integrated at present. The software is written such that the user can focus on implementation of high-level control schemes, while lower-level primitives like those needed to identify obstacles, connections, and viable motions are abstracted away. An architecture overview is shown in **Figure 7**.

A programmer can experiment with path planning in cluttered environments with their choice of the number of modules and the amount and complexity of the clutter. The simulation framework may be easily modified to support other task settings and distributed algorithms as well, similar to how we used it in Ceron et al. (2019b). Upon initialization, the programmer may specify the number of modules, the target location, and either pre-determined or randomly generated obstacles with a user-specified size. The software can generate either a rectangular or a random configuration of interconnected modules; it can also run a random initial configuration, where each of the aforementioned variables is randomly generated.

4.1. Module Primitives

Next, we introduce several low-level behaviors to support operation of the DONUtS.

4.1.1. Motion Restrictions

To determine whether module i can physically move, we make three successive checks related to the following properties:

1. i is connected to at least one other module, such that at least one c_{ij} exists, where c_{ij} is the set of connection positions on i between i and j . j refers to the set of modules that are connected to module i and $c_{ij} \in [1, \dots, 12]$.
2. Connections to other modules are contained within 180° , i.e., the module has five consecutive free connections.
3. Movement will not disrupt global connectivity of the modules.

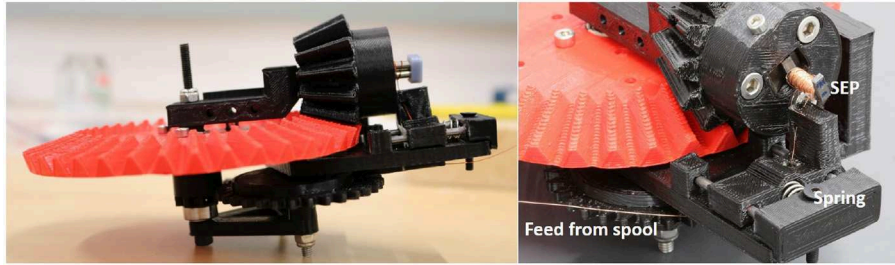
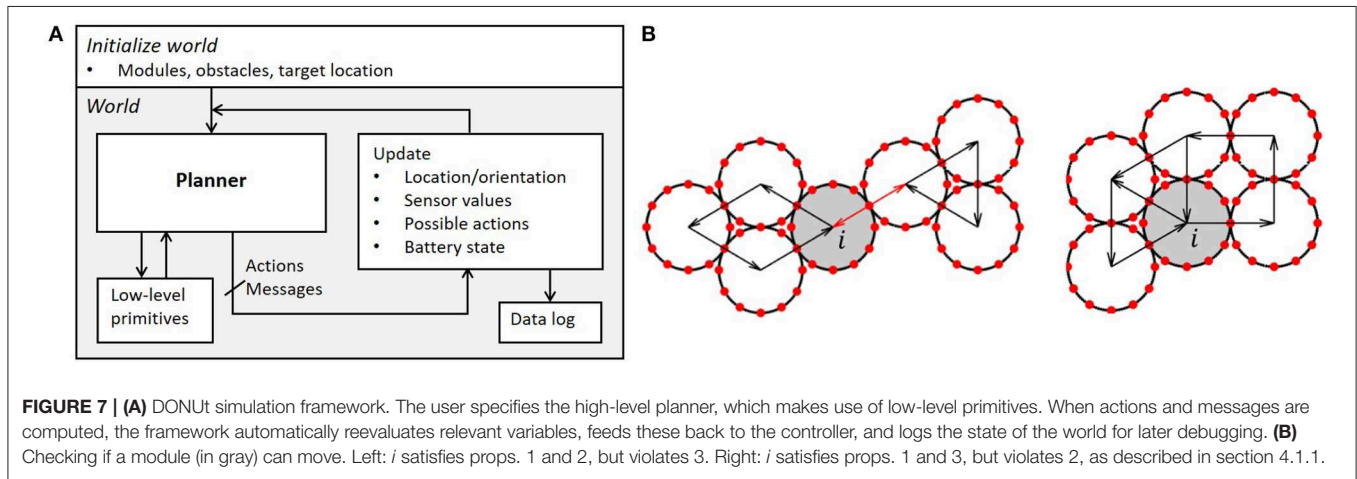


FIGURE 6 | 3D printed, manual winding device for rapid production of SEPs. The SEP is mounted into the black piece with a screw; the spring keeps the wire from a spool taught during winding, similar to the mechanism on a sewing machine. A full turn of the red wheel adds roughly 100 turns to the SEP.



The first property relates to the fact that modules cannot move on their own; the second to the fact that they need physical clearance to move; the third check ensures a cohesive collective (**Figure 7B**). The latter is done by checking for loops in the connectivity graph; i.e., we pass a message to all neighboring modules to see if it can loop back to the origin without passing the same edge twice. After verifying these properties, we compute the possible movements [clockwise (CW)/counterclockwise (CCW)/both] taking into account the presence of other modules and obstacles in the environment.

4.1.2. Motion

To physically move a module, it must pass a message to the neighbor which it is rotating about to prepare the next connection (i.e., switch on the correct magnet with the correct polarity). The attraction of the two successive magnets, alongside the repulsion from the previous connection point will propel the module forward. The geometric movement of each module is a function of the center of the module about which they are moving (**Figure 3B**). We can compute the center position (x, y) of a moving module by Equation (2):

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \end{bmatrix} + 2R \begin{bmatrix} \cos(\theta_j + \frac{2\pi}{12}(c_{ji} + u)) \\ \sin(\theta_j + \frac{2\pi}{12}(c_{ji} + u)) \end{bmatrix} \quad (2)$$

The terms i and j refer to two adjacent modules; module i moves about the perimeter of stationary module j . R is the module radius, θ_j is the orientation of j with respect to the world reference frame, and c_{ji} is the magnet position of the connection between j and i on j , where $c_{ji} \in [1, \dots, 12]$. The term u is the control input for i which determines whether i will move CW or CCW about j 's reference frame, $u \in [-1, 0, 1]$. When $u = -1$, i moves CCW about j ; when $u = 1$, i moves CW about j ; and when $u = 0$, i remains static at its current location.

To keep track of modules and their orientations, we allocate specific IDs to every magnet on the perimeter, and map these to relative IDs as they rotate. An array stores the position of the magnet ID with respect to the inertial frame of reference. Initially, all modules have magnets mapped one to one, such that magnet 1 is at position 1 ($c_1 = 1$), magnet 2 is at position 2 ($c_2 = 2$), etc. When a module moves CCW about another, the moving module's magnet positions are updated by -1 , such that $c_k = c_k - 1$, $k \in \{1, \dots, 12\}$. Similarly, CW movement results in updates by $+1$. A check ensures proper rollover when surpassing 1 and 12. The software updates all magnet positions, c_k , through the module's control input, u :

$$c_k(t+1) = (c_k(t) + u) \quad (3)$$

After the movement has occurred, the module may find itself near new neighbors. To determine the presence of

such neighboring modules, a module will briefly activate all connection points, transmit its ID, and await an acknowledge message. In general this sequence needs to be performed only by a module after movement. However, it is possible that occasional checks by all modules to verify their connectivity will improve system robustness.

5. MODULE COORDINATION

Translating algorithms developed in simulation to real hardware often requires considerable effort. However, such simplified simulations may still be used to quickly iterate the overall coordination methodology as well as to illuminate non-intuitive pitfalls related to the hardware design. In this section, we discuss important findings related to robust coordination of many DONUt modules for gradient tracking, introduced by the hardware-specific constraints. Gradient tracking is a robust and potentially scalable basic behavior necessary for navigation in coordinate-free environments. This behavior may support applications such as identifying the source of chemical spills or simply navigating toward a source of light to harvest power. We introduce controllers for gradient tracking in clutter-free and cluttered environments, using the available sensors described in section 3 and building on the simulation framework and the low level primitives introduced in section 4.

Specifically, we introduce two types of controllers toward robust collective behavior: an A* search-based controller and a more naive, iterative controller. We discuss implementation details, and compare these in terms of complexity and optimality with respect to the number of module moves which directly impacts energy efficiency and maintenance. To produce a benchmark for “optimal behavior,” we also introduce an Oracle planner with complete knowledge of the world. Beyond control methodologies, we discover and discuss a type of connection

topology that generally impedes progress along the gradient, and discuss how to avoid this with the naive controller.

Intuitively, sophisticated controllers should not be necessary for gradient tracking as every agent in the collective can simply navigate according to the local gradient. Here, however, we target controllers that advance the entire collective efficiently toward the gradient source. Note that, because (1) we enforce a globally connected collective, (2) modules cannot move on their own, and (3) only perimeter modules are capable of moving, this is not a simple problem. Were we, for example, to perform a naive graph-search across all possible moves of a state in which ten modules are configured in two adjoining rows, this state would have twenty children for a single module move. In other words, the search space quickly becomes intractably large.

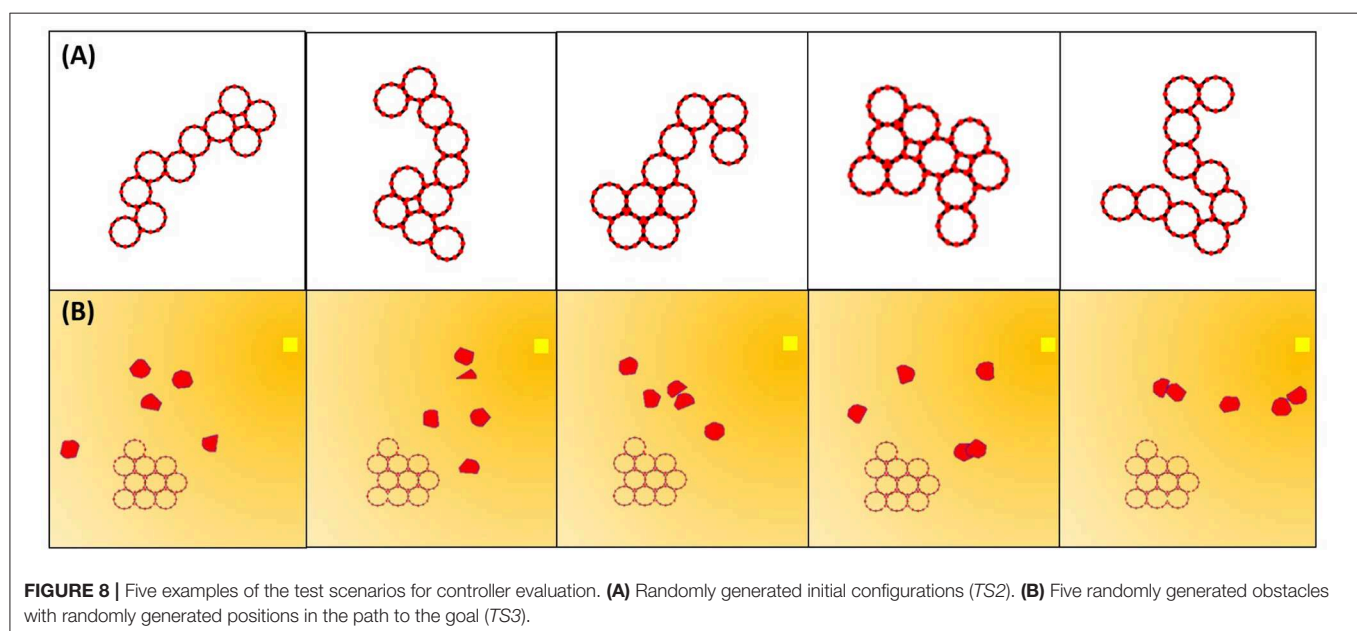
To evaluate our controllers, we use different subsets of the following three scenarios:

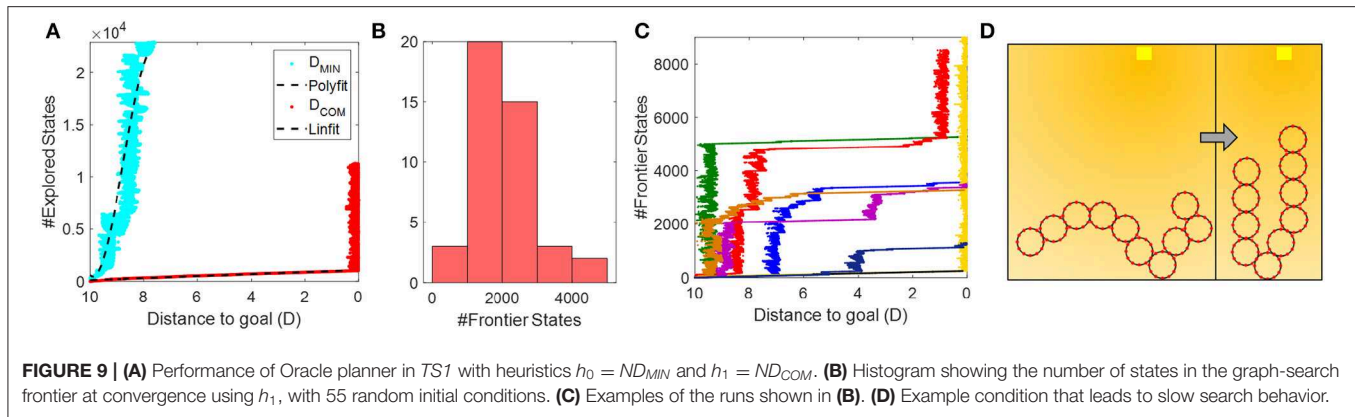
- *Test Scenario 1 (TS1)*: 10 modules starting in a cluster 10 module diameters (20R) from the goal in a clutter-free environment.
- *Test Scenario 2 (TS2)*: 10 modules starting from 10 random configurations 20R from the goal in a clutter-free environment (**Figure 8A**).
- *Test Scenario 3 (TS3)*: 10 modules starting in a cluster 20R from the goal in an environment with five randomly generated and randomly placed obstacles (**Figure 8B**).

Note that, unless otherwise noted, we abort runs which exceed ~7,000 states; furthermore, we limit the scope to sequential module movement.

5.1. Oracle Path Planning

To provide a baseline against which our centralized controllers can be compared, we implement an Oracle planner that computes an optimal path in terms of module moves to a global light source,





given complete knowledge of its environment. This planner is based on A* graph-search, where the nodes in the graph correspond to the state of the robot (i.e., the location of all DONUt modules) and the edges correspond to module moves. The cost of a node, $cost_{state}$, is calculated as the number of moves it takes to get to that state. To search the state space efficiently, we compute an admissible search heuristic, h , expressed in module moves, and prioritize nodes with lower total cost: $cost_{total} = cost_{state} + h$. The combination of graph-search and an admissible heuristic allows us to prune the large search space and return globally optimal results (Russell and Norvig, 2016). To ensure that the modules cluster around the light source, we complete the search when the distance from the collective center of mass (COM) to the goal is within two module radii, $2R$.

We examined two heuristics in terms of search space efficiency. The first heuristic, h_0 , is based on the intuition that it is beneficial for the collective to quickly align their orientation with the highest gradient. To do this we make h_0 a function of the euclidean distance between the module with the highest measured light intensity (corresponding to the lowest distance to the goal, D_{min}). We compute the number of moves it would take one module to travel this distance and multiply by the total number of modules in the collective, N , i.e., $h_0 = ND_{min}$. The second heuristic, h_1 , is more generally based on the intuition that all modules need to move toward the goal. We make h_1 dependent on the euclidean distance between the collective's COM and the goal ($h_1 = ND_{COM}$). Because modules actually have to travel around the perimeter of other modules to reach the goal, the straight line distance is an underestimate of the true distance and results in an admissible heuristic.

We found that h_1 far out-competes h_0 in terms of search space efficiency. An example of what happens is shown in **Figure 9A**; the number of expanded states in the tree grows exponentially with h_0 , and closer to linear with h_1 . The intuition behind h_0 's performance is that once a module has been moved as close as possible to the goal within a configuration, all other moves have an equal cost. That is, when a new module moves closer to the goal, but not enough to surpass the current closest module, D_{min} is the same as the scenario when that module moves away from the goal. In contrast, h_1 ensures that until the collective reaches the goal, we favor states that directly impact the progress of the entire collective. Once the COM gets within $2R$ of the goal, the

frontier grows rapidly simply because the heuristic no longer supports closer clustering.

To reason about how well the h_1 -heuristic worked, we calculated the effective branching factor, b^* in *TS1*. Briefly explained the effective branching factor denotes how many branches every node would have on average if the solution was recast as a breadth first search ($n = b^0 + b^1 + b^2 + \dots + b^a$, where n is the number of states, b is the number of branches, and a is the depth of the search tree). A b^* close to 1 indicates that we almost always guess the optimal move and therefore keep the search tree from branching excessively. We found that h_1 is a very efficient guess, with an average $b^* = 1.034 \pm 0.0054$, confirming that moving the module that advances the collective COM as much as possible toward the goal is preferred. Note, that this result does not necessarily translate to cluttered environments or take into account the fact that extra connections between modules may improve their redundancy in case of failures.

Interestingly, we found that the search efficiency was heavily dependent on the initial configuration of the collective. To examine this phenomenon more, we ran 55 iterations of *TS2* and plotted the maximum number of states reached in the frontier before convergence. The results are shown in **Figure 9B**. Forty-seven out of 55 trials converged within the allotted number of expanded states. The fastest searches converged after evaluating about 500 states, but the majority required 2–4 times as many evaluated states. **Figures 9C,D** illustrates why this occurred. We see that the number of states in the search frontier grows linearly over most of the path, but exhibits periods of exponential growth. These periods occur when the search reaches a state where the collective forms a single chain with the center point of the chain is closest to the goal. In this state, moving either of the two end-modules forward is the fastest predicted way to reach the goal, eventually leading the search to a state in which the collective forms a U-shaped chain. Once this state is found, the search must explore all other higher cost states before it again finds a move that will bring the COM closer than when it was in the U-shape. We designed the following on-board controllers with this risk in mind, to support faster convergence.

5.2. A* Search-Based Controller

In a realistic scenario, the modules will not have access to the state of the world and must plan according to what they know;

i.e., their connection topology, the measured light intensity, and nearby obstacles. We first implement an A* search-based controller for the modules similar to the Oracle planner with two exceptions. First, instead of planning toward the actual goal, we choose an intermediate goal location which corresponds to the module with the highest measured light intensity, i.e., the module closest to the goal. The modules plan their path to the temporary goal, execute this path, recalculate the new temporary goal, and re-plan. Second, based on the discussion of the effective branching factor, we calculate a new admissible heuristic based on the distance to the intermediate goal from the collective's COM, $h_2 = CND_{COM}$, where C is a constant scaling factor. Effectively, this means that the collective moves in stages (each locally optimal): first communicating and identifying which module is closest to the goal, then planning how to bring their COM to that location before reevaluating which module is now closer. They repeat this process until the collective COM is within $2R$ from the light source.

To find the optimal value of C , we did a parametric sweep from $C = 0.1 - 1.0$. We did this sweep with a square initial configuration (*TS1*), and found that $C \leq 0.5$ performed very poorly, rarely making progress beyond $2R$ toward the goal in the allotted number of states. Conversely, $C > 0.5$ yielded results that were too similar to draw a conclusion. We therefore ran an additional sweep from 0.6 to 1, using 10 configurations in *TS2*. The results are shown in **Figure 10A**. We found that coefficients at the extremes (0.6 and 1) rarely converged in the allotted number of expanded states. The best results were with $C = 0.7$, which converged in all cases and reached a collective COM within $4R$ and $2R$ from the goal more quickly than when using other values for C . Note that with $C = 0.7$, the heuristic is still admissible.

Based on these simulations, we further make the observation that the collective may enter live lock, i.e., an infinitely repeated movement pattern, before reaching the goal depending on the collective's angle to the gradient. An example of what causes this is shown in **Figure 10C**; modules 10 and 5 oscillate back and forth leaving the collective within the temporary goal, but not the global goal. Because the collective has no memory from previous planning iterations these movement patterns will execute forever. To overcome this, we added a check to assess how much progress the collective has made within one planning iteration. If the collective converge on the temporary goal after moving just one module, no multi-iteration progress occurs. In this case, we move a random module (excluding the last moved module) 1 step. By adding a degree of randomness, we avoid local minima like these, and ensure that the collective will eventually converge at the goal.

In **Figure 10B**, we next compare the performance of the A* search-based controller to the Oracle in *TS2*. As would be expected, the A* search-based controller is less than optimal. The performance especially degrades as the collective approaches the goal, because at this point the number of moves it takes to reconfigure the collective's COM to the temporary goal dominates the difference of which module is closer to the goal.

We further make the observation that the controller often generates chain-like configurations, where every module on average has only two neighbors. These are problematic because a

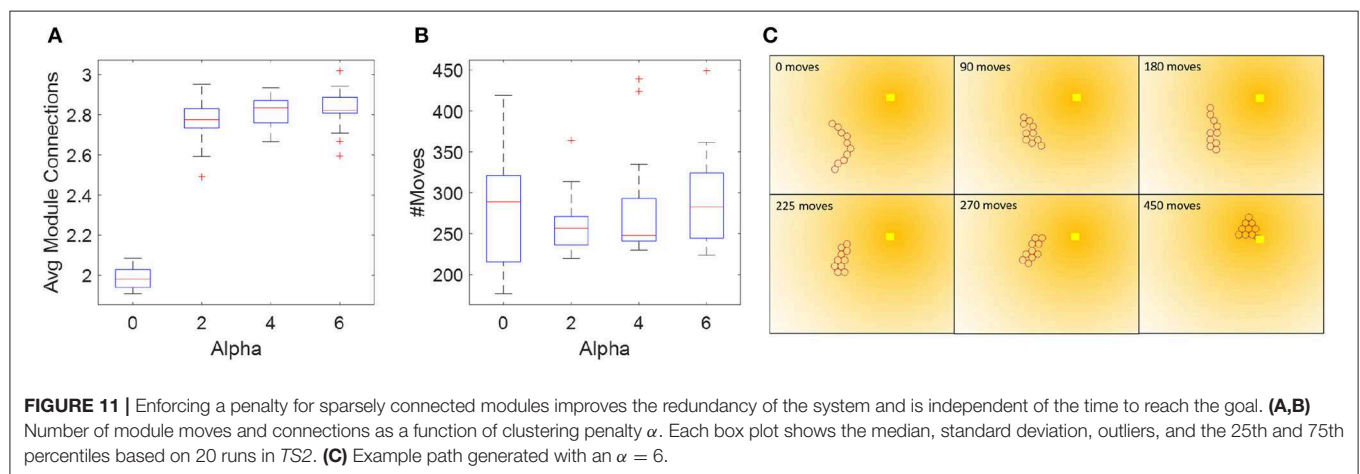
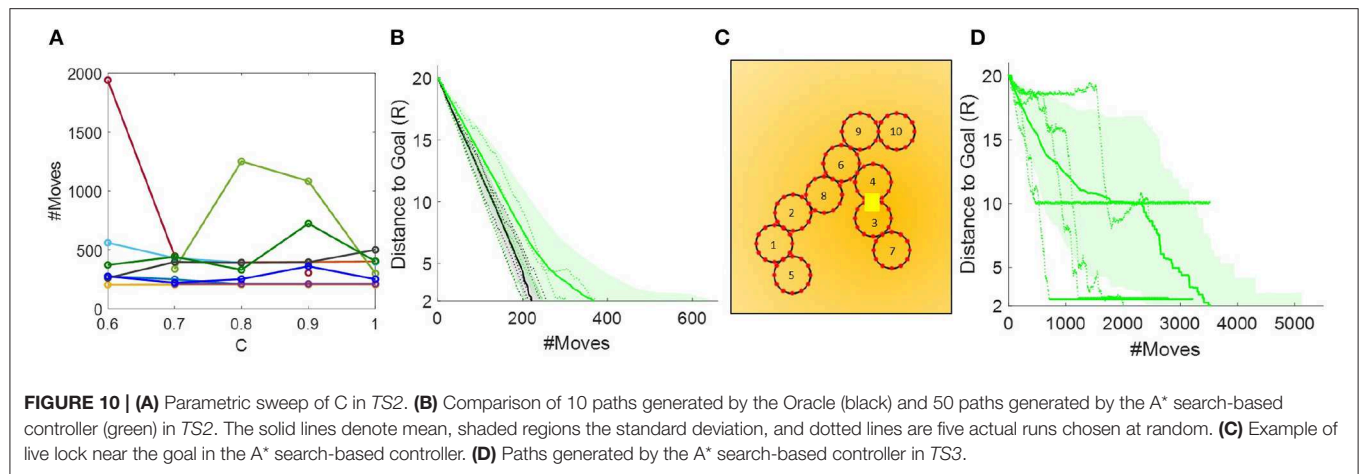
single module failure can split the chain in two disrupting global performance. The heuristic-based control approach permits a simple way to deal with this issue: we simply add a penalty for a loosely connected graph. Note that this effectively makes the heuristic inadmissible and results in (locally) sub-optimal, but (globally) more robust plans. We used a coefficient α to change the severity of the calculated penalty, P , where x is the number of connections a module possesses:

$$p = \begin{cases} x_i > 2, 0 \\ x_i = 2, 1 \\ x_i = 1, 2 \end{cases} \quad (4)$$

$$P = \sum_{i=1}^N p(x_i)\alpha \quad (5)$$

In other words, we add a penalty of 2α for modules that are configured in a chain and only have two neighbors, and a penalty of α for modules that are at the end of a chain. The new cost per node comes to: $cost_{total} = cost_{state} + h_2 + P$. We ran this simulation in *TS2* using $\alpha = [0 \ 2 \ 4 \ 6]$, and compared both the total number of moves needed for the collective's COM to reach the goal within $4R$ and the average number of connections in each step along the way (**Figures 11A,B**). Again, we see that the initial configuration has a big impact on performance, and that, as expected, with increasing penalty, the modules stay more clustered. The choice of α relates both to the desired redundancy and the number of modules in the collective. For example, with ten modules configured in a double-row the average number of connections per module corresponds to 3.4. We see that the graph levels out at $\alpha = 2$, i.e., at 2.8 connections per module which is reasonable given that some modules have to deviate from the double row for the collective to move. This experiment is a repeated measures, correlated samples test, thus we perform a one-way ANOVA for correlated samples and find that α has a statistically significant effect on the average number of module connections [$F_{(3,57)} = 663, p < 0.0001$]. Conversely, α does not have a statistically significant effect on the number of module moves [$F_{(3,57)} = 1.32, p = 0.28$]. The average number of moves between the α s vary by ~ 30 . To explain this, we examine the simulations at $\alpha = 0$ and $\alpha = [2 \ 4 \ 6]$. We find that while with a value of $C = 0.7$ and $\alpha = 0$ is an admissible heuristic for local optimization, it results in chain-like configurations, which are more susceptible to temporary live lock. These instances of temporary live lock require modules to move sub-optimally to break out of temporary live lock and resume regular planning. This causes an increase in the number of modules moves to reach the goal. In the cases of $\alpha = [2 \ 4]$, we observe that clustered configurations lead to sub-optimal local planning, but more robust global planning, thus fewer temporary live lock instances occur, and the average number of module moves is less than with $\alpha = 0$. An example of the path taken given $\alpha = 6$ is shown in **Figure 11C**. This brief study indicates that adding a clustering penalty is a viable and simple way to ensure higher collective redundancy.

Finally, we tested the A* search-based controller in cluttered environments (*TS3*). The results shown in **Figure 10D** indicate



live lock near obstacles. By studying the actual runs closely, we find that this happens because the number of modules which can move randomly is severely limited by either their connection topology (chain-like configuration) or their proximity to obstacles. We may deal with this by adding either a higher degree of randomness or memory between planner iterations. The former comes at the cost of planner efficiency and without guarantees that live lock can be avoided in all situations. The latter is complicated because the collective may enter configurations that appear similar to previous ones, but at geographically different locations. Modules may compute their trajectory to overcome this problem, however, this would require perfect dead reckoning skills which is not practically feasible with the hardware.

We can further discuss the ability of this planner to operate on the actual hardware processor, i.e., the 2 KB of RAM in the ATmega328P (Ceron et al., 2019a). The state space of the planner grows somewhere between linear and exponential with the number of modules, depending on (1) the optimality of the heuristic and (2) the configuration of the collective, i.e., the number of modules that are capable of movement. Every node in the search tree contains the collective's connection topology and a reference to the parent and child nodes. For 10 modules

in a perfect cluster this would correspond to 19 connections and 1 parent node, i.e., a memory footprint of 20B. In this state, 8 modules are capable of moving CW, CCW, or staying, therefore the node has 24 children; i.e., just two levels in, the search would take up 500 B of memory. Alternatively, we could trade off memory for computation by storing only the move and recomputing the configuration for every explored node. In this case we spend 20 B on the first node, and 1 B per node moving forward. With a good heuristic, the memory would then grow close to linear as in a depth-first search. To improve memory, we could further explore how this search could be distributed to over the two on-board processors (Colbrook and Smythe, 1990). Given the current hardware constraints, however, we are unlikely to be able to support planning for more than a few tens of modules. In the next section, we instead focus on more naive, iterative planners that require less memory and computation altogether and are inspired by what we learned from the graph-based controllers.

5.3. Naive, Iterative Controllers

To produce an algorithm that scales well in memory and computation with near-optimal control, we next examine a naive, iterative controller for gradient tracking. In this controller, we

simply prioritize moves of modules that are farthest from the light source, hence we name this type farthest-first or “faf-controllers.” As before, we identify the module with the highest light intensity, i.e., the one closest to the light source, and treat it as an intermediate goal. We then identify a movable module with the lowest brightness, and move it toward the intermediate goal along the shortest path around the collective perimeter. This process repeats until all modules are clustered around the goal location. We explore two versions of this controller. In the first, we move the darkest module one step before searching for a new darker module (“faf0”). Intuitively, this approach works well for highly dynamic environments where information quickly becomes stale. In more static environments, or when communication between modules or between modules and the centralized controller is costly, the update rate can be lowered by only re-planning when the moving module reaches its intermediate goal (“faf1”).

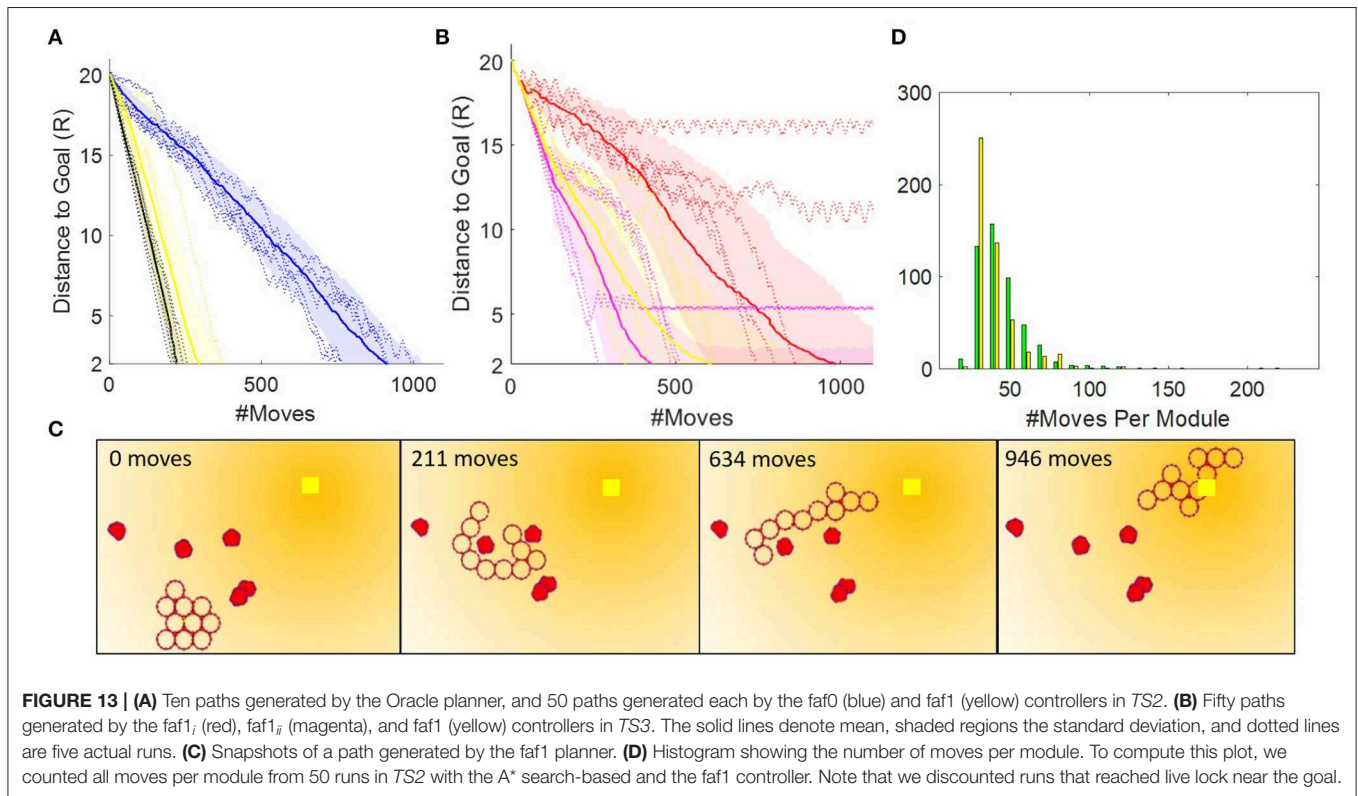
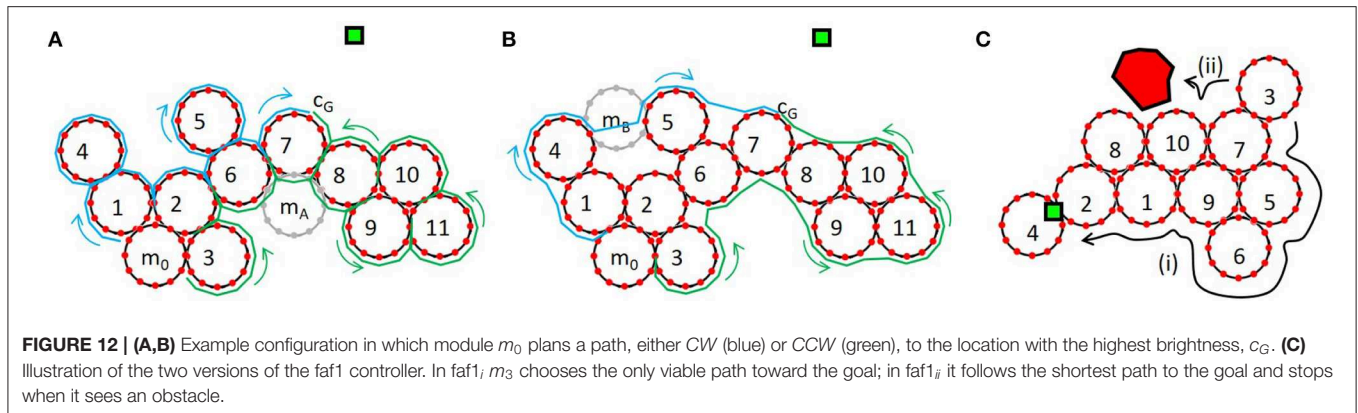
The following list details how this controller works, using the example shown in **Figure 12**.

1. *Initialize*: We first identify the intermediate goal, i.e., the brightest module, m_7 ; and the darkest module which is capable of moving, m_0 . We also identify which connection on m_7 is closest to the light source, c_G .
2. *Generate viable paths and identify external obstacles*: We next decide which direction (CW or CCW) around the collective will yield the most efficient path. We construct two candidate sequences, S_{CW} and S_{CCW} , along the perimeter modules from c_G to the closest connection on m_0 , c_0 . To construct S_{CCW} , we first append c_G ($S_{CCW} = [m_7(c_G)]$) and then continue appending connections on m_7 in a CCW direction until the neighboring module, m_6 , is reached ($S_{CCW} = [m_7(c_1 : c_G)]$). We continue this process until we reach the m_0 ($S_{CCW} = [m_1(c_1 : c_7)]...m_6(c_1 : c_4)m_7(c_1 : c_G)]$). These steps unambiguously define the sequence of connection points that m_0 must go through to reach the goal. In other words, by using connection points rather than module center points, we avoid confusing any connection on the closest module (e.g., m_A in **Figure 12A**) with the goal location. Every time we encounter a new module, we check for obstacles seen by that module to indicate whether or not the sequence is tenable.
3. *Optimize viable paths and identify geometrically incompatible modules*: The second step overestimates the number of moves needed to reach the goal, because it does not take into account locations like m_B illustrated in **Figure 12B**, where the module can connect directly between two non-neighboring modules, m_4 and m_5 . To prevent this overestimate, we loop through each tenable sequence, projecting m_0 along the sequence generated above, checking for physical proximity of connectors that would reveal new potential neighbors. By iterating forward through the sequence and for each projection checking backwards from the goal we ensure that m_0 always identifies the connection that is closest to the goal. If one exists, we delete all of the intermediate entries and produce new sequences, CW' and CCW' . If at any point the m_0 projection overlaps with another module we label the sequence untenable. Note again, that overlap is possible because we are not operating in a discrete occupancy grid. On the real hardware, overlap could also occur because of module deformation.
4. *Move module along shorter path*: Finally, we simply compare the length of the sequences and move the module in the direction of the shortest path.

Figure 13A shows the performance of faf0 and faf1 in *TS2*. Because the test is performed in a static, clutter-free environment faf1 outperforms faf0, here by a factor of ~ 4 . The oscillations in faf0 occur when the collective, similar to what we saw with the Oracle planner, finds itself in a U-shaped chain where it greedily moves the darkest module up the gradient at each cycle, effectively making the collective gather at one extreme of the connection topology, then the other, until it finally reaches the global goal. We see that faf1 performs almost as well as the Oracle planner, but that the performance is still dependent on the initial configuration.

For operation in cluttered environments, we explored three variations of faf1, also illustrated in **Figure 12C**. In faf1_i , if the shortest path which the darkest module must take to the intermediate goal is intercepted by an obstacle, we instead move the module in the opposite direction; if obstacles are detected in both directions, we move another module. In faf1_{ii} , when an obstacle is encountered we simply move the darkest module as close to the obstacle as possible. In faf1, when an obstacle is encountered, we choose according to faf1_i and faf1_{ii} with 50% likelihood, and, with 20% likelihood move a random movable module one step in a CW direction. **Figure 13B** compares the performance of these three variations in *TS3*. Generally, faf1_{ii} outperforms the others, however, it may enter livelock. Similar to our previous observations, we find that this happens in U-shaped chain configurations where the two ends point toward the goal and are near obstacles that hinder further movement. At this point each end module simply moves back and forth along the collective, without making actual progress. faf1_i does not show issues with live lock, but take nearly twice as long to reach the goal. faf1 overcomes issues with live lock due to stochasticity, at the cost of ~ 1.5 times more module moves. An example path generated by faf1 is shown in **Figure 13C**.

Deriving the exact scaling behavior for this planner is complicated due to the motion restrictions discussed in section 4.1.1. In the algorithm, most operations scale constant or linear with the number of modules; however, optimizing the path along the collective, i.e., step number 3 in the description above, approximates polynomial time. Intuitively explained, in the worst-case scenario where the collective is spaced out in a single file line and not in the presence of obstacles, the darkest module has to be projected along every other module to check for short-cuts. This step is an interesting point for future work. Another obvious direction for improving the scalability of this algorithm is to outsource computations. In Ceron et al. (2019b) we, for example, detail how the connection topology can be computed in a distributed manner. To extend the current planners to a completely distributed system, one can imagine combining these algorithms with a consensus-based scheme to identify the modules with highest and lowest brightness.



5.4. Discussion

In summary, in the context of gradient tracking in clutter-free environments, our 10-module simulations indicate that both controllers may perform nearly optimal despite the lack of global knowledge. The locally-optimal A* search-based controller performed well in terms of the number of modules moves for clutter-free environments, but additional measures must be taken to overcome potential live lock near obstacles. We also find that even with a good search heuristic, the algorithm scales poorly in terms of memory and will not support more than a few tens of modules if implemented on the two on-board ATmega328 processors. The naive, farthest-first controller performed equally well and had the ability to deal with live lock near obstacles via a small degree of

randomness. This controller is simple and may support more scalable behavior.

Through simulations, we further found that both types of controllers generally create chain-like, rather than clustered, configurations. Chain-like configurations are bad for the DONUTs, because (1) they severely limit the amount of modules that are capable of moving, (2) simulations show that chains often end up creating U-shaped configurations that impede general progress toward the goal, and (3) they leave the collective at risk of complete failure if just a single module breaks. We showed that encouraging redundant connection topologies in the A* search-based controller was fairly simple; encouraging these in the farthest-first controllers will be an important area of study in future work.

Finally, system energy consumption warrants explicit discussion because it is a major contributor to system autonomy and robustness, affecting the strategy of exploration vs. exploitation as the modules traverse an environment. The DONUt hardware, for example, was designed around SEPs which keep their polarization without continued supply of power, the number of power consuming components was minimized, and the modules were designed as light weight as possible (25.4 g). In **Figure 13D**, we compare the A* search-based and the fa1 controller in terms of how well they distribute energy consumption among the modules. As we have yet to consider energy spent on communication in our centralized controller, the energy we can estimate is directly correlated with the number of moves a module has to make. The plot shows that the fa1 controller inherently distributes power usage more evenly, whereas a few modules in the A* search-based controller moves many times further than the others. Evening out power consumption will also be an interesting future extension to our work.

6. CONCLUSION

In summary, we have introduced a new planar, modular, self-reconfigurable robot. Although more work is needed before practical large-scale demonstrations are feasible, this initial hardware-software design cycle has contributed several concepts that may translate to other platforms. Most importantly, by basing our design on a single flexible PCB without mechanically moving parts, we were able to achieve simple, fast manufacturing, and support low maintenance in terms of breakage and wear. By creating an open source simulation platform with realistic movement and sensing, we explored two control schemes and non-intuitive challenges that arose because of the module-specific motion constraints. We explicitly focused on enabling a large configuration space to enable operation in dynamic environments, and explored a range of challenges related to collective efficiency, scalability, redundancy, and adaptability. More generally, we showed that enabling scalability and system-level robustness, rely on tightly integrated design decisions that span fabrication, operation, and control with an explicit focus on constituent robustness.

We have several agendas moving ahead. On the hardware side, we will focus on decreasing cost, increasing battery life, and improving motion reliability before pursuing a large-

scale collective. So far, we have depended only on the passive compliance for added robustness, however, long term, we hope to investigate novel collective behaviors enabled by the compliance, including their ability to generate macroscopic materials with different density and tensile strength. Similarly, their spring-like properties promises interesting dynamic behaviors which may be leveraged for both communication and motion. Finally, the fact that every single module weighs only 25.4 g also indicates a new set of potential applications beyond those seen with previous platforms. On the control side, we are exploring several avenues. The most near-term is to combine centralized and decentralized algorithms for better scaling properties. Longer term, we hope to better investigate the trade-off between control redundancy and efficiency. A video description of this project can be found in **Supplementary Material**.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

All authors collaborated on the contents of this article. Hardware design and characterization was lead by SC and LH. Simulation environment by SC and NW. Path planners by NW.

FUNDING

This work was supported by the Cornell University SLOAN fellowship, the GETTYLAB, and the Packard Fellowship for Science and Engineering.

ACKNOWLEDGMENTS

The authors would like to thank Claire Chen, Daniel Kim, Nick Parker, and Dr. Kevin O'Brien for their help in the early stages of the hardware design.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2020.00044/full#supplementary-material>

REFERENCES

- Bhat, P., Kuffner, J., Goldstein, S., and Srinivasa, S. (2006). "Hierarchical motion planning for self-reconfigurable modular robots," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Beijing: IEEE), 886–891.
- Brunete, A., Ranganath, A., Segovia, S., de Frutos, J. P., Hernando, M., and Gambao, E. (2017). Current trends in reconfigurable modular robots design. *Int. J. Adv. Robot. Syst.* 14:1729881417710457. doi: 10.1177/1729881417710457
- Butler, Z., Kotay, K., Rus, D., and Tomita, K. (2004). Generic decentralized control for lattice-based self-reconfigurable robots. *Int. J. Robot. Res.* 23, 919–937. doi: 10.1177/0278364904044409
- Campbell, J., Pillai, P., and Goldstein, S. C. (2005). "The robot is the tether: active, adaptive power routing modular robots with unary inter-robot connectors," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Edmonton, AB), 4108–4115.
- Ceron, S., Horowitz, L., Wilson, Jenae, N., Chen, C., Kim, D., et al. (2019a). "DONUTs: a scalable, self-reconfigurable robot with compliant modules," in *International Symposium on Multi-Robot and Multi-Agent Systems* (New Brunswick, NJ).
- Ceron, S., Wilson, N., Horowitz, L., and Petersen, K. (2019b). "Comparative analysis of sensing modalities in rigid and deformable modular robots for shape estimation," in *International Symposium on Multi-Robot and Multi-Agent Systems* (New Brunswick, NJ).

- Colbrook, A., and Smythe, C. (1990). Efficient implementations of search trees on parallel distributed memory architectures. *IEE Proc. E* 137, 394–400. doi: 10.1049/ip-e.1990.0049
- Daudelin, J., Jing, G., Tosun, T., Yim, M., Kress-Gazit, H., and Campbell, M. (2018). An integrated system for perception-driven autonomy with modular robots. *Sci. Robot.* 3:eaat4983. doi: 10.1126/scirobotics.aat4983
- Gilpin, K., and Rus, D. (2010). Modular robot systems. *Robot. Autom. Mag. IEEE* 17, 38–55. doi: 10.1109/MRA.2010.937859
- Goldstein, S., Campbell, J., and Mowry, T. (2005). Programmable matter. *Computer* 38, 99–101. doi: 10.1109/MC.2005.198
- Ishiguro, A., Shimizu, M., and Kawakatsu, T. (2006). A modular robot that exhibits amoebic locomotion. *Robot. Auton. Syst.* 54, 641–650. doi: 10.1016/j.robot.2006.02.011
- Jorgensen, M. W., Ostergaard, E. H., and Lund, H. H. (2004). “Modular ATRON: modules for a self-reconfigurable robot,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566)*, Vol. 2 (Sendai), 2068–2073.
- Karagozler, M. E., Campbell, J. D., Fedder, G. K., Goldstein, S. C., Weller, M. P., and Yoon, B. W. (2007). “Electrostatic latching for inter-module adhesion, power transfer, and communication in modular robots,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (San Diego, CA)*, 2779–2786.
- Kawahara, Y., Hodges, S., Cook, B. S., Zhang, C., and Abowd, G. D. (2013). “Instant inkjet circuits: lab-based inkjet printing to support rapid prototyping of UbiComp devices,” in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (Zurich)*, 363–372.
- Kirby, B. T., Aksak, B., Campbell, J. D., Hoburg, J. F., Mowry, T. C., Pillai, P., et al. (2007). “A modular robotic system using magnetic force effectors,” in *IEEE International Conference on Intelligent Robots and Systems (San Diego, CA)*, 2787–2793.
- Kubica, J., Casal, A., and Hogg, T. (2001). “Complex behaviors from local rules in modular self-reconfigurable robots,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)* Vol. 1 (Seoul), 360–367.
- Lee, J. Y., Kim, W. B., Choi, W. Y., and Cho, K. J. (2016). Soft robotic blocks: introducing SoBL, a fast-build modularized design block. *IEEE Robot. Autom. Mag.* 23, 30–41. doi: 10.1109/MRA.2016.2580479
- Li, S., Batra, R., Brown, D., Chang, H.-D., Ranganathan, N., Hoberman, C., et al. (2019). Particle robotics based on statistical mechanics of loosely coupled components. *Nature* 567, 361–365. doi: 10.1038/s41586-019-1022-9
- Naz, A., Piranda, B., Tucci, T., Goldstein, S. C., and Bourgeois, J. (2018). “Network characterization of lattice-based modular robots with neighbor-to-neighbor communications,” in *Distributed Autonomous Robotic Systems*, eds R. Groß, A. Kolling, S. Berman, E. Frazzoli, A. Martinoli, F. Matsuno, and M. Gauci (Boulder, CO: Springer), 415–429.
- Neubert, J., Rost, A., and Lipson, H. (2014). Self-soldering connectors for modular robots. *IEEE Trans. Robot.* 30, 1344–1357. doi: 10.1109/TRO.2014.2344791
- Pamecha, A., Ebert-Uphoff, I., and Chirikjian, G. S. (1997). Useful metrics for modular robot motion planning. *IEEE Trans. Robot. Autom.* 13, 531–545. doi: 10.1109/70.611311
- Piranda, B., and Bourgeois, J. (2018). Designing a quasi-spherical module for a huge modular robot to create programmable matter. *Auton. Robots* 42, 1619–1633. doi: 10.1007/s10514-018-9710-0
- Rubenstein, M., Cornejo, A., and Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science* 345, 795–799. doi: 10.1126/science.1254295
- Rus, D., and Vona, M. (2001). Crystalline robots: self-reconfiguration with compressible unit modules. *Auton. Robots* 10, 107–124. doi: 10.1023/A:1026504804984
- Russell, S. J., and Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*. Malaysia: Pearson Education Limited.
- Stoy, K., and Nagpal, R. (2004). “Self-repair through scale independent self-reconfiguration,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566)*, Vol. 2 (Sendai), 2062–2067.
- Swissler, P., and Rubenstein, M. (2018). “FireAnt: a modular robot with full-body continuous docks,” in *ICRA 2018-IEEE International Conference on Robotics and Automation (Brisbane, QLD)*.
- Vergara, A., Lau, Y. S., Mendoza-Garcia, R. F., and Zagal, J. C. (2017). Soft modular robotic cubes: toward replicating morphogenetic movements of the embryo. *PLoS ONE* 12:e169179. doi: 10.1371/journal.pone.0169179
- Walter, J. E., Tsai, E. M., and Amato, N. M. (2005). Algorithms for fast concurrent reconfiguration of hexagonal metamorphic robots. *IEEE Trans. Robot.* 21, 621–631. doi: 10.1109/TRO.2004.842325
- Walter, J. E., Welch, J. L., and Amato, N. M. (2002). Concurrent metamorphosis of hexagonal robot chains into simple connected configurations. *IEEE Trans. Robot. Autom.* 18, 945–956. doi: 10.1109/TRA.2002.805648
- Yim, M., Shen, W.-M., Salemi, B., Rus, D., Moll, M., Lipson, H., et al. (2007). Modular self-reconfigurable robot systems [grand challenges of robotics]. *Robot. Autom. Mag. IEEE* 14, 43–52. doi: 10.1109/MRA.2007.339623
- Zhu, L., and El Baz, D. (2019). A programmable actuator for combined motion and connection and its application to modular robot. *Mechatronics* 58, 9–19. doi: 10.1016/j.mechatronics.2019.01.002

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Wilson, Ceron, Horowitz and Petersen. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Mutual Shaping in Swarm Robotics: User Studies in Fire and Rescue, Storage Organization, and Bridge Inspection

Daniel Carrillo-Zapata^{1,2,3†}, Emma Milner^{1,2,3†}, Julian Hird^{1,2,3†}, Georgios Tzoumas^{1,2}, Paul J. Vardanega², Mahesh Sooriyabandara⁴, Manuel Giuliani^{1,2,3}, Alan F. T. Winfield^{1,3} and Sabine Hauert^{1,2*}

¹ Bristol Robotics Laboratory, Bristol, United Kingdom, ² University of Bristol, Bristol, United Kingdom, ³ University of the West of England, Bristol, United Kingdom, ⁴ Toshiba Research Europe Limited, Bristol, United Kingdom

OPEN ACCESS

Edited by:

Vito Trianni,
Institute Italian National Research
Council, Italy

Reviewed by:

Yara Khaluf,
Ghent University, Belgium
Danesh Tarapore,
University of Southampton,
United Kingdom

*Correspondence:

Sabine Hauert
sabine.hauert@bristol.ac.uk

[†]These authors have contributed
equally to this work

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 20 November 2019

Accepted: 24 March 2020

Published: 21 April 2020

Citation:

Carrillo-Zapata D, Milner E, Hird J,
Tzoumas G, Vardanega PJ,
Sooriyabandara M, Giuliani M,
Winfield AFT and Hauert S (2020)
Mutual Shaping in Swarm Robotics:
User Studies in Fire and Rescue,
Storage Organization, and Bridge
Inspection. *Front. Robot. AI* 7:53.
doi: 10.3389/frobt.2020.00053

Many real-world applications have been suggested in the swarm robotics literature. However, there is a general lack of understanding of what needs to be done for robot swarms to be useful and trusted by users in reality. This paper aims to investigate user perception of robot swarms in the workplace, and inform design principles for the deployment of future swarms in real-world applications. Three qualitative studies with a total of 37 participants were done across three sectors: fire and rescue, storage organization, and bridge inspection. Each study examined the users' perceptions using focus groups and interviews. In this paper, we describe our findings regarding: the current processes and tools used in these professions and their main challenges; attitudes toward robot swarms assisting them; and the requirements that would encourage them to use robot swarms. We found that there was a generally positive reaction to robot swarms for information gathering and automation of simple processes. Furthermore, a human in the loop is preferred when it comes to decision making. Recommendations to increase trust and acceptance are related to transparency, accountability, safety, reliability, ease of maintenance, and ease of use. Finally, we found that mutual shaping, a methodology to create a bidirectional relationship between users and technology developers to incorporate societal choices in all stages of research and development, is a valid approach to increase knowledge and acceptance of swarm robotics. This paper contributes to the creation of such a culture of mutual shaping between researchers and users, toward increasing the chances of a successful deployment of robot swarms in the physical realm.

Keywords: users, mutual shaping, swarm robotics, firefighting, rescuing, storage organization, bridge inspection, responsible research and innovation

1. INTRODUCTION

Swarm robotics uses a large number of robots that follow simple rules and use only local interactions to achieve seemingly complex group behaviors (Şahin, 2004; Brambilla et al., 2013). It has been demonstrated as a useful technology under laboratory conditions (Bayindir, 2016). Swarms have a wide array of application areas such as search and rescue (Penders et al., 2011),

construction (Werfel et al., 2014), and space exploration (Vassev et al., 2012). Despite a lengthy list of real-world applications, there is a lack of research into the practicalities of swarm robot deployment (Bayindir, 2016). One important factor that has not yet been properly investigated is public perception and likelihood of acceptance of robotic swarm products by users. The media and entertainment industry have depicted swarm robotics as something to be feared, according to Hamann (2018). This is troubling for the field since 21% of the respondents to a report by Ipsos MORI and the Royal Society (2017) said that their perception of AI was heavily influenced by mainstream media and entertainment (including science fiction). Additionally, a survey by the European Commission in 2017 (Eurobarometer, 2017) found 37% of respondents felt uncomfortable with robots assisting them at work. In relation to swarm robotics, it is not known what workers expect from robot swarms, and whether they would be comfortable working alongside them.

This work aims to address this gap by engaging with potential users of future swarm robotics systems. We create a two-way relationship between researchers and users which will encourage and inform mutual shaping of the technology. In particular, users acquire knowledge about the technology from researchers, and researchers learn about potential exploitation of the technology from users, hence critically revising the technology. In this paper, we present qualitative results from user participatory design style discussions with a total of 37 participants across three different sectors: fire and rescue, storage organization, and bridge inspection. Our goal during the three studies was to identify the challenges users face in their profession, learn from their reactions to possible assistive swarm systems, and discover any barriers to the system's acceptance, as well as to introduce them to the field of swarm robotics. By incorporating users in the early stages of research and development of swarm robotics systems, we aim to increase their adoption of the technology. This is essential to successfully implement such systems in real-world applications that have economic and societal benefits (Winfield and Jirotko, 2018).

2. RELATED WORK

There has been an abundance of research in human-robot interaction research in industrial settings (Berg et al., 2019) and in search and rescue (Murphy, 2004). There has also been important work into understanding what users need from search and rescue technologies such as Adams (2005), Driewer et al. (2005), Yanco et al. (2006), or Harbers et al. (2017). User studies help shape the technology itself and inform the requirements that the design processes should follow to produce a successful robotic product for an application. Successful here would mean working well alongside the human workers. For this, roboticists should investigate the attitudes of these workers toward robotics. Authors of studies such as Katz and Halpern (2014) have conducted interviews with people (in this case, students) about their opinions on the suitability of robots for various occupations. For example, it was found that the appearance of the robot played a part in the human worker's attitudes toward it and their

perceptions of its likely performance. Similarly, investigations have been conducted into the perceptions of robot capability and how desirable they are to workers (Takayama et al., 2008).

There is a lack of similar research into swarm robotics. There has been some research into human-swarm interaction (Couture-Beil et al., 2010; Nagi et al., 2012, 2014; Pourmehr et al., 2013; Kolling et al., 2016; Nam et al., 2019; St-Onge et al., 2019). However, the attitudes, perceptions and desires of workers for swarms has not yet been researched (to our knowledge). Existing research into how humans feel about swarms has focused on the psychophysiological response rather than opinions or expectations. For example, Podevijn et al. (2016) studied the effect increasing the size of robot groups had on the stress and anxiety of participants and found that a higher number of robots provoked a heightened response.

While a wealth of literature exists mentioning the sectors in this project, few describe a swarm system that operates in reality. For example, a range of robots have been developed for fire and rescue (see Murphy, 2014; Delmerico et al., 2019). Of these, the most complete swarm system is the GUARDIANS project (Penders et al., 2011). The GUARDIANS project developed a swarm of autonomous robots to assist firefighters with navigational support in low vision scenarios. In the context of the second study, storage organization, robots have been used in warehouses successfully for a number of years (Bahrin et al., 2016). Swarm algorithms for typical tasks in a storage facilities have also been developed such as cooperation when lifting objects (Wilson et al., 2014). The final study, bridge inspection, robotic solutions have generally used single UAVs (Murphy et al., 2011; Khaloo et al., 2018) and computer vision to process captured images (Yeum and Dyke, 2015). Swarm based mapping algorithms such as Kegeleirs et al. (2019) have been proposed which could be used for this application.

This work extends the current state of the art by examining the attitudes of users to real-world deployments of robot swarms. Based on this, we propose design principles that can facilitate the development of swarms for real-world applications, by increasing user acceptance of swarm robotics technology.

3. METHODOLOGY

User studies were designed following the principles of mutual shaping, a framework which aims to create a bidirectional relationship between users and technology developers to incorporate societal choices in all stages of research and development. This approach facilitates the creation of "more socially robust, responsive, and responsible robots" (Šabanović, 2010). In particular, the mutual shaping structure successfully applied by Winkle et al. (2019) was used to structure our three studies. Winkle et al. propose to split up mutual shaping sessions in three main parts:

1. **Pre-demonstration Discussion** to understand participants' initial ideas on the topic before being given information,
2. **Project Presentation and Robot Demonstrations** to introduce participants to the topic of the session by giving an overview of the state of the art, aims of the

project, an explanation of the topic, and (perhaps) a robot demonstration, and

3. **Post-demonstration Discussion** for participants to give their opinions to researchers about the topic as well as their requirements to advance in the development of the particular technology in discussion.

We adapted this methodology to the topic of swarm robotics. A summary of the resulting common structure that we followed across the three studies is given below. For a complete description for each structure, please see the **Supplementary Materials**.

1. **Art of their profession:** Participants are asked about their area of work, typical tasks/procedures in their jobs, tools/equipment they normally use, challenges they face, and their attitudes toward robots in their workplace. This first part allowed researchers to understand the art of the participants' profession, as well as their initial attitudes toward the use of robots at work.
2. **Introduction to swarm robotics and possible scenarios of application:** An explanation of swarm robotics is given for participants to learn about this technology and the current state of the art of robot swarms in their fields. Then, a series of imaginary-but-possible scenarios related to their fields, where robot swarms could be applied, are described by the researchers.

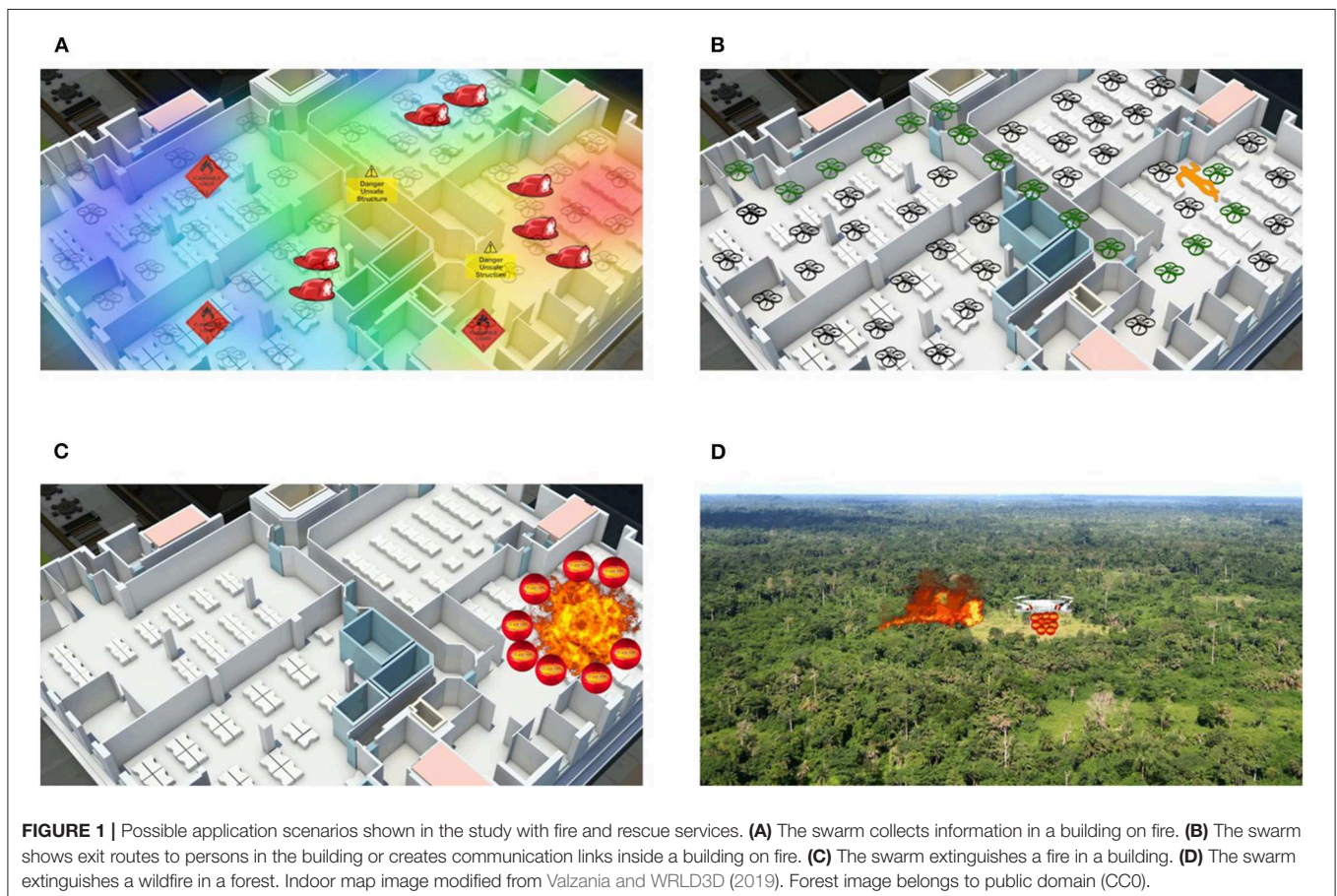
- **Fire and rescue study:** In this study, four different scenarios are presented to participants (see **Figure 1**). These scenarios show swarms collecting information in a building on fire, creating communication links or finding exit routes in the building, extinguishing the fire in the building, and extinguishing a wild fire in a forest.

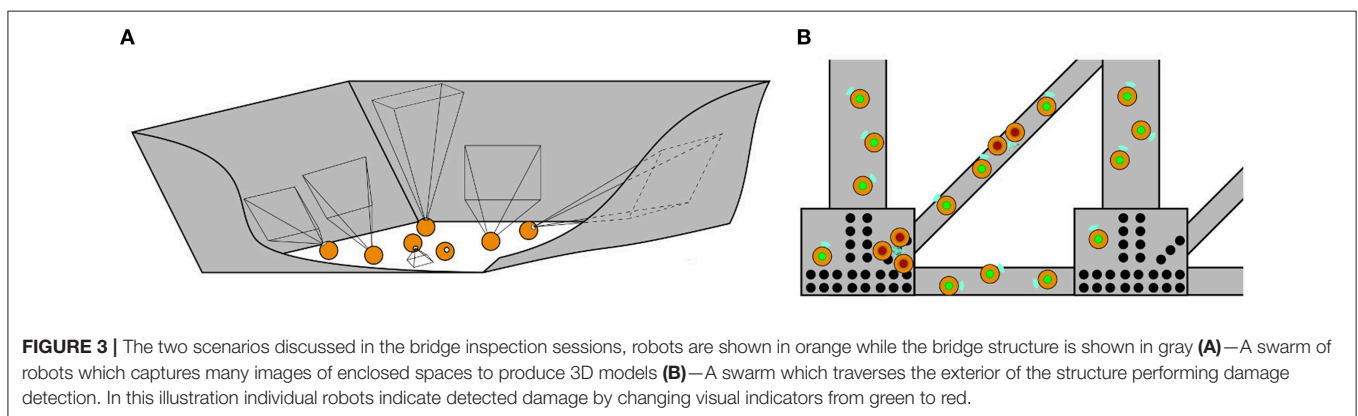
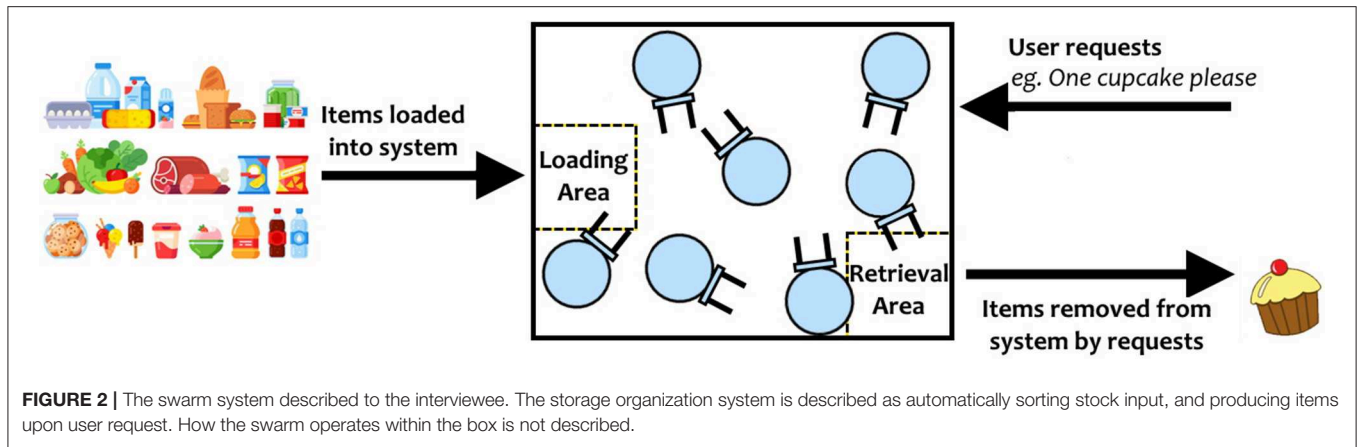
- **Storage organization study:** An out-of-the-box swarming system is described to participants (see **Figure 2**). This system can sort stock efficiently, provide information about the stock to the user, and retrieve items.

- **Bridge inspection study:** In this study, two scenarios are given to participants (see **Figure 3**). These scenarios describe swarms being released in a bridge and creating a 3D model of it by taking individual pictures, and swarms exploring the bridge to detect damage.

3. **Discussion of scenarios:** Finally, a group discussion of the previous scenarios, and others suggested by participants, is held between participants and researchers. In the discussions, the topics of acceptance, levels of autonomy, trust, swarm robotics vs. single robot approach, opportunities for swarm robotics in their fields and their concerns are brought up by the researchers. This part was used to identify the way forward to successfully apply robot swarms to their fields in the future.

For the study with fire and rescue services, focus-group-style sessions were chosen to have teams with different roles discussing





the topics and contrasting opinions during the same session, as opposed to interviewing firefighters individually. A total of 23 participants from three different fire and rescue services were recruited, with experience ranging from 1 to 20 years of service, as they verbally stated. Participant recruitment was done via email, word of mouth and on-site visits to fire and rescue services in the UK and Spain by the researcher in charge of this study. Participants were given an information sheet with a full description of the study and the focus group. They were also asked to sign a consent form to participate in the study and accept audio recording of the session, complying with university ethics regulations for experiments with human participants. Ethical approval was given by the University of the West of England. Three focus groups were held, one per service. The first focus group consisted of six participants from a UK fire and rescue service. There were participants working in the risk intelligence unit, IT, group management, media communication, operational effectiveness in instant ground, technology management, and drone piloting. This focus group was held at the Bristol Robotics Laboratory. In the second focus group, four firefighters from a fire station belonging to another UK fire and rescue service came to participate. This focus group was also held at the Bristol Robotics Laboratory. Finally, a third focus group was organized at a Spanish fire station with the participation of 13 firefighters. The diversity in participants allowed for the opinion of firefighters with real firefighting and rescuing experience

as well as people working in more technical fields related to development of processes. A pre-questionnaire and post-questionnaire was handed out at the beginning of the session (before any discussion could occur), and at the end of the session, respectively. Both questionnaires had the same questions, which are listed in **Supplementary Materials**. These questionnaires were used to measure the impact that the mutual shaping sessions had on participants.

In the storage organization study, an interview-style session was used rather than focus groups. This method was chosen because the interviews took place in the workplace to make arrangements easier for the subjects and to include an inspection of the storage space. The variation in locations and working hours meant that collecting participants together in a focus group was not possible. Interviewees were found mostly via email but also by word of mouth. A total of 25 introduction emails were sent out to 25 possible interviewees who fit the use case briefs. The following use case categories and sub-categories were contacted:

1. **Retail:** Charity shops; Shoe shops; Book shops; Jewellery shops.
2. **Food:** Supermarket; Food banks; Cafés.
3. **Industrial warehouses:** Supermarket depot; Retail e-businesses; Aerospace factories.
4. **Supplies in remote locations:** Space missions; Scientific (e.g., Antarctic) treks and laboratories.
5. **Other:** Museums; Independent shops; Stationary shops.

A total of eight interviews were conducted from six distinct use cases: a supermarket; a charity shop; a charitable food bank; a museum with a café and a gift shop; a large-scale industrial warehouse; and the space industry (specifically manned missions to other planets or space stations). Ethical approval was received from the University of Bristol on the condition of consent from interviewees, no audio recordings, and anonymous information gathering. For this reason, it was emphasized in the email that no recordings would be taken during the interviews, only handwritten notes, for which the interviewees gave permission for their answers to be used on consent forms. Attached to the introduction email was an information sheet that was written just for this purpose, explaining what swarm robotics is and what the benefits are of swarming systems. The interviews were performed on a semi-structured basis with a framework of key questions but the flexibility to move around topics that the interviewees wanted to discuss. All of the questions were asked without visual aids and spoken either in person or over the phone.

The bridge inspection study was conducted using focus groups because all participants were in the same industry, and it allowed the data to be collected more efficiently than with individual interviews. Ethical approval was given by University of Bristol. This required that only hand written notes were used to record participants' responses and that all participants remained anonymous. Four companies within the UK bridge inspection industry were contacted via email directly. Two different companies responded to the request leading to two sessions with six participants in total. All participants were engineers and inspectors involved in the management of bridge structures or the inspection process itself. The focus groups were executed in a semi-structured fashion. One researcher lead the discussion while another made handwritten notes. Once participants had read an information sheet, and were happy to participate, a consent form was signed and the session could begin.

This paper aims to be a first step toward understanding requirements of robot swarms through a mutual shaping methodology, built on in-depth, qualitative analysis of interviews with users to identify common themes across the three studies. It does not intend to be a quantitative analysis of user needs, which would require a different methodology based on broader sampling and recorded demographic data. In this sense, questionnaires were not used in the storage organization and bridge inspection studies because they had fewer numbers of participants, and were shorter in duration, due to the nature of the professions targeted.

4. RESULTS

4.1. Fire and Rescue Study

Below we combine the results from the three focus groups held with fire and rescue services to summarize their current processes, challenges, and attitudes toward using robots in fire and rescue.

4.1.1. The Art of the Profession

Nowadays, firefighters are in charge of many different tasks, not only firefighting. Apart from fires, they go to vehicle collisions, major transport incidents, and hazmat incidents. They also do urban search and rescue (when a building collapses), mine rescue, water rescue, animal rescue, and community-based roles to educate the public. When facing incidents, the first things they do are related to gathering as much information as possible for their risk assessment decision-making processes. Before handling the incident, firefighters perform quick checks to guarantee their safety first, e.g., they assess that the structure is safe to operate, or locate access/exit points. After enough information has been collected, firefighters start actions, i.e., firefighting or rescuing, until the incident is completely handled. Then, a fire investigation to discover the cause of the incident might take place. When participants were asked during the focus groups about the current tools they use for firefighting and rescuing, they stated that all tools they use are not automated, but require human operation. A summary of the tools that they currently use is given below:

- **Sensors and actuators fitted to buildings.** They said that smoke detectors, heat detectors, and water sprinklers assist them before/during firefighting.
- **Thermal imaging cameras.** They are used to create a map of temperatures to look for the source of the fire and casualties. They are particularly useful to predict what is behind areas with difficult access in buildings. Firefighters highlighted how this type of camera improves their performance:

“Thermal image cameras are one of the great tools we've got. So we can actually see in darkness and make our way around.”

- **Hydraulics.** They use hydraulic tools to cut through things.
- **Maps in the fire truck.** These maps are used to locate possible risks, water supplies, or weather conditions before arriving to the incident.
- **Radio-frequency identification (RFID).** Used for tracking of firefighters.
- **Colcut™ cobra.** A system that uses high-pressure water to pierce through walls and fog when they cannot access the room next door.
- **Teleoperated ground robots (QinetiQ™).** Sometimes they use them to gather information in hazmat incidents:

“It's got several cameras and a small water jet for testing temperatures rather than actually extinguishing anything. We used to use them with some level of success.”

- **Drones.** Pilots mostly use them to gather information about incidents to make an assessment of scenarios. They have also used them to track people who have gone missing.
- **Air fans.** Used for tactical ventilation, which means creating positive pressure in a building to push smoke out.

4.1.2. Their Challenges

Participants highlighted their main challenges are related to obtaining enough, accurate, and quick information about the

incident so that they can feed it into their decision-making processes. In fact, they mentioned they are quite quick in dealing with fires. The challenge for them is to find the location of those fires, and casualties to rescue. They said this is a challenge because many times the information they get is not accurate:

“In a lot of cases even the information you get [...] is not always 100% accurate. The address could be wrong or the actual type of fire could be wrong. It will come in as a hedge on fire and you get there to find a fire in a building. Your site has no persons trapped, there's no persons involved in anything at that point, and you get there and you find that there are. There's always a variable. You have minimal information.”

4.1.3. Opinions on Usefulness of Robots for Fire and Rescue

Participants could see value in using robots for fire and rescue, as shown in the results of question 1 (“In your opinion, how useful could robots as a firefighting/rescuing tool be in the future?”) in **Figure 4**. In fact, 20 out of 23 participants ticked *very useful* or *extremely useful* in the post-questionnaire. There was a slight shift from *very useful* to *extremely useful* from the pre-questionnaire to the post-questionnaire, meaning that participants' attitudes were already positive before the sessions. However, they did not think robots should be used for all tasks. Results from question 2 (“In which firefighting/rescuing tasks would robots be most useful?”) in **Figure 4** show that information-gathering tasks (locating victims, risk/incident

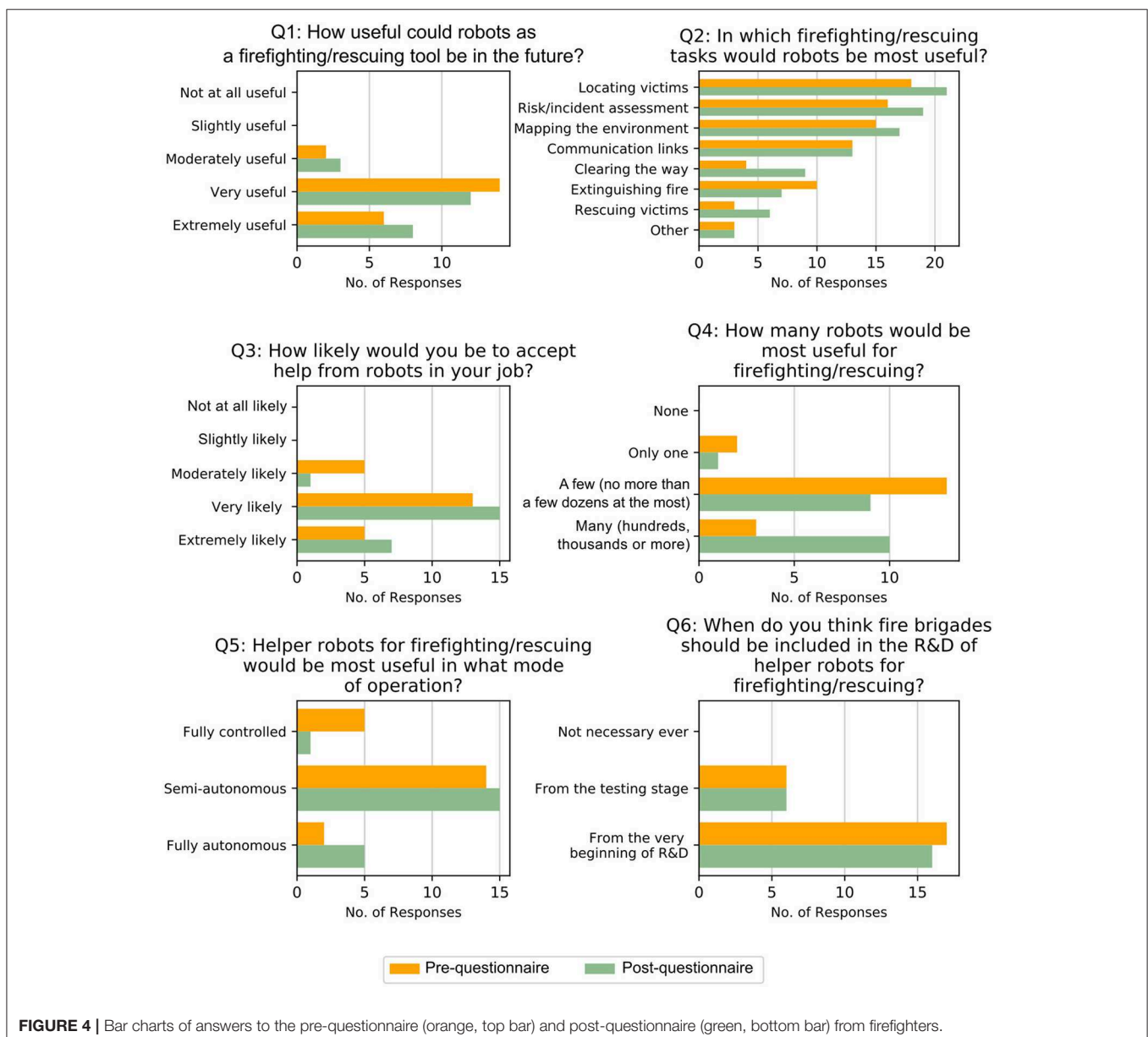


FIGURE 4 | Bar charts of answers to the pre-questionnaire (orange, top bar) and post-questionnaire (green, bottom bar) from firefighters.

assessment, mapping the environment, communication links) were the ones that participants preferred—they were ticked by over half of the participants. Action-based tasks (clearing the way, extinguishing fire, rescuing victims) were ticked less often, by much less than half of the participants. It is worth highlighting that all tasks but *extinguishing fire* were ticked by the same or more participants in the post-questionnaire, compared to the pre-questionnaire. Hence, participants could see more value in using robot swarms after the session, but thought that extinguishing a fire was too complex to be done by robots. Their preference for information-gathering tasks was also highlighted during the focus groups. Participants said that they would prefer robots doing simple tasks, such as going inside a house, mapping it and coming back to them with information; locating casualties; or sending them to gather information before they get to the incident or searching large areas (e.g., ships).

Participants also highlighted the benefit of using robots to create communication links among firefighters (to coordinate their operations) and between firefighters and casualties (to send them reassuring messages). Indeed, one participant mentioned that their research team was looking specifically at what technology they could deploy into a building to have communication across the whole building. Also, they said that there is poor radio communication in many areas where they go, and they would benefit from deploying relays to establish communications in those areas. Apart from the tasks listed in question 2 of the questionnaire, participants had the choice to specify other tasks that they thought robots could do. In the questionnaires, some participants wrote down the following tasks: Hazardous environment identification, post fire investigation (imagery), bring emergency kit (water, oxygen, food, etc.), protection of victims, rescuer, and habitable zones. During the discussion, even more examples of tasks were raised, such as:

- **Real-time information.** They said it would be useful to have a swarm of robots deployed across the area of the incident to send constant updates.
- **Dangerous or repetitive tasks.** They mentioned they would rather have robots where a human being would be in danger, e.g., hazmat environments. Also, some firefighters mentioned they would like to have robots for repetitive tasks, especially to prevent injuries in firefighters.
- **Finding exit routes.** Participants highlighted their difficulties when dealing with heat stress, because they sometimes get confused/lost inside fires. For that, they thought that having robots finding/lighting up the exit route for them would be particularly beneficial:

“A building could be like a maze that we’re not familiar with [...] You want something that could light up [...] the floor glow [...] something that could glow in the dark.”

- **Tactical ventilation.** Participants gave the example of a swarm of drones using their propellers to perform tactical ventilation to push smoke out of the building.

- **Accessing inaccessible places for firefighters.** Participants said that robots attacking fire in high buildings, where their ladders cannot reach, could be a positive application. They also pictured robots rescuing people from cliffs or water, which sometimes are inaccessible to them.

4.1.4. Opinions on Acceptance of Robots for Fire and Rescue

Participants answered positively to question 3 of the questionnaire (“How likely would you be to accept help from robots in your job?”). All participants but one ticked *very likely* or *extremely likely* in the post-questionnaire, and there was no answer below *moderately likely*, as can be seen in **Figure 4**. As in question 1, there was also a slight shift toward more positive answers with respect to the pre-questionnaire, but participants were very positive before the session. In fact, during the focus group, participants pointed out that they do not fear robots becoming a replacement for firefighters. Instead, they see them as a tool that could assist them and enhance/complement their operations:

“None of us are negative. We all would like it to happen. Yeah it’s just better to have an extra pair of eyes and another person. You just add it to what you’re doing visually anyway. Bring it all together, I can certainly see it being really useful for giving us more information.”

When thinking about acceptance from citizens being rescued by robots (or with the help of robots), participants felt that citizens should be educated. They should know what to expect if robots are used for firefighting and rescuing in the future.

4.1.5. Opinions on Robots Swarms for Fire and Rescue

After the session, participants could see how using a large swarm of robots may be the most advantageous option. In question 4 (“In your opinion, how many robots would be most useful for firefighting/rescuing?”), using *many* robots came out as the preferred choice by 10 participants, over *a few* (ticked by nine participants) and *only one* (ticked by only one participant) in the post-questionnaire. It is worth mentioning that two participants did not answer this question, and another one ticked both *a few* and *many*, which was not allowed. Thus, it was not included in the graph of **Figure 4**. Remarkably, using *many* robots was ticked by only three participants in the pre-questionnaire. Therefore, participants did see the advantages of using a swarm of robots after the sessions.

During the group discussions, participants understood the base principles of swarm robotics, and highlighted their benefits for fire and rescue. In particular, they said that redundancy is one of the key benefits. Most participants preferred to use a robot swarm even if robots could become obstacles (but left this as a requirement for the future). Also, most participants commented that having a large number of robots would be very useful to quickly search an area and gather as much information in the least amount of time as possible:

“The whole idea around swarm is you got some redundancy built in. [...] And some of the things we talked about is about location of casualties when it's dark. So deploying small agile devices that can search the rooms at the same time so that firefighters go in and then at least it's a beeping sound, 'yes, okay, let's prioritize that room.' [...] I think those sorts of things would be our friends.”

4.1.6. Opinions on Autonomy

Their preferred mode of operation for robots is semi-autonomy (15 responses in the post-questionnaire), as seen in results for question 5 (“Helper robots for firefighting/rescuing would be most useful in what mode of operation?”) in **Figure 4**. In fact, this was the participants preferred mode of operation before they participated in the session, as seen from the results of the pre-questionnaire. The session made them mostly abandon the idea of having *fully controlled* robots. It is worth mentioning that answers from two participants who ticked *fully autonomous* and *semi-autonomous*, and *semi-autonomous* and *fully controlled* were not taken into account. The directive stated multiple answers were not allowed, hence these answers were discarded.

From the group discussion, we understood that participants did not like the idea of robots taking autonomous decisions. They would trust robots carrying out information-gathering tasks or simple actions rather than stepping in the firefighters' decision-making process. Basically, participants feared that the robot system could cause more harm than benefits (e.g., knock-on effects) because there are many variables during fire and rescue, and lives at risk. They gave the example of robots opening up a window and changing the dynamics of the fire due to a change in air flow and the addition of oxygen to it.

In their opinion, robots could support their decision-making processes, but should not be in charge of them. From their comments, they would rather have a human in the loop being responsible for the actions taken when handling the incident:

“If it is autonomous just for firefighting, then I don't think that this is a corporate risk we would accept in this site. You can just imagine the headlines, it can help you and save you a thousand times. But one time it doesn't work properly and we lost a building through fire. Or loss of life even worse. Imagine the headlines: 'Firefighters sit outside and do nothing while robots sacrifice and get it wrong'. That's a risk that, until the idea is developed and understood more widely, probably we would not accept.”

4.1.7. Opinions on Involvement in the Research and Development Process

The final question was related to when fire and rescue services should be included in the research and development process (“When do you think fire brigades should be included in the research and development process of helper robots for firefighting/rescuing?”). A total of 16 participants answered *from the very beginning*, whereas only six participants ticked *from the testing stage* in the post-questionnaire. One participant did not answer this question, so it does not appear in question 6 in **Figure 4**. This aspect was not discussed during the focus groups. As seen in the answers to this question in the pre-questionnaire, mostly the same number of participants already

thought that fire brigades should be included from the beginning. Their participation in the sessions did not change this opinion.

4.1.8. Requirements for Trust in Robots That Assist in Fire and Rescue

This final section summarizes all the key requirements that participants felt robots used in firefighting and rescuing should have for them to trust these systems:

- **Robots should be easy and quick to learn, deploy, and maintain.** Participants said that setup time should be kept to a minimum to proceed as soon as possible, as well as the number of checks needed to maintain the robots because they do not have enough time. Cost of training should also be low, according to them.
- **Swarms should not become a physical obstacle for firefighters/casualties.** Firefighters described that fires are usually chaotic, with unpredictable conditions.
- **Info given from robots should be relevant and not complex.** Due to the amount of information they manage when dealing with an incident, firefighters said that robots should not give all, raw information, but instead should provide information that is as clear, relevant, and digested as possible.
- **Robot swarms should be reliable.** They stressed the importance of guaranteeing that robots work when deployed, and that the information they provide is accurate. This is because they would make decisions based on what robots tell them:

“The reliability needs to be on there because again, the first time it fails that's it, you've lost the cause in there. [...] Get through those cultural barriers and then you'll find that the actual application implementation of that would be a lot easier.”

- **Robot swarms should be accountable.** Participants said that the data gathered from robots must be stored and timestamped. This is important for their internal investigations.
- **Robot swarms should be safe.** Finally, participants said that robots should guarantee firefighters' safety.

4.2. Storage Organization Study

Results for storage organization study were also gathered, using similar quasi-structured questions. One-to-one interviews were used here, rather than focus groups. The answers to the questions and discussions in interviews are given in this section:

4.2.1. Summary of Use Cases

The following are descriptions of the use case stock rooms, based on the answers given by the interviewees when asked how they characterize their day-to-day work:

- **Supermarket** Stock is transported from the depot to the shop where it is moved from the van to the stockroom by employees. The stock is transferred within a large cage on wheels and is kept in this container while in the shop stockroom or stacked on the shelves by employees. There is no stockroom organization system.

- **Charity shop** Donations come in at sporadic, unpredictable times and range across a wide variety of items and quality. Staff initially sort items into two categories: bric-a-brac and clothing and then into further subcategories. The donations are stored in large piles in one corner of the sorting room before they are sorted. The sorting process includes putting clothing on hangers and individually pricing each piece.
- **Food bank** Food donations come in a random order and amount. Volunteers sort the items into categories and write the best-before date on the packaging so it is easily visible. The food is sorted and stored into crates of the same food type and best-before date. When people come in for food donations they make a request for a list of products and the volunteers create a bespoke food package for each customer from the stock in the warehouse.
- **Industrial warehouse** When new stock arrives at the warehouse, before it enters the automated part of the system (which uses a combination of humans and robots to pick and pack items) it is moved using automated guided vehicles (AGVs) into the warehouse storage.
- **Museum** There are multiple parts of the museum building: café; gift shop; event locations; main collection; and archives. All of these areas have their own stock to organize and store. They coordinate their timetables for stock incoming and outgoing using a shared calendar system but do not discuss details beyond this. Extra storage (such as fridges) is hired in for large café events. Volunteers sort through the collection archives, which have over 1 million pieces, and record details about each piece on pen and paper to categorize what is in each storage box.
- **Space missions** On the International Space Station (ISS), astronauts keep track of the food and supplies stock and search the store when they want something. It is predicted that when humans go for extended, residential trips to the Moon or Mars they will need to do this as well. Orders for new supplies need to be made many months in advance because it is difficult to send to them. This means that keeping a careful log of what stock they use and when is important to avoid running out of supplies too early.
- **Large-scale Retailer** The stock is stored in large warehouses from which online orders are packed and shipped. A centralized, robotic system is used to move stock to a conveyor belt where it is transported to human pickers who pack the products.

4.2.2. Current Processes

The following are descriptions of the current, storage organization tasks used by the different use cases, as discussed in interviews. Common processes are grouped together:

4.2.2.1. Inventory

The robots in the system used by the **Large-scale Retailer** automatically scan all stock items and all items are kept in cardboard boxes. This means that there is a constantly updated inventory and corresponding location list. In the **Supermarket**, when a delivery comes in from the depot, a list of what is included in the stock is added to a central database on a computer. The

individual items are not checked by the shop employees against the list for errors. This can lead to “negative stock” which is stock that is counted as being in the inventory but never actually arrived. Any items that were on the list of items that arrived but have not been sold or wasted are assumed to have been stolen. In the **Museum**, technology is not used (i.e., no digital record) because there is no network infrastructure in the archives and the volunteers tend to not want to work or train to work with computers. Additionally, it was noted that management was afraid of a risk of losing data due to a computer problem and stated that pen and paper were therefore more reliable. Supplies on **Space** missions such as the International Space Station (ISS) are counted and recorded by crew members. The **Food Bank** and the **Charity Shop** do not keep any inventory or map of their stock and instead they both do stock rotation by eye.

4.2.2.2. Sorting

When donations come into the **Food Bank** or the **Charity Shop**, items are sorted into different categories and stored with other like items. In the case of the charity shop, prices are decided based upon the sorter’s personal opinion but reasoning is according to: current trends; brands; quality; judgement. It was stated by one staff member that the reasoning for a price is often just a feeling about how much it’s worth.

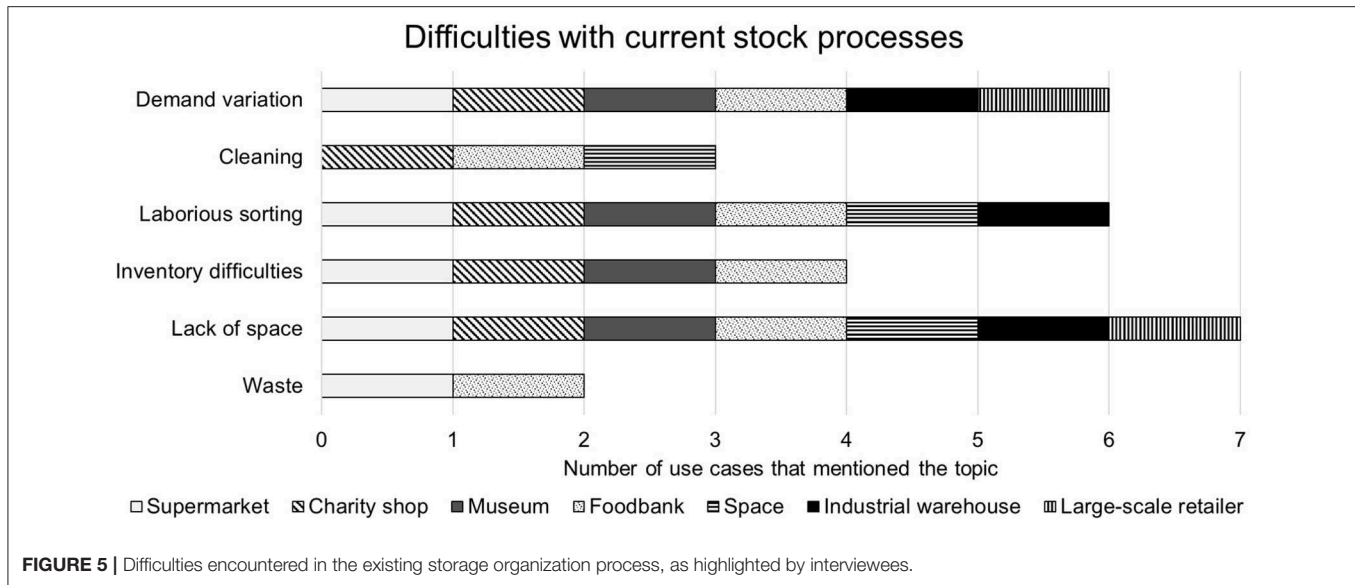
In the **Supermarket**, similar items arrive together and no re-sorting is done, they are just stored in the stockroom as they are when they arrive. Similarly, in the **Museum**, items are kept in the archive in boxes of mixed types, but no sorting is done. The **Industrial Warehouse** has workers who drive the pallets of new stock from the deliveries into the storage unit of the warehouse where it is collected and sorted by robots into the high-level system. Items in the **Large-scale Retailer** warehouse are sorted into locations depending on speed of movement. Here, fast movement means items are likely to be needed soon such as returned items. Beyond this, items are sorted at random with stock being constantly rearranged by the robots, even overnight, to be more efficiently stored. This is because the items are stored three rows deep so constant rearranging makes it less likely that something will be blocked behind other items for too long to be inefficient.

4.2.3. Challenges With Current Processes

The parts of the current stock organization processes that were highlighted by the interviewees as being negative or difficult are summarized in **Figure 5** and given in the following:

4.2.3.1. Sorting issues

All interviewees said that they thought the sorting system that was currently used could be significantly improved and that they wanted to do less sorting themselves. In this way they were all enthusiastic about a technological solution that would mean that they had to do less sorting of stock and/or the process would be quicker and easier. For example, the **Space Industry** experts stated that an astronaut’s time was expensive and limited, meaning that sorting stock was considered a waste of resources that should be automated. Similarly, the **Industrial Warehouse** interviewee said that loading speed could be vastly increased to



save time and money. This opinion was shared by all of the use cases for similar reasons.

4.2.3.2. Limited space

All of the interviewees said that a disadvantage of their current processes for stock handling was limited space in which to do it. For example, the **Food Bank** storage space was limited, meaning that piles of crates were three rows deep in some places which made it difficult to reach items at the middle or back of the pile. Similarly, in the **Charity Shop**, it was noted by the interviewee that this makes it especially difficult to search the donations for specific items to replenish supplies that are out of stock. The **Space Industry** representatives said that the storage space available is limited because it has to be habitable for humans to manipulate stock. The alternative, which would save money and therefore allow more available storage space, would be to not pressurize or supply oxygen to it, meaning the astronauts would walk around it in their spacesuits. The disadvantage of this is that it takes a long time to get spacesuits on which would also be a waste of time, especially as going and retrieving stored food is likely to be necessary multiple times a day.

The **Industrial Warehouse** employee also stated that there were economic reasons (i.e., cost of land) for keeping the space used for sorting goods to a minimum. The resulting problem is that it is difficult for the AGVs to move around and to prevent traffic jams as goods are being transported from delivery to storage. The **Large-scale Retailer** said that they wanted their system to be more dynamic. This is because the limited space for the robots to move means that when a robot breaks down it can block the way and make some stock areas inaccessible.

4.2.3.3. Demand variation

All interviews except the representatives from the space industry said that demand variation and unpredictable incoming and outgoing stock made it more difficult to do their stock-handling jobs. For example, in the **Supermarket** the stock is more

predictable but orders often vary, which can cause the stockroom to become busy and therefore difficult to keep organized. Similarly, in the **Industrial Warehouse** demand can go up and down in the same day, which puts a strain on the current processes due to the need for quickly adapting behaviors.

4.2.3.4. Inventory

The **Museum** stated that mistakes are often made by their volunteers when recording archived items. Similarly, the **Supermarket** said that they do not check stock against the stock list as it comes in so they are not aware of inventory errors but they do occur without their knowledge. No inventory is taken for the **Charity Shop** or the **Food Bank**, which can make the stockrooms hard to search for specific items when they are needed. This is a particular problem for the food bank when a customer requests a specific brand or has an allergy requirement because they do not keep any record of this information. The volunteers have to go to the area of the warehouse with the correct type of food and look at individual items for a matching one. This is laborious and slows down the whole process.

4.2.3.5. Cleaning

The **Food Bank** expressed that they spent a lot of their volunteered time cleaning the products. They resented having to do this and blamed the layout of the warehouse which was difficult to rearrange because of lack of space and heavy crates. The **Charity Shop** also said that cleaning incoming donations was part of their job but that they only did it when an item was likely to get a good enough price to be worth the cleaning time, otherwise they would put it in recycling or scrap materials. They consider cleaning an annoying part of their job, which is why they do not clean most items. The **Space Industry** representatives stated that general cleaning is a necessary part of an astronaut's duties but is considered to be a waste of their valuable time.

4.2.3.6. Waste

The **Supermarket** employee said that due to the way items are stacked together, the items at the bottom of the piles are often damaged. This is particularly common for products where the packaging is irregular in shape which also causes a waste of space due to inefficient packing. The **Food Bank** said that food can go out of date without the staff knowing because they do not have an inventory and cannot see the crates of food that are buried within the pile.

4.2.4. Attitudes Toward the Swarm System

At this point in the interview, the swarm system is described by the interviewer. It is described as a swarm system that automatically sorts stock that is input, and produces items upon user request. The following are the answers given to questions about this swarm system:

4.2.4.1. Features of the storage organization system

Many answers were common to more than one user and they are summarized in **Figure 6** about the desirable features to be included in a swarm system for storage organization. The most common desirable features were efficient storage (7/7 of use cases), automatic inventory check (5/7 of use cases), and automatic sorting abilities (5/7 of use cases). For example, the **Food Bank** said of the automatic inventory that this would allow them to cater to preferences and allergies more easily. They said that they would like a system that could allow them to do this and cater to other dietary requirements.

The next most useful features of the swarm system stated by 5/7 of use cases was automatically ordering items (e.g., the system would be able to recognize when there was favorable weather conditions or low stock of an item and make orders

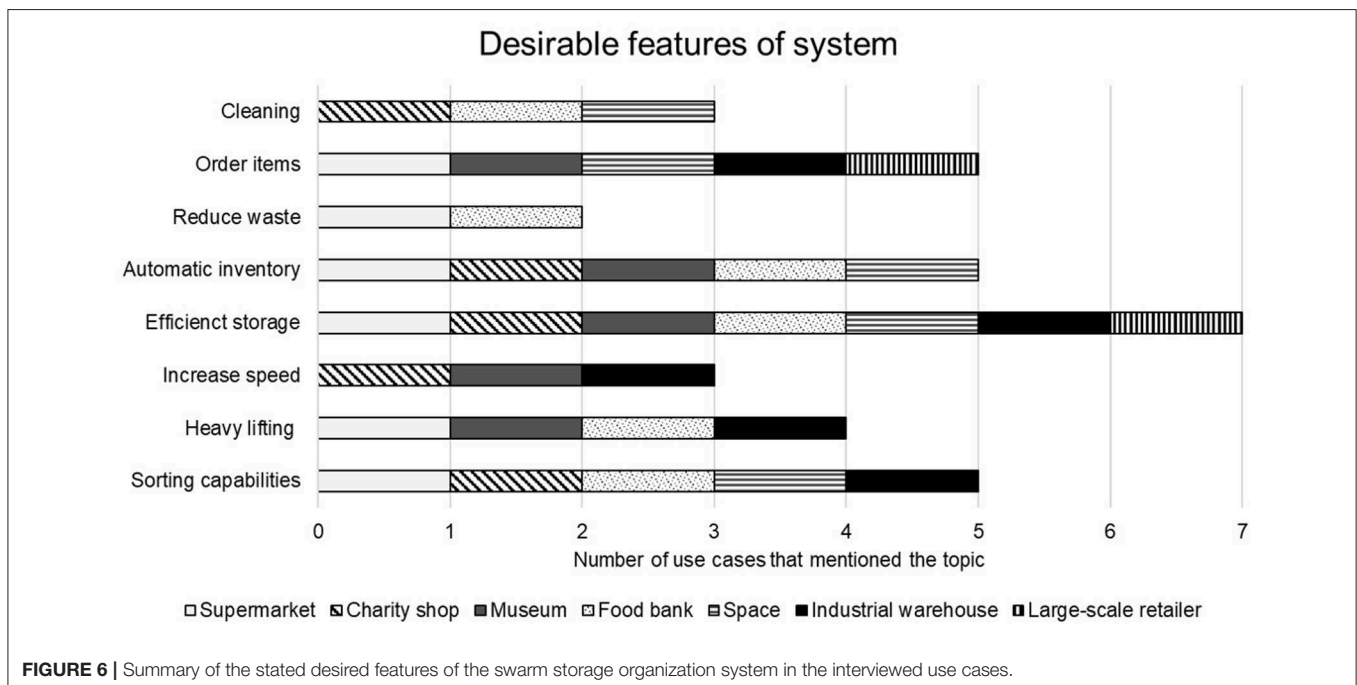
for new items as a result) and heavy lifting of stock. Finally, the other desirable features of the system stated were: cleaning abilities (3/7), increase loading speed or speed of processes such as inventory or transfer of goods (3/7), reduce wasted products (2/7).

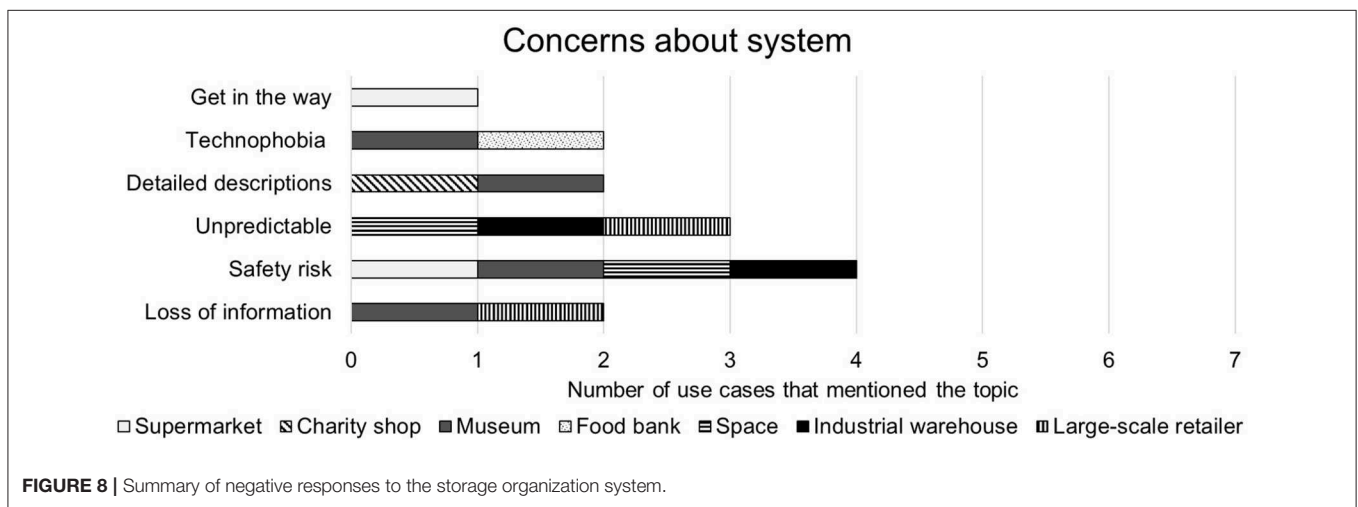
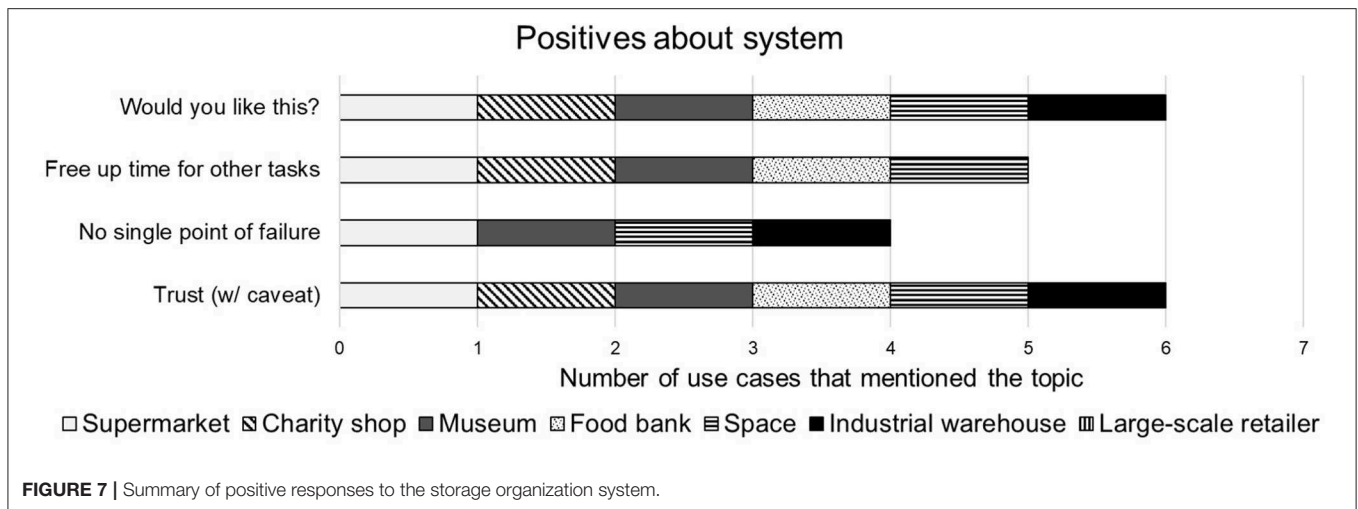
4.2.4.2. Positive comments

The interviewees were given the swarm system and asked for their thoughts about it. The main positive points are given in **Figure 7**. When specifically asked about how they felt about working alongside swarms of robots in general or compared to working alongside single robot systems the reactions were very positive with 6/7 stating that they would like to have this system in their place of work. Almost all (5/7) of the interviewees expressed positive opinions toward the suggested system for the given reason that it would free up time for some other task. 4/7 interviewees stated that they preferred the swarm system to a similar single robot system because there is no single point of failure in a swarm system.

4.2.4.3. Negative comments

Concerns expressed during interviews about the swarm system are given in **Figure 8**. The **Large-scale Retailer** was the only use case to state outright that it would not want this system. They said that their priority was stock control and they did not like that the individual agents would not be centrally controlled at all times. They also said that they thought that the swarm would require initial learning stages and they could not afford to have a system that was not good enough to work right away. This was not something that was given in the swarm system, but it is an opinion of swarm robotics that was felt before the interview. They also said that they did not like not knowing





where the information and behavior was coming from at all times within the swarm. They stated that they felt a swarm would risk losing information that could create a disastrous fault within the warehouse management.

The most common concerning topic was safety with 4/7 interviewees citing it as a risk factor when working with robots. For example, the **Museum** said that battery fires and trip hazards were both safety risks in the proposed system. The **Space Industry** and **Industrial Warehouse** representatives were both concerned about the unpredictability of a swarming system as opposed to a directly controlled system. The **Museum** and the **Charity Shop** both said that they did not think that a robotic system of any kind would be able to give rich enough descriptions of stock to improve upon human workers. The charity shop did not think that the system would be able to price items because of this gap, but they were happy with the idea of a technology that worked alongside humans, where swarms would sort items and humans would check and price them. There were also worries expressed for the risk of loss of information due to technology failure (expressed by the **Museum**) and that volunteers or staff

would not be able to work with the technology (expressed by the **Museum** and the **Food Bank**). The **Supermarket** employee said that they would like the system but were concerned that it would get in their way if it used drone technology. It should be noted that drone technology was in no way mentioned to the interviewee prior to this comment.

4.2.4.4. Trust

The **Large-scale Retailer** said that they would not trust a swarming system because they would not be able to know the information about where everything was in the warehouse and why it was there at any time. This is compared to their current, centralized system which is heavily controlled. 6/7 of the use cases said that they would trust the system but most 4/6 had a condition to add to this statement. The **Food Bank** had no caveats and the **Supermarket** said that damage was already caused to their products so they thought that the system would only improve this rate of damage even if it made some mistakes and therefore they would trust it. The **Charity Shop** said that they would trust the system with sorting and handling items but they did not think it

would be good enough to trust with pricing items without human supervision. The **Museum** was concerned about practical safety risks including the possibility of a collection piece being damaged if a robot collided with it. They said that if the system was proven to be safe then they would trust it.

The **Space Industry** said that it would trust the system if risk could be eliminated but the interviewees were split on how possible this would be. One representative said they considered swarms too unpredictable to ever be accepted in space applications where any mistakes can be mission critical. However the other representative said that they thought that swarms could be trustworthy if they were sufficiently developed and tested.

The **Industrial Warehouse** expressed that they were very interested in the system and would like to make it work but they would find it difficult to trust until it passed sufficient safety regulations. They expressed concern that this would be difficult, as no regulations currently exist.

4.3. Bridge Inspection Study

The focus groups conducted for the bridge inspection use case produced distinct themes as shown in **Table 1**. The details of these results are presented below.

4.3.1. The Art of the Profession

The participants described the task of bridge inspection as finding and assessing defects in the structural components of a bridge. This assessment was said to be crucial to ensure the bridge was safe and could be maintained properly. This task is not trivial with expertise required for identifying, quantifying and determining the consequences of any defect. One participant mentioned a difficulty in finding people with such skills. Both groups operated in fairly distinct sectors with one primarily inspecting railway bridges while the other inspected a variety of short to medium sized road bridges. Multiple levels of inspection were mentioned. The first level was a general inspection in which the bridge areas which were easy to access were surveyed (mostly visually) every 2 years. The second level was a detailed physical inspection, known as a principle inspection, that was carried out every 6 years. The principal inspection required all bridge elements to be inspected at close range. These procedures are inline with industry standards outlined in *Highways England* (2017). The first group were primarily concerned with these types of inspections as their expertise were in special access measures. The second group administered both types of inspections on behalf of a local authority.

4.3.2. Their Challenges

The challenge mentioned most by inspectors was accessing structural components of the bridge in difficult to reach areas. Participants described current measures such as rope-access and scaffolding as costly in terms of money and time. Participants of both groups also highlighted the diversity of the structures they have to inspect as another challenge. Each group described having to deal with bridges made from different materials and with different designs, many of which were built without any consideration of how they would be inspected.

TABLE 1 | Distinct themes were identified in notes taken in the bridge monitoring focus groups.

Task mentioned for robots	Positives	Concerns
Collecting data to help plan human inspections	More data collected	Speed
Constructing 3D models of bridge	Possible time savings	Cost
Providing information on hard to reach areas	Cost of individual units	Safety of inspection units Value of collected data Locomotion ability Retrieval of units

These were tasks participants thought robots could help with alongside positives and concerns participants had with the scenarios mentioned.

Another challenge frequently mentioned by both groups was that inspections had to be carried out in a way which minimized the disruption to the traffic on the bridge. For the first group this was a consequence of the dangers involved in inspecting railway bridges such as passing trains and high voltage cables. This led to small timeframes where inspections could take place. The second group highlighted that many of their bridges are essential links for rural communities and so closing the bridge would adversely effect these people.

4.3.3. Positives About Bridge Swarm Systems

In the discussion of the scenarios, most participants were receptive to working with robots and using data gathered from robots to help inform inspections. One group in particular saw the value in using a swarm, similar to that in scenario 2, to do a thorough sweep across large structures that could help target the deployment of roped access teams by logging the positions of detected defects on an existing 3D model. Participants also viewed enclosed small spaces such as culverts as useful environments to deploy a swarm in. Participants mentioned how current robot inspection of these areas uses a CCTV camera attached to a caterpillar track chassis, but these are very expensive. Hence they liked the idea that a swarm system was more modular and so losing single robots would represent a small financial risk. However, they did not imagine the swarm would be able to inspect the culvert itself but could provide useful information before human teams enter. For example the swarm could provide a rough dimensional survey to detect collapsed sections, or sense if hazardous gases had accumulated.

4.3.4. Concerns About Bridge Swarm Systems

The following are concerns expressed by the bridge inspection participants following a description of the scenarios. In this study, requirements for trust were not explicitly asked to participants, but the concerns expressed by participants indicate a lack of trust in some elements of the scenarios presented.

4.3.4.1. Data value

The type of data gathered by any system was one concern raised frequently. Both groups viewed touching as an essential part of an inspection but not something they thought a robot would

be able to emulate. They stated the importance of being able to sound parts of the structure with a hammer and examining the depth of any defect. Examples given included listening for hollow sounding areas which can indicate delamination in concrete or establishing the extent of paint flaking on steel elements. One participant also made the point that being able to identify issues in images of the structure came from doing this hands on work and this skill could be lost or diminished if the entire bridge inspection process was automated. While 3D models were viewed as a useful deliverable of a robotic system by participants, they also thought there were limitations with using them. They said that while potential defects could be identified using them, establishing their severity would often require visiting the defect in person as the model could not provide the same interactions as being on the structure in person. The groups also mentioned that changes in the condition of the structure were more important rather than one-off detections, such as in the second scenario. They thought robots would not be able to evaluate the severity of defects without knowing the previous state of the defect they had detected.

4.3.4.2. Time and cost constraints

Both groups referred to time and cost constraints as a major factor in determining if a particular technology was valuable and whether they would use it. For example, a textured 3D model was viewed as useful, and if there was no cost it probably would be widely used. However, participants in the second group viewed the time and cost in obtaining such a model as too high for the number and size of the bridges they dealt with. In their opinion many areas on these types of structures can be documented in sufficient detail from the ground with a few photos, so deploying a robotic system to get a very detailed model is overkill. Hence technology was only viewed as valuable when the environment was more constricted or complex, since this cannot be obtained easily with current practices.

4.3.4.3. Data processing

The amount of data that was collected and how it was processed was also highlighted. Both groups mentioned that a large amount of unprocessed data would not be helpful. For example, participants said that trawling through footage captured from robotic systems, or large collections of images had been tedious in the past. They also mentioned that structural health monitoring systems have this problem if not used precisely. This issue came up a lot in response to the second scenario, in which many robots covered the sides of the bridge detecting damage. Many participants stated that they imagined a system that captured data indiscriminately would flag up a large number of possible defects. Participants were then worried they would not have the resources to check each one during the inspection period. Some participants suggested the second scenario would be more useful if the system's output could be tied to a 3D model, as this would mean the data would not have to be checked in real time. In this case the system would simply collect more data about the structure than they do now. Both groups also highlighted that the top priority was identifying safety critical issues on the structure. Some participants felt a system that gathered more

data, if processed properly, would help in this task. However, others felt that the robots performing damage detection would be challenging, a view supported by the general observations on structural health monitoring by Webb et al. (2015).

4.3.4.4. Robot capabilities

Participants also had concerns about the abilities of the robots themselves, asking how they would move in such difficult environments. Many assumed the robots would need to fly and mentioned issues with using current inspection drones such as risks of collisions and flight restrictions. Another issue brought up was the retrieval of a large number of robotic units. Participants stressed that everything would need to be retrieved so that it would not contaminate natural habitats.

5. DISCUSSION

Although the three studies featured different fields of application (fire and rescue, storage organization, bridge inspection), there were similar results and opinions across the participants. In this section, we highlight those similarities to help shape future responsible and successful deployments of robot swarms in the physical realm.

5.1. There Is Opportunity for Swarm Robotics in the Workplace

Participants across the three studies welcomed robots for certain tasks, especially robot swarms. For them, the main advantages are the ones related to robustness via redundancy (no single point of failure) and high performance due to the use of a large number of robots. In the case of the fire and rescue focus group, these properties would be helpful in scenarios that participants felt were most useful, as identified in the focus groups (real-time information gathering, dangerous tasks, communication channels, finding exit routes, testing for hazards/traps, victim location/tracking, tactical ventilation), and from the questionnaire (locating victims, risk/incident assessment, mapping the environment, communication links), as Driewer et al. (2005) also found in their study. In these applications, high speed and large area coverage are common aspects, hence benefiting from a robot swarm collectively performing them in parallel.

Similarly in the storage organization study, almost all of those interviewed said that their current sorting systems would benefit from additional autonomy and they welcomed robot swarms (with caveats and assurances). Many of the use cases said that having an automated sorting system using a swarm of robots would be desirable because it would allow them to perform other less tedious and more useful jobs at the shop front. Tasks that they projected the robot swarm could do, that were not part of their current capabilities, included taking automatic inventory which many interviewees stated would improve the efficiency of their warehouses. This extended in almost all cases (automated warehouse, food bank, supermarket, charity shop, museum, and space) to predictive ordering based on projected demand changes informed by customer patterns, weather forecasts etc.

The bridge inspection participants were receptive to any technology which could gain valuable information about the structure. However, the value of the information was crucial to the participants' views of a swarm system. This value depended on the measurements being made, pictures could be used to identify defects but were less useful for characterizing their extent. The value also increased with the difficulty in accessing a given area by humans, such as confined spaces, or with increasing size of the structure, at which point human inspection becomes very slow and costly. The aspects the participants value also fit well with the swarm's abilities. For large structures, the area needing to be covered would be sizeable and suitable for a swarm's parallel operation. Enclosed spaces represent an unknown environment that would require the swarm's robust operation.

5.2. Identifying the “Art of Their Profession” Will Inform Tasks to Be Automated

There is a common theme across most participants in the three studies. They welcome technology that can assist them with certain tasks, but not all of them. In the fire and rescue study, participants' priority would be on robots that could assist them with information-gathering (e.g., locating casualties, mapping, communication links) or simple actions (clearing the way, lifting heavy things, tactical ventilation) with no autonomous decision-making process in place. This preference can be explained from two different points of view. On the one hand, participants pointed out that finding the fire/casualties and gathering information for their decision-making processes are the main challenges they face. On the other hand, they highlighted their fear that robots making autonomous decisions could cause more harm than good because of unforeseen consequences (many factors are in place during firefighting and rescuing), or lack of understanding of such decisions. Particularly interesting is their preference for not having fire extinguishing robots. Participants felt that there were many aspects involved in firefighting, and that only them, humans, would be capable enough to extinguish fires. This suggests there are certain aspects of their profession that they would not like automated, but done by humans—the art of their profession.

In the storage organization use case interviews, a concern from the workers was that swarm robots were not capable enough to sort the warehouse with full autonomy. For example, the Charity Shop workers said that they did not think that a robotic swarm system would be able to price the items correctly. They said that this is because when the human workers price items, it is a judgement that can be based on current trends, how the item feels, brands etc. In the same way, the Museum workers doubted that a robotic swarming system could replace human workers in being able to provide a detailed enough description of collection items to sort them. In this way, the art of the profession (i.e., the charity shop worker knowing from experience and instinct how much an item is worth) is something that workers consider to be an important part of the sorting process and not something they consider robots capable of doing without a human.

The bridge inspection study found that participants also doubted the robots' abilities to evaluate things, in this case

the condition of structural members. They stated that touching and sounding the structure are essential for finding the extent of any defects identified. Additionally many defects needed to be evaluated over time to determine their severity, hence a robot which is only measuring some quantity at one time point would not be able to quantify its seriousness. Participants agreed they would rather have robots supporting their decision-making processes as much as possible, but not acting autonomously when it comes to making decisions. Takayama et al. (2008) also found that robots were not preferred for occupations that require evaluation and judgement. Semi-autonomy, meaning that robots can perform some tasks by themselves but always subject to human input (human in the loop), is the preferred mode of operation. Semi-autonomy was also the preferred mode of operation in the study done by Driewer et al. (2005).

Robots are often negatively portrayed as machines taking over jobs. The fact that there are some aspects of their profession that participants would like to protect could seem to be related to this, although a direct question about fear of losing their jobs was not asked to participants. Participants broadly welcomed the use of robots in their jobs, and agreed they would be a tool to enhance/assist in their operations rather than a replacement. This is similar to the findings of the survey by Takayama et al. (2008) in which non-expert participants were more likely to prefer robots in a given occupation with people, rather than instead of people. Taking into account that there are barely any robot swarms currently in place in the professions explored in this paper, the fact that participants welcomed their use for certain tasks shows a high degree of preliminary acceptance. Therefore, when looking at how to best deploy robots in the physical realm, it is important to identify with end users which aspects are/are not desired to be automated to increase user acceptance.

5.3. Tackle Concerns to Increase Acceptance and Trust

Participants were mainly positive about robot swarms and the applications in their fields. However, there were caveats in each case, meaning that participants would trust swarm robotics systems under certain conditions. It is then crucial to address these concerns, if a successful implementation in society is sought. In fact, user acceptance and trust have been identified as the major bottleneck when taking robots to real-world applications (Kruijff et al., 2014).

5.3.1. Transparency and Accountability

In the study with fire and rescue services, participants pointed out that robot swarms should always store all the data they generate/process—timestamped. It is very important for them to understand *what* the swarm is doing, especially in case an investigation is required. In this sense, the swarm must be accountable, i.e. able to be queried and return a human-understandable answer. This is the concept of an ethical black box, described by Winfield and Jirotko (2017) as a mechanism to improve public trust by designing robots with accountability at the core.

Storage organization use cases were also concerned about the risk of loss of information and unpredictable behaviors. This

was particularly concerning for the Large-scale Retailer and the Museum who both store millions of products in one place and can therefore not afford to lose control of their stock. Other use cases with fewer active stock pieces were more willing to experiment with new technologies because the risk of loss of information if the system were to go down is not as great.

In the bridge inspection study, transparency and accountability were not mentioned directly by the participants when talking about the swarm scenarios. This may have been due to the described data collection task not involving the swarm taking substantial decisions that would need explaining to bridge inspectors. Participants also expressed doubts over the individual robots' ability to make substantial decisions such as evaluating detected defects. Whether participants maintained this view in the case of the system being made fully accountable was not investigated.

5.3.2. Reliability and Safety

For firefighters, another aspect to help build trust in robot swarms is reliability, i.e., the guarantee that if the robot swarm is deployed in a fire and rescue operation, it will work properly. In the scenarios they face, faults cost lives. Hence, all the information that the robot swarm might gather or the actions they perform must be completely accurate. This requires thorough verification and validation of the swarm robotics system before deployment. However, predicting the emergent collective behavior of robot swarms given the individual rules of each robot, and making sure that it is the only behavior that the swarm shows is a major challenge (Dixon et al., 2012). Further research on designing reliable swarms should be prioritized to increase trust, as well as reduce the number of risks arising from the use of swarms (Harbers et al., 2017).

Safety also came out as another requirement for acceptance and trust in the focus group with fire brigades. The robot swarm not becoming a physical obstacle (either for firefighters or casualties) was especially regarded as a crucial feature of the swarm robotics system. As argued above, this has to do with the requirement for robots not being detrimental to their operations. All in all, "technology is, in general, trusted if it brings benefits and is safe and well regulated" (Winfield and Jiroka, 2018).

The storage organization use case interviews found that safety was an important concern but of varying degrees. For example, the Museum cited worries about battery fires and trip hazards but it was not overly concerned about them since they are easily avoidable. On the other hand, the Space Industry representatives stated that missions are safety critical and therefore any technology that is included would have to have all risk removed before deployment could be achieved. They said that although they are interested in future developments of swarm robotics and its usefulness in space applications, they perceive its unreliability at this stage of development to be too high a safety risk to be viable for space missions. Both the Space Industry representatives and Industrial Warehouse stated they could not accept swarm technology until it passes safety regulations that are specific to swarm technologies.

Safety and reliability were also a primary concern of the bridge study's participants. For reliability, participants were concerned

about how the swarm individuals would move over the structure or inside an enclosed space without getting stuck. They were also concerned about how the individuals could be retrieved given the lack of a tether. There were other concerns related to robots falling or hitting things such as people, high voltage cables or traffic. Although, it should be mentioned other bridge inspection technologies such as drones, scaffolding, and roped access are not without their own risks (Dorafshan and Maguire, 2018). These safety concerns indicate that for swarm technologies to be accepted in the future, relevant safety standards will need to be developed (Winfield et al., 2004; Bjerknes and Winfield, 2013; Beltrame et al., 2018).

5.3.3. Ease of Training, Use, and Maintenance

Finally, most participants across the three studies agreed that they would trust the robot swarm assisting them at work as long as it was easy to learn about, use and maintain. In the study with firefighters, time is a crucial aspect for them. Hence, they require a system that can be deployed fairly quickly (ready by the time they arrive to the incident location), not too complex to use (their cognition abilities are harmed when firefighting, for example) and that does not require complex maintenance (always ready to be used). This places the focus on the scalability and adaptability of the robot swarm operations. Essentially, this means that if an action has to be done on the swarm, it should be independent of the number of robots in the swarm or the location of deployment.

Many of the storage organization workers interviewed said that their staff are volunteers and/or do not have a lot of spare time to train in how to use technologies. For this reason, out-of-the-box swarming systems would be needed to reduce set-up and maintenance during use. Any human-swarm interface would need to be very intuitive with little need for technology skills (for example, the museum said that their volunteers struggle with basic computer skills so they avoid technological solutions). In the bridge inspection study, participants were concerned with operating in tight cost and time constraints and so would also benefit from easy to use systems.

These results are in line with the findings from Yanco et al. (2006), where participants expressed their desire for the system to be easy to use—in fact, the system being difficult to use was the main cause for their test missions failing. Moreover, participants from the study led by Driewer et al. (2005) preferred an easy-to-use system. Authors then suggested having the ability to select different layers of information depending on what the specific user might require. This could indeed improve adaptability of the systems to users.

5.4. Mutual Shaping Can Facilitate the Deployment of Robot Swarms in the Physical Realm

The analysis of the responses to the pre-questionnaire and post-questionnaire in the study with fire brigades was used to understand the role of mutual shaping through focus group discussions in changing their opinions. The following changes in attitudes were noticed:

- **More tasks for robots:** In terms of tasks where robots could be useful, there was an overall increase in all tasks after the session (except extinguishing). This tells us that the session made them see how robots could be used for more tasks than they previously had thought.
- **Acceptance of robots increased:** There was over 20% increase to the *very likely* or *extremely likely* responses to the question related to acceptance of assistance from a robot. A total of 18 participants ticked either of these options in the pre-questionnaire, whereas 22 participants ticked participants ticked in the post-questionnaire.
- **A large swarm of robots is preferred:** When firefighters were asked about the number of robots they would rather have assisting them, *a few* was the most selected answer (13 participants), whereas *many* was chosen by only 3 participants. After the session, nine participants ticked *a few* and 10 participants ticked *many*.

Mutual shaping has been shown to be a successful way to engage in a two-way conversation with potential users and incorporate societal choices into the research and development process. If robot swarms are to be used in real-world applications, it is important to listen to all the parties who will be affected by it in the future. Almost three quarters of the firefighters said that they would like to be involved in the research and development process from the very beginning, in both questionnaires.

6. CONCLUSION

Robot swarms have been demonstrated performing a variety of tasks under laboratory conditions. However, potential users' exposure to the technology is limited. This has led to a number of unanswered questions around what people's perception of swarm robotics is, how comfortable people would be using the technology and what tasks they would like the technology to perform. In this work, three studies with a total of 37 potential swarm users were performed across three different sectors: fire and rescue, storage organization, and bridge inspection. Each study used participatory design style discussions that were structured to develop an understanding of each user's profession before introducing them to swarm robotics and discussing potential assistive swarm systems. It was found there was a generally positive reaction to robot swarms, but also some caveats. In both the fire and rescue and bridge inspection studies, participant's desired systems which would gather information to help inform human decisions. For the storage organization sector, a system which would sort stock and manage inventory in a space efficient manner was desired. Moreover, a common theme across the three studies was that there are some aspects of their jobs (especially when it comes to decision-making) that participants would not like to be done by autonomous robots. We call this *the art of the profession*. Therefore, it is important to identify with end users which aspects should be automated, and which should not, to increase users' acceptance. The caveats found were either due to doubts about the system's capabilities compared to a human or trust in its operation. To improve trust and acceptance in swarm systems in the future participants highlighted a number of areas

including: transparency, accountability, safety, reliability, ease of maintenance and ease of use. Finally, it was shown that designing the study with personnel from fire and rescue services following a mutual shaping approach positively changed their opinions about robot swarms assisting them.

Because swarm robotics technology is still being developed, now is the perfect time for swarm robotics researchers to create a link with users to identify what needs to be done to build trust and to ensure the technology is fulfilling a desired role. This will facilitate the deployment of robot swarms in the physical realm.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by Research and Enterprise Development (RED) at University of Bristol and Research Ethics at the University of West England. The participants provided their written informed consent to participate in this study. Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

AUTHOR CONTRIBUTIONS

DC-Z designed, recruited participants and facilitated study with personnel from fire and rescue services, analyzed transcripts and questionnaires, and substantially contributed to the writing of the manuscript. EM designed, recruited participants and facilitated study with personnel from storage organization, and substantially contributed to the writing of the manuscript. JH designed, recruited participants and facilitated study with personnel from the bridge inspection industry, and substantially contributed to the writing of the manuscript. GT assisted DC-Z during one focus group, and transcribed audio recordings. PV and SH made substantial contributions to the conception of the study with the bridge inspection industry, and assisted with participants recruitment. MG and SH made substantial contributions to the conception of the storage organization study, and assisted with participants recruitment. AW and SH made substantial contributions to the conception of the fire and rescue study, and assisted with participants recruitment. MS supported the storage organization study, including but not limited to the funding of the storage organization project via Toshiba Research Europe Limited. PV, MG, AW, and SH also critically revised the work.

FUNDING

DC-Z, EM, and JH were supported by the EPSRC Centre for Doctoral Training in Future Autonomous and Robotic Systems (FARSCOPE) at the Bristol Robotics Laboratory. The authors declare that a small part of this research received funding from

Toshiba Research Europe Limited. The funder had the following involvement with the study: co-funded and co-supervised Emma Milner's work related to the storage organization study.

ACKNOWLEDGMENTS

All authors would like to sincerely thank all the participants who took part in the three studies. JH would like to

thank Daniel Gosden for taking notes during the focus group sessions.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2020.00053/full#supplementary-material>

REFERENCES

- Adams, J. A. (2005). Human-robot interaction design: understanding user needs and requirements. *Proc. Hum. Fact. Ergon. Soc. Annu. Meet.* 49, 447–451. doi: 10.1177/154193120504900349
- Bahrin, M. A. K., Othman, M. F., Azli, N. N., and Talib, M. F. (2016). Industry 4.0: a review on industrial automation and robotic. *J. Teknol.* 78, 137–143. doi: 10.11113/jt.v78.9285
- Bayindir, L. (2016). A review of swarm robotics tasks. *Neurocomputing* 172, 292–321. doi: 10.1016/j.neucom.2015.05.116
- Beltrame, G., Merlo, E., Panerati, J., and Pincioli, C. (2018). "Engineering safety in swarm robotics," in *Proceedings of the 1st International Workshop on Robotics Software Engineering* (Gothenburg), 36–39. doi: 10.1145/3196558.3196565
- Berg, J., Lottermoser, A., Richter, C., and Reinhart, G. (2019). Human-robot-interaction for mobile industrial robot teams. *Proc. CIRP* 79, 614–619. doi: 10.1016/j.procir.2019.02.080
- Bjerknes, J. D., and Winfield, A. F. (2013). "On fault tolerance and scalability of swarm robotic systems," in *Distributed Autonomous Robotic Systems*, eds A. Martinoli, F. Mondada, N. Correll, G. Mermoud, and M. Egerstedt (Lausanne: Springer), 431–444. doi: 10.1007/978-3-642-32723-0_31
- Brambilla, M., Ferrante, E., Birattari, M., and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective. *Swarm Intell.* 7, 1–41. doi: 10.1007/s11721-012-0075-2
- Couture-Beil, A., Vaughan, R. T., and Mori, G. (2010). "Selecting and commanding individual robots in a multi-robot system," in *2010 Canadian Conference on Computer and Robot Vision* (Ottawa, ON), 159–166. doi: 10.1109/CRV.2010.28
- Delmerico, J., Mintchev, S., Giusti, A., Gromov, B., Melo, K., Horvat, T., et al. (2019). The current state and future outlook of rescue robotics. *J. Field Robot.* 36, 1171–1191. doi: 10.1002/rob.21887
- Dixon, C., Winfield, A. F., Fisher, M., and Zeng, C. (2012). Towards temporal verification of swarm robotic systems. *Robot. Auton. Syst.* 60, 1429–1441. doi: 10.1016/j.robot.2012.03.003
- Dorafshan, S., and Maguire, M. (2018). Bridge inspection: human performance, unmanned aerial systems and automation. *J. Civil Struct. Health Monit.* 8, 443–476. doi: 10.1007/s13349-018-0285-4
- Driewer, F., Baier, H., and Schilling, K. (2005). Robot-human rescue teams: a user requirements analysis. *Adv. Robot.* 19, 819–838. doi: 10.1163/1568553050011519
- Eurobarometer, S. (2017). 460. *Attitudes Towards the Impact of Digitisation and Automation on Daily Life*. European Commission.
- Hamann, H. (2018). *Swarm Robotics: A Formal Approach*. Cham: Springer International Publishing.
- Harbers, M., de Greeff, J., Kruijff-Korbayová, I., Neerincx, M. A., and Hindriks, K. V. (2017). "Exploring the ethical landscape of robot-assisted search and rescue," in *A World With Robots*, eds M. I. Aldinhas Ferreira, J. S. Sequeira, M. O. Tokhi, and E. E. Kadar (Lisbon: Springer), 93–107. doi: 10.1007/978-3-319-46667-5_7
- Highways England (2017). *BD 63/07 Inspection of Highway Structures, Vol. 3 Design Manual for Roads and Bridges*. London: HMSO.
- Ipsos MORI and the Royal Society (2017). *Public Views of Machine Learning*. Royal Society.
- Katz, J. E., and Halpern, D. (2014). Attitudes towards robots suitability for various jobs as affected robot appearance. *Behav. Inform. Technol.* 33, 941–953. doi: 10.1080/0144929X.2013.783115
- Kegeleirs, M., Ramos, D. G., and Birattari, M. (2019). "Random walk exploration for swarm mapping," in *Annual Conference Towards Autonomous Robotic Systems* (London: Springer), 211–222. doi: 10.1007/978-3-030-25332-5_19
- Khaloo, A., Lattanzi, D., Cunningham, K., Dell'Andrea, R., and Riley, M. (2018). Unmanned aerial vehicle inspection of the placer river trail bridge through image-based 3d modelling. *Struct. Infrastruct. Eng.* 14, 124–136. doi: 10.1080/15732479.2017.1330891
- Kolling, A., Walker, P., Chakraborty, N., Sycara, K., and Lewis, M. (2016). Human interaction with robot swarms: a survey. *IEEE Trans. Hum. Mach. Syst.* 46, 9–26. doi: 10.1109/THMS.2015.2480801
- Kruijff, G.-J. M., Janíček, M., Keshavdas, S., Larochelle, B., Zender, H., Smets, N. J., et al. (2014). "Experience in system design for human-robot teaming in urban search and rescue," in *Field and Service Robotics*, eds K. Yoshida and S. Tadokoro (Matsushima: Springer), 111–125. doi: 10.1007/978-3-642-40686-7_8
- Murphy, R. (2014). *Disaster Robotics*. Cambridge, MA: The MIT Press.
- Murphy, R. R. (2004). Human-robot interaction in rescue robotics. *IEEE Trans. Syst. Man Cybern. Part C* 34, 138–153. doi: 10.1109/TSMCC.2004.826267
- Murphy, R. R., Steimle, E., Hall, M., Lindemuth, M., Trejo, D., Hurlebaus, S., et al. (2011). Robot-assisted bridge inspection. *J. Intell. Robot. Syst.* 64, 77–95. doi: 10.1007/s10846-010-9514-8
- Nagi, J., Giusti, A., Gambardella, L. M., and Di Caro, G. A. (2014). "Human-swarm interaction using spatial gestures," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Chicago, IL), 3834–3841.
- Nagi, J., Ngo, H., Giusti, A., Gambardella, L. M., Schmidhuber, J., and Di Caro, G. A. (2012). "Incremental learning using partial feedback for gesture-based human-swarm interaction," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication* (Paris), 898–905.
- Nam, C., Walker, P., Li, H., Lewis, M., and Sycara, K. (2019). Models of trust in human control of swarms with varied levels of autonomy. *IEEE Trans. Hum. Mach. Syst.* 1–11. doi: 10.1109/THMS.2019.2896845
- Penders, J., Alboul, L., Witkowski, U., Naghsh, A., Saez-Pons, J., Herbrechtsmeier, S., et al. (2011). A robot swarm assisting a human fire-fighter. *Adv. Robot.* 25, 93–117. doi: 10.1163/016918610X538507
- Podevijn, G., O'grady, R., Mathews, N., Gilles, A., Fantini-Hauwel, C., and Dorigo, M. (2016). Investigating the effect of increasing robot group sizes on the human psychophysiological state in the context of human-swarm interaction. *Swarm Intell.* 10, 193–210. doi: 10.1007/s11721-016-0124-3
- Pourmehr, S., Monajjemi, V. M., Vaughan, R., and Mori, G. (2013). "You two! take off!": creating, modifying and commanding groups of robots using face engagement and indirect speech in voice commands," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Tokyo), 137–142.
- Šabanović, S. (2010). Robots in society, society in robots. *Int. J. Soc. Robot.* 2, 439–450. doi: 10.1007/s12369-010-0066-7
- Şahin, E. (2004). "Swarm robotics: from sources of inspiration to domains of application," in *International Workshop on Swarm Robotics* (Springer), 10–20.
- St-Onge, D., Levillain, F., Zibetti, E., and Beltrame, G. (2019). Collective expression: how robotic swarms convey information with group motion. *Paladyn J. Behav. Robot.* 10, 418–435. doi: 10.1515/pjbr-2019-0033
- Takayama, L., Ju, W., and Nass, C. (2008). "Beyond dirty, dangerous and dull: what everyday people think robots should do," in *2008 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (Amsterdam), 25–32.

- Valzania, G., and WRLD3D (2019). *Indoor Map*. Available online at: <https://www.wrl3d.com/blog/creating-indoor-maps-6-key-steps-for-beginners/> (accessed November 11, 2019).
- Vashev, E., Sterritt, R., Rouff, C., and Hinchey, M. (2012). Swarm technology at nasa: building resilient systems. *IT Profess.* 14, 36–42. doi: 10.1109/MITP.2012.18
- Webb, G. T., Vardanega, P. J., and Middleton, C. R. (2015). Categories of shm deployments: technologies and capabilities. *J. Bridge Eng.* 20:04014118. doi: 10.1061/(ASCE)BE.1943-5592.0000735
- Werfel, J., Petersen, K., and Nagpal, R. (2014). Designing collective behavior in a termite-inspired robot construction team. *Science* 343, 754–758. doi: 10.1126/science.1245842
- Wilson, S., Pavlic, T. P., Kumar, G. P., Buffin, A., Pratt, S. C., and Berman, S. (2014). Design of ant-inspired stochastic control policies for collective transport by robotic swarms. *Swarm Intell.* 8, 303–327. doi: 10.1007/s11721-014-0100-8
- Winfield, A. F., Harper, C. J., and Nembrini, J. (2004). “Towards dependable swarms and a new discipline of swarm engineering,” in *International Workshop on Swarm Robotics* (Santa Monica, CA: Springer), 126–142. doi: 10.1007/978-3-540-30552-1_11
- Winfield, A. F., and Jiotka, M. (2017). “The case for an ethical black box,” in *Annual Conference Towards Autonomous Robotic Systems* (Guildford: Springer), 262–273. doi: 10.1007/978-3-319-64107-2_21
- Winfield, A. F., and Jiotka, M. (2018). Ethical governance is essential to building trust in robotics and artificial intelligence systems. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 376:20180085. doi: 10.1098/rsta.2018.0085
- Winkle, K., Caleb-Solly, P., Turton, A., and Bremner, P. (2019). Mutual shaping in the design of socially assistive robots: a case study on social robots for therapy. *Int. J. Soc. Robot.* 1–20. doi: 10.1007/s12369-019-00536-9
- Yanco, H. A., Baker, M., Casey, R., Keyes, B., Thoren, P., Drury, J. L., et al. (2006). “Analysis of human-robot interaction for urban search and rescue,” in *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics* (Gaithersburg, MD: Citeseer), 22–24.
- Yeum, C. M., and Dyke, S. J. (2015). Vision-based automated crack detection for bridge inspection. *Comput. Aided Civil Infrastruct. Eng.* 30, 759–770. doi: 10.1111/mice.12141

Conflict of Interest: MS was employed by the company Toshiba Research Europe Limited.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Carrillo-Zapata, Milner, Hird, Tzoumas, Vardanega, Sooriyabandara, Giuliani, Winfield and Hauert. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Blockchain Technology Secures Robot Swarms: A Comparison of Consensus Protocols and Their Resilience to Byzantine Robots

Volker Strobel^{1*}, Eduardo Castelló Ferrer^{1,2} and Marco Dorigo¹

¹ IRIDIA, Université Libre de Bruxelles, Brussels, Belgium, ² MIT Media Lab, Cambridge, MA, United States

OPEN ACCESS

Edited by:

Eliseo Ferrante,
Vrije Universiteit
Amsterdam, Netherlands

Reviewed by:

Michael Crosscombe,
University of Bristol, United Kingdom
Heiko Hamann,
University of Lübeck, Germany
Julian Petzold,
University of Lübeck, in collaboration
with reviewer HH

*Correspondence:

Volker Strobel
vstrobel@ulb.ac.be

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 20 November 2019

Accepted: 26 March 2020

Published: 12 May 2020

Citation:

Strobel V, Castelló Ferrer E and
Dorigo M (2020) Blockchain
Technology Secures Robot Swarms: A
Comparison of Consensus Protocols
and Their Resilience to Byzantine
Robots. *Front. Robot. AI* 7:54.
doi: 10.3389/frobt.2020.00054

Consensus achievement is a crucial capability for robot swarms, for example, for path selection, spatial aggregation, or collective sensing. However, the presence of malfunctioning and malicious robots (Byzantine robots) can make it impossible to achieve consensus using classical consensus protocols. In this work, we show how a swarm of robots can achieve consensus even in the presence of Byzantine robots by exploiting blockchain technology. Bitcoin and later blockchain frameworks, such as Ethereum, have revolutionized financial transactions. These frameworks are based on decentralized databases (blockchains) that can achieve secure consensus in peer-to-peer networks. We illustrate our approach in a collective sensing scenario where robots in a swarm are controlled via blockchain-based smart contracts (decentralized protocols executed via blockchain technology) that serve as “meta-controllers” and we compare it to state-of-the-art consensus protocols using a robot swarm simulator. Additionally, we show that our blockchain-based approach can prevent attacks where robots forge a large number of identities (Sybil attacks). The developed robot-blockchain interface is released as open-source software in order to facilitate future research in blockchain-controlled robot swarms. Besides increasing security, we expect the presented approach to be important for data analysis, digital forensics, and robot-to-robot financial transactions in robot swarms.

Keywords: swarm robotics, blockchain technology, Byzantine fault-tolerance, resilient robotics, verifiable robotics

1. INTRODUCTION

Disasters, such as the collapse of a nuclear plant (e.g., Fukushima) or the release of petroleum into the environment (e.g., the Deepwater Horizon oil spill), present huge challenges and require quick and efficient responses. For example, it might be crucial to determine the average presence of radiation in a contaminated area (Brown et al., 2016). For security and efficiency reasons, on-site intervention might be better delegated to autonomous robots; and, to make the response more effective and mitigate potential adverse effects, the robots might have to perceive and act in different places at the same time. The coordination of such distributed activities by a central unit of control is not ideal as it makes the system less reliable (single point of failure) and possibly less efficient (communication overheads, delay in the collection of data, and in the release of control commands). Robot swarms, that communicate and collaborate in a peer-to-peer manner, are excellent candidates for these situations.

One important capability that robot swarms need to have to cooperate effectively is to be able to make collective decisions. Accordingly, collective decision-making is a well-studied subject in the field of swarm robotics (Schmickl et al., 2009; Montes de Oca et al., 2011; Reina et al., 2015; Valentini et al., 2016b, 2017). In general, to make a collective decision, robots in a swarm need to share their information and to aggregate this information using a distributed consensus protocol. The prevailing consensus protocol for averaging the values held by the individual entities in the swarm is the linear consensus protocol (LCP) (Olfati-Saber and Murray, 2004). However, this consensus protocol and most other protocols used in swarm robotics make the unrealistic assumption that all the robots in the swarm work as expected.

Unfortunately, real-world operation will almost certainly result in robots in the swarm that either fail (e.g., due to dust blocking their sensors) or that are malicious (e.g., due to a hacker who gains control). These failures can damage people, nature, animals, and other robots, making the reliable detection of failures a crucial task (Tarapore et al., 2019). We use the term Byzantine robot—based on Byzantine fault-tolerance and the *Byzantine Generals Problem* (Lamport et al., 1982)—as an umbrella term to describe robots that show unintended or inconsistent behavior, independent of the underlying cause. A Byzantine robot can appear well-functioning to some part of the swarm but faulty to others and might arbitrarily change its behavior. An extension of the LCP capable of managing these Byzantine robots is the weighted-mean-subsequence-reduced (W-MSR) algorithm (LeBlanc et al., 2013). While W-MSR's outlier detection limits the influence of Byzantine robots as long as their number is low, it breaks down as soon as their number is high or an attacking robot forges pseudo-identities (Sybil attack).

To pave the way for real-world deployments, *secure* robot swarms must continue to operate effectively in the presence of Byzantine robots, potentially performing Sybil attacks. Peer-to-peer networks are particularly prone to Sybil attacks: without a trusted system, it is easy for a malicious agent to create an unlimited number of new identities and gain a disproportionate amount of power in the swarm (Douceur, 2002). We contend that *blockchain technology* can be used to create such secure robot swarms due to its decentralized nature, resilience, and versatility. Blockchain technology was originally developed for Bitcoin (Nakamoto, 2008), the first widely successful digital peer-to-peer currency. In the context of Bitcoin, the blockchain presents a tamper-proof financial ledger in a network of mutually untrusting agents without relying on a central authority. The Ethereum framework (Buterin, 2014) further demonstrated that the blockchain cannot only be used for financial transactions but can store snippets of programming code and come to an agreement regarding their outcome. These snippets of programming code are called blockchain-based smart contracts (or *smart contracts* for short). Every node (robot in this article) in the network runs a virtual machine and executes these snippets. We show how smart contracts can provide the infrastructure for implementing secure “meta-controllers” in robot swarms.

Blockchain-based meta-controller: We define a blockchain-based meta-controller to be a controller that coordinates the swarm at a higher level than the local controllers of the

individual robots. To this end, crucial information from the individual robots is securely stored, aggregated, and processed via a smart contract residing on the blockchain. This ensures that information or control commands are based on a consensus in the swarm.

We release our developed framework as open-source software. It facilitates blockchain research in swarm robotics by providing an interface between the robot swarm simulator ARGoS (Pinciroli et al., 2012) and the blockchain framework Ethereum.

In this article, we study whether robot swarms need blockchain technology. To this end, we formulate the following research questions:

- RQ 1: Can smart contracts be used to replace existing consensus protocols in robot swarms?
- RQ 2: Can smart contracts be used to mitigate the effect of Byzantine robots in robot swarms?
- RQ 3: Can smart contracts introduce scarce resources into robot swarms and prevent Sybil attacks?

To address these research questions, we compare the two existing protocols LCP and W-MSR to our blockchain-based approach in a collective decision-making scenario (**Figure 1**) where the robot swarm moves on a floor covered with black and white tiles and has to determine the relative frequency of the white tiles in an ARGoS environment. The scope of this study is strictly limited to swarm robotics, where global communication is not available.

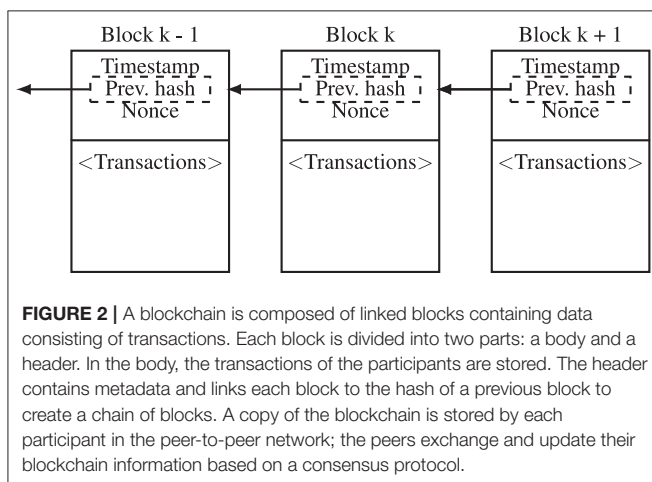
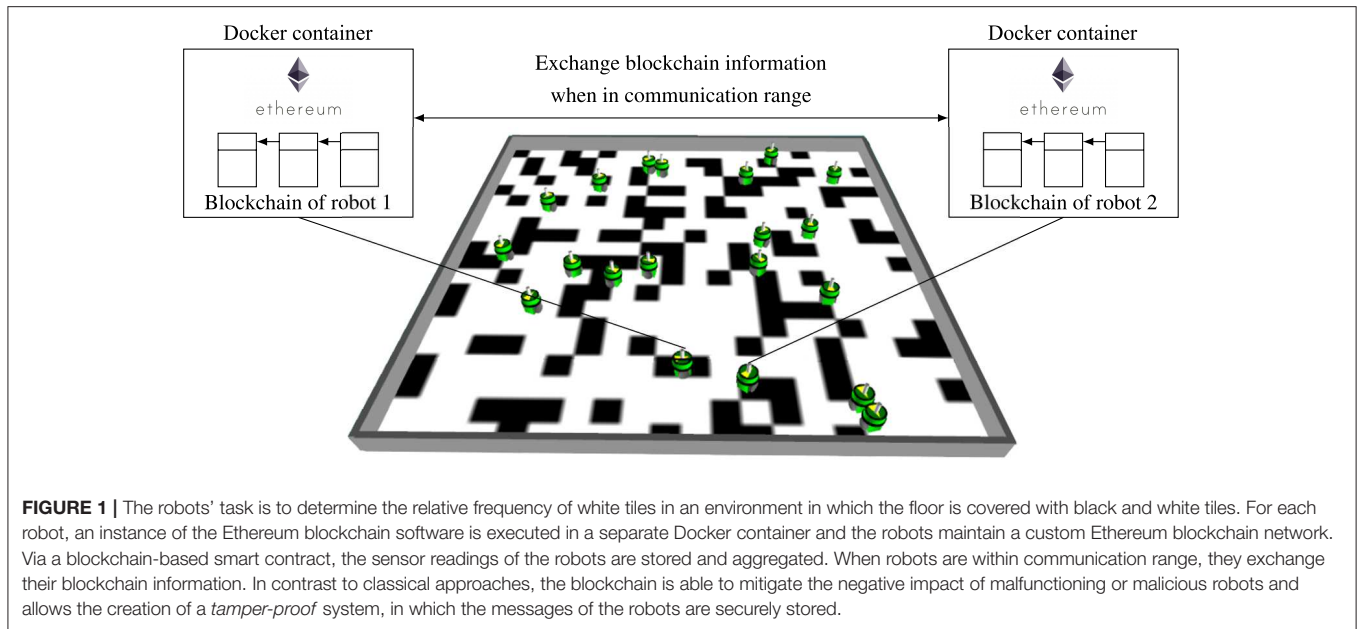
The remainder of this paper is structured as follows. Section 2 summarizes the fundamentals of blockchain technology. Section 3 reviews related work in consensus achievement, security issues, and blockchain-controlled robot swarms. Section 4 lays the foundation for practical implementations by describing the ARGoS-blockchain interface. Section 5 describes the general framework for conducting the simulations in ARGoS and the technical aspects of the used consensus protocols. Section 6 presents and discusses the results of five sets of simulations—in the presence and absence of Byzantine robots. Section 7 extends the discussion to robustness, feasibility, and scalability and draws directions for future work. Section 8 presents the conclusions.

2. FUNDAMENTALS OF BLOCKCHAIN TECHNOLOGY FOR SWARM ROBOTICS

This section summarizes the main characteristics of blockchain technology (section 2.1) and explains blockchain-based smart contracts (section 2.2).

2.1. General Foundation

Blockchains are databases and computing platforms that are replicated and shared among the participants (robots in this work) of a peer-to-peer network (**Figure 2**). The pseudonymous Satoshi Nakamoto originally devised the blockchain to record digital coin transactions (transactions of cryptotokens) of the cryptocurrency Bitcoin (Nakamoto, 2008). Shortly after, there have been proposals to use the decentralized ledger for other



specific, non-financial applications, such as voting, identity management, and supply chain management (Crosby et al., 2016). In 2014, Ethereum further generalized these use cases and released a framework for storing and executing programming code via blockchain technology (*blockchain-based smart contracts*) based on a Turing-complete programming language.

To interact with a blockchain and store new data, participants create transactions and distribute them among their peers. Examples of transactions are: “Send 5 ether (Ethereum’s cryptocurrency) from digital address A to B” or “Execute function X using Y as input.” A transaction is digitally signed by the sender using a private key. Hence, all transactions can be unambiguously assigned to a digital address (public key) and attackers cannot create transactions under a false digital identity. In most blockchain frameworks, all data is public and can be read by every participant of the network.

Still, in blockchains without an access control layer (public blockchains or permissionless blockchains), the real identities of entities (persons, organizations, robots) involved in a transaction can remain unknown since only the public keys are visible.

For a transaction to become part of the blockchain, it has to be bundled into a block and added to the end of the chain of blocks. Before being part of a block, transactions are called *unconfirmed* transactions and are disseminated across nodes of the blockchain network. Bitcoin introduced a consensus protocol which allows the participants in the network to agree on which blocks to add and in what order to add them. The consensus protocol used by Bitcoin is called Proof-of-Work (PoW) and was the first protocol to effectively reach decentralized consensus preventing at the same time double-spending (i.e., a situation where the same cryptocurrency is spent twice). PoW requires the participants to solve a computational puzzle in order to add a block to the blockchain; the puzzle consists of finding a hash value below a target value using the bundled transactions and an adjustable nonce value as input to the hash function. The nonce is a number that can be arbitrarily varied in order to change the input to the hash function and, therefore, the result of the hash function. The process of solving this puzzle (i.e., modifying the nonce value given a list of transactions and calculating the resulting hash values) is called *mining*. The number of hashes a device can compute per second is stated by its hash power. Miners are motivated to perform the PoW since the first one that finds a solution to the puzzle can append the corresponding block to the blockchain and as a consequence is rewarded by immutable cryptocurrencies stored on the blockchain. Due to delays in the communications between the network participants, the participants can have conflicting blockchain versions (*forks*). For example, during the experiments conducted in the scope of this research, the information written in the blockchain differs among

the robots that are not in communication range. However, via the PoW-based consensus protocol, conflicting blockchain versions can be resolved: whenever a robot has to choose between possible blockchains, the blockchain that required the highest PoW (i.e., the longest blockchain) gets accepted as the true blockchain, while shorter blockchains are discarded. Transactions that were in the discarded blockchains but not in the longest blockchain become unconfirmed transactions again and can be included in later blocks (Figure 3).

2.2. Blockchain-Based Smart Contracts

A blockchain-based smart contract (or a *smart contract* for short) is programming code that encapsulates variables and functions and is stored on the blockchain. To create a smart contract or call its functions, one needs to create a transaction and distribute it in the blockchain network. The nodes in the blockchain network keep track of the internal state (e.g., value of variables) and execute the computations of smart contracts, e.g., via the Ethereum Virtual Machine (EVM). While there are now multiple blockchain-based smart contract platforms, Ethereum remains the platform with the largest user base and the most mature technical setup.

Smart contracts were originally devised by Szabo (1997) to enforce contractual agreements between parties via computer protocols. Szabo's theoretical notion was made practically possible for the first time by the Ethereum framework: via a blockchain-based smart contract, a certain event can trigger an unstoppable financial transaction (programmable payment). However, blockchain-based smart contracts are not limited to programmable payments and the term smart contract is now used to describe any computer program that is executed on a blockchain.

For example, an Ethereum smart contract could provide the functions for selecting the winner of a talent show on TV. The audience has the possibility to vote for their favorite candidate (Alice or Bob) by sending a transaction (e.g., including 0.01 ether) to the TV's station smart contract. The smart contract on the public Ethereum blockchain keeps track of the number of votes for both candidates. Moreover, it specifies the following programmable payment: if the number of votes for one candidate reaches 100,000, the prize money of 1,000 ether is transferred to that candidate's Ethereum address. This example highlights some advantages of smart contracts in contrast to classical voting scenarios: (i) contract conditions and vote counts are transparent, (ii) existing votes cannot be manipulated or discarded, and (iii) the prize money will definitely be paid as soon as the condition is reached.

In order to use Ethereum smart contracts in swarm robotics, the target robotic platforms need to meet certain requirements in terms of communication, processing, and storage. The size of one Ethereum transaction is around 150 Bytes. In order to communicate with each other, robots should be able to send and receive some Kilobytes per seconds, otherwise, they may not be able to synchronize their blockchains in an adequate amount of time. During the simulations conducted in our research, the blockchain grew on average to 6.8 MB, a size

which could be stored on many state-of-the-art robots in swarm robotics¹.

3. RELATED WORK

This section first discusses consensus achievement in robot swarms (section 3.1), followed by work related to security issues (section 3.2), and concludes by reviewing existing work on blockchain technology used in swarm robotics (section 3.3).

3.1. Consensus Achievement

Consensus achievement problems in robot swarms can be divided into *discrete* and *continuous* problems (Valentini et al., 2017). Discrete problems can be formalized as best-of- n problems, where the swarm has to agree upon a choice among a finite set of n choices. Examples of discrete problems are path selection (Montes de Oca et al., 2011), site selection (Reina et al., 2014), and collective perception (Valentini et al., 2016a). In continuous problems, in contrast, the swarm's goal is to agree upon a choice among an infinite set of continuous choices. Examples of continuous problems are collective motion (Ferrante et al., 2012), spatial aggregation (Soysal and Sahin, 2005), and collective estimation (as studied in this work).

In this work, we study the influence of Byzantine robots on efficiently reaching swarm consensus in a continuous collective estimation problem. However, *exact consensus* in continuous problems is typically unattainable on spatially distributed robot systems (Elhage and Beal, 2010), since it would require each robot to agree upon exactly the same value. Connectivity limitations, large distances, local information, or different sensor readings, can hinder that progress. Although the blockchain can overcome this limitation, for the purpose of comparing our blockchain approach to existing approaches, we here only consider *approximate consensus*. This entails that each robot calculates a weighted local average based on its own estimates and those received from neighbors. A consensus has then been reached as soon as the difference between the maximum and the minimum value in the network is smaller than a given threshold. For the comparison, we selected the commonly used consensus algorithms LCP and W-MSR.

3.1.1. Linear Consensus Protocol

The linear consensus protocol (LCP) is the prevailing approach for achieving approximate distributed consensus (Beal, 2016) and has been used in a wide variety of use cases, such as formation control, flocking, and sensor fusion (Olfati-Saber and Murray, 2004; Xiao et al., 2005). The main idea is to reach approximate consensus on a set of beliefs held by the agents.

While this linear consensus protocol achieves high accuracies, it does not account for the presence of Byzantine agents. As a result, a single Byzantine robot keeping a constant value will make all non-Byzantine robots converge to that value (Gupta et al., 2006), potentially fully disrupting the functioning of the robot swarm. This confirms the insights and intuitions presented

¹Note that in this article, as said before, all experiments are run in simulation. Porting our system on real robots will be the subject of future work.

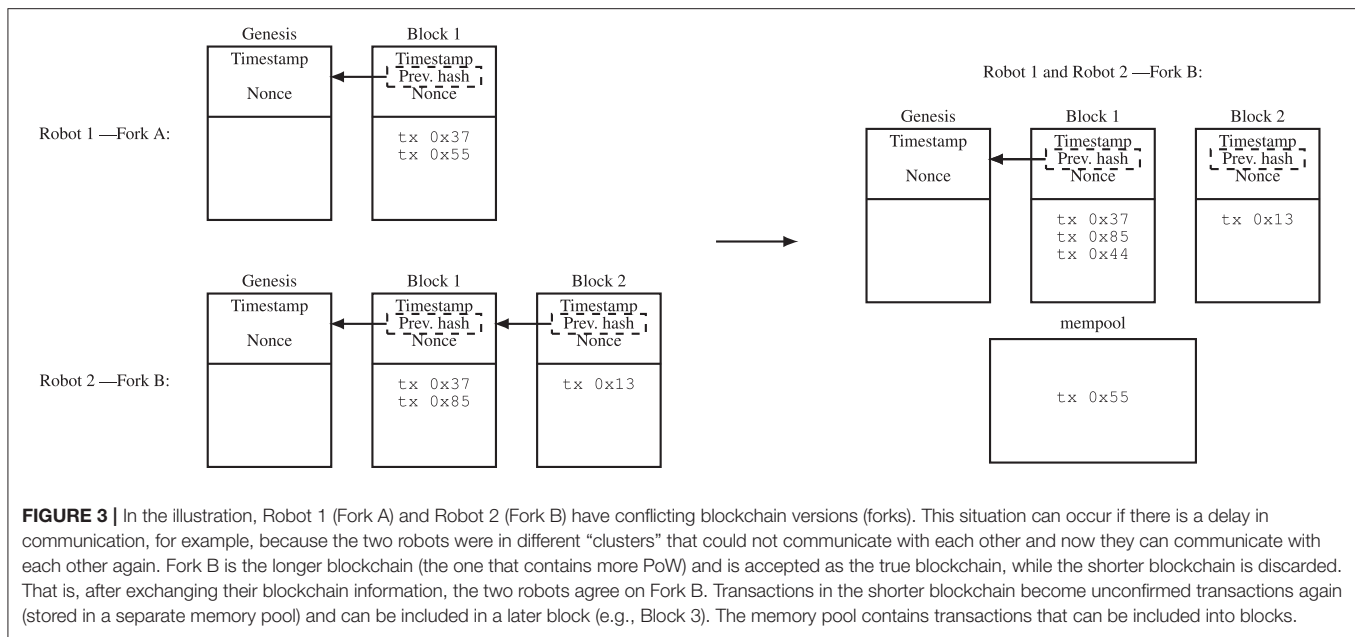


FIGURE 3 | In the illustration, Robot 1 (Fork A) and Robot 2 (Fork B) have conflicting blockchain versions (forks). This situation can occur if there is a delay in communication, for example, because the two robots were in different “clusters” that could not communicate with each other and now they can communicate with each other again. Fork B is the longer blockchain (the one that contains more PoW) and is accepted as the true blockchain, while the shorter blockchain is discarded. That is, after exchanging their blockchain information, the two robots agree on Fork B. Transactions in the shorter blockchain become unconfirmed transactions again (stored in a separate memory pool) and can be included in a later block (e.g., Block 3). The memory pool contains transactions that can be included into blocks.

by Winfield and Nembrini (2006) and Higgins et al. (2009) that fault tolerance in robot swarms cannot be taken for granted and that Byzantine robots can compromise the correct functioning of robot swarms.

3.1.2. W-MSR: Byzantine Approximate Consensus

To overcome the susceptibility to Byzantine interference, LeBlanc et al. (2013) introduced the weighted-mean-subsequence-reduced (W-MSR) algorithm as a Byzantine fault-tolerant extension of LCP. W-MSR is a state-of-the-art method for achieving resilient consensus in distributed sensor networks and robot swarms (Guerrero-Bonilla et al., 2017; Saldaña et al., 2017).

The functioning of W-MSR is based on outlier detection: given a design parameter F , the algorithm discards the smallest and the largest F values received from neighbors, including the agent's own belief. A limitation of the algorithm is that in order to select a proper value for the parameter F it assumes that the agents have knowledge of the network topology or that they are able to sustain a desired connectivity through control algorithms, such as flocking (Saulnier et al., 2017). However, this is not always possible in robot swarms since robots might become sparsely connected due to changes in the topology of the network (e.g., due to movements, failing units, or communication problems). As we will show later, W-MSR fails if the number of Byzantine robots is greater than F or when confronted with Sybil attacks.

3.2. Security Issues in Swarm Robotics

At the outset of swarm robotics research, robot swarms were assumed to be fault-tolerant by design, due to the large number and redundancy of the robot units (Dorigo et al., 2004; Millard et al., 2014). While this assumption holds true in some cases, it has been increasingly called into question when researchers began to study explicit fault detection (Winfield and Nembrini, 2006).

A distinction has been made between endogenous and exogenous fault detection. In endogenous fault detection, robots detect faults in themselves; in exogenous fault detection robots detect faults in other robots (Christensen et al., 2009). In early robotics research, most work was devoted to endogeneous fault detection (see for example, Roumeliotis et al., 1998; Christensen et al., 2008). However, it can be difficult to detect certain endogeneous faults, e.g., a robot might have a broken sensor but only realize it if its sensor readings are compared to its neighbor robots. Therefore, more recently swarm robotics research shifted its focus to exogeneous fault detection. Christensen et al. (2009) present a robot swarm whose robots are programmed to flash their LEDs in synchrony. LED flashing indicates correct functioning of a robot. Therefore, broken robots are easily identified by their non-flashing LEDs and this identification is made easy by the fact that flashing is synchronized across the robot swarm. A disadvantage of this system is that it can only detect robots that are either completely broken or that report an endogeneous error by not flashing their LED anymore: malicious robots cannot be detected nor is exogeneous *partial* fault detection possible. Yet, Winfield and Nembrini (2006) argue that complete failures (e.g., power failure) are significantly less severe than partial failures (e.g., motor failure, communication failure, and sensor failure). One reason for this is that partially failed robots can still unfavorably interact with the remaining robots. For example, because of a broken sensor, they could send wrong sensor readings to other robots, misleading the rest of the swarm. The authors point out that future research should focus on the detection of partial failures; this is what we do in this article.

In the first survey on security issues in robot swarms, Higgins et al. (2009) identify tampered swarm members or failing sensors, attacked or noisy communication channels, and loss of availability as the main threats to robot swarms. Tarapore et al.

(2015, 2017, 2019) address the detection of *faulty* robots in both simulated and physical robot swarms. Their method is based on outlier detection using the bioinspired crossregulation model. To this end, robots exchange their behavior vectors. Outliers (faulty robots) are detected by comparing the behavior vectors to other behavior vectors in the swarm: if the majority of the swarm has the same behavior vector, this behavior is classified as an inlier, otherwise as an outlier. While this approach does not require a priori knowledge about abnormal behavior, it assumes that every robot shares its behavior vector truthfully.

Security issues related to external factors, such as attacks on the swarm, only started to be studied recently. For example, Zikratov et al. (2016) propose a reputation-based management system where robots keep trust levels about each other based on the correct execution of a predefined protocol. Sargeant and Tomlinson (2016) study a wider range of attacker strategies, such as eavesdropping, data manipulation, and denial of service in robot swarms. Primiero et al. (2018) show that the propagation of deceitful information through the swarm can be prevented if robots probabilistically change their belief.

In contrast to the systems presented above, the blockchain is capable of logging events in a tamper-proof way and of implementing generic meta-controllers. Moreover, all of the above-mentioned systems are susceptible to attacks: e.g., using the LED flashing method of Christensen et al. (2009), an attacker can flash its LEDs in synchrony but send wrong sensor values to the remaining swarm members. The other systems that rely on wireless messages are susceptible to Sybil attacks: without a trusted third-party, it is always possible for a malicious agent to create an unlimited number of new identities in peer-to-peer networks (Douceur, 2002). Through this large number of identities, an attacker can gain a disproportionate amount of power (Gil et al., 2017), potentially causing much damage, e.g., in voting scenarios. The blockchain can prevent Sybil attacks from disrupting swarm behavior by introducing *scarcity* to decentralized systems: a robot wanting to exert influence must pay for this by spending a scarce resource (cryptotokens). It is thus, not the number of entities forged but rather an attacker's wealth that determines the success of the attack.

3.3. Related Work on Blockchain Technology in Robot Swarms

In swarm robotics research, it is often assumed that robots do not have access to shared knowledge. This is mainly due to three reasons: (i) it could be unfeasible to set up the infrastructure for such a shared knowledge system; e.g., if the robots are in a remote area and scattered throughout a large physical space; (ii) the shared knowledge system could represent an unacceptable single point of failure; and (iii) it might be computationally too complex to process all incoming and outgoing data in a single system. However, robot swarms could greatly benefit from shared knowledge, for example, for determining whether a consensus has been reached within the swarm, for calculating the mean value of the sensor readings of the single robots, or for determining malfunctioning units. Hence, decisions could be based on a shared view of the world. This would not only possibly

simplify several swarm robotics tasks but also enlarge their field of applications facilitating decision processes.

Castelló Ferrer (2016) was the first to describe a variety of use cases for using a blockchain in robot swarms, such as secure communication, data logging, and consensus agreement. Strobel et al. (2018) delivered the first proof-of-concept, using the blockchain framework Ethereum and the robot swarm simulator ARGoS in a binary collective decision scenario. The authors show how a blockchain-based meta-controller improves the quality of the collected sensor data by providing a blockchain security layer on top of existing algorithms developed by Valentini et al. (2016a). The meta-controller detects inconsistencies in a robot's behavior when it deviates from the agreed-upon behavior and excludes it from the swarm. In contrast, prior collective decision-making algorithms could not reach a consensus whenever one or more robots in the swarm are Byzantine.

Fernandes and Alexandre (2019) and Lopes and Alexandre (2019) study the use of blockchain technology for the registration of robotic events (e.g., robot *x* finished job *y*) in industrial scenarios, where the different robots might come from different manufacturers. The authors additionally demonstrate the use of blockchain-based smart contracts for anomaly detection. However they do not assume local time-delayed communication and maintenance of the blockchain among the robots but rather use the blockchain as an external computing platform. Other work addressed obstacles that might hinder the use of blockchain-based controllers in real-world applications. McAbee et al. (2019) discuss how blockchain technology can help to solve problems in military intelligence applications. Nishida et al. (2018) outline an approach to reduce the blockchain size for information sharing in swarm robotics systems by storing the hash of data—in their case image data—in the blockchain instead of the information itself.

The work presented in this article is based on two previous works (Strobel and Dorigo, 2018; Strobel et al., 2018). However, it is significantly extended: (i) instead of solely determining if there are more black or white tiles (i.e., a binary decision task), in the present work, the swarm's goal is to determine the relative frequency of white tiles expressed as a value between 0.0 and 1.0—a collective *estimation* scenario which yields more information and might be more interesting for real-world deployments; (ii) as soon as a consensus on a specific value is reached, the experiment can be stopped in a fully decentralized way via the consensus mechanism of the blockchain; (iii) in the present article, we study different distributions of the features of the scenarios; (iv) we show how the blockchain limits the number of messages a robot can send, thus preventing Sybil attacks; (v) we present the ARGoS-blockchain interface which enables researchers to test and extend the presented scenarios on different platforms.

4. ARGOS-BLOCKCHAIN INTERFACE

The ARGoS robot simulator (Pinciroli et al., 2012) is the state-of-the-art research platform to conduct simulations in swarm robotics. In our research, each robot acts as an Ethereum

blockchain node, maintaining a custom Ethereum network. In order to connect ARGoS and Ethereum, we developed the ARGoS-Blockchain interface that provides access to the Ethereum nodes for the robots (Figure 4). The interface is intended to facilitate research in blockchain-based robot swarms by allowing to call Ethereum functions in ARGoS. Additionally, Docker makes it easy to install and run the interface on different platforms. The interface is available on GitHub².

The implementation of the custom Ethereum network is based on Capgemini AIE's Ethereum Docker³. Docker containers (Merkel, 2014) contain all the necessary dependencies to run specific applications and are more lightweight than a virtual machine. In our setup, for each robot, the Ethereum implementation *geth* is executed in a separate Docker container. The simulated robots maintain a *custom* Ethereum network, i.e., a network that is shared among the simulated robots and independent of Ethereum's main network. Different containers can communicate with each other via channels.

In order to execute an Ethereum function (e.g., create a new smart contract) from ARGoS, a robot uses its C++ controller to attach to the Docker container. The Docker containers provide shell scripts⁴ with customizable templates (e.g., one of the templates compiles the smart contract, uses the binary code to send a blockchain transactions, and waits until the contract is mined). Via Ethereum's IPC (interprocess communications) interface, the shell scripts execute the Ethereum functions.

We use an auxiliary "bootstrap" node for publishing the smart contract to the blockchain at the beginning of each run of the simulations (Figure 5). The bootstrap node then mines the smart contract and sends the contract address and the ABI (application binary interface; the ABI specifies which functions a smart contract provides and how to call them) to the controllers of the robots. As soon as this is done, the bootstrap node is removed from the network. The bootstrap node is not necessarily required and the smart contract could also be created by a robot. However, we used an auxiliary node to make sure (i) that the smart contract is available at the start of the actual experimental run and (ii) that robots have the same initial conditions in all experiments.

The experiments were conducted on a computer cluster. To simulate the limited hardware of real robots, one core with 2.0 GHz and 1.8 GB of memory was assigned to each Docker container⁵. The communication channels between the Docker containers were only established when robots were within a 50 cm communication range in order to simulate the local communication capabilities of real robots.

²<https://github.com/Pold87/ARGoS-Blockchain-interface>

³<https://github.com/Capgemini-AIE/ethereum-docker>, accessed on November 6, 2019

⁴The interface uses shell scripts, since, during development, it became evident that they are executed much faster than other Ethereum APIs.

⁵This is a reasonable choice as a robot's computer could easily have such characteristics. It is also a convenient choice because on a computer with 2.0 GHz and 1.8 GB of RAM, Ethereum works "out-of-the-box," without any modifications; therefore, any interested user can obtain the most recent release of Ethereum from the official depository and use it with our publicly available ARGoS-Blockchain interface.

5. MATERIALS AND METHODS

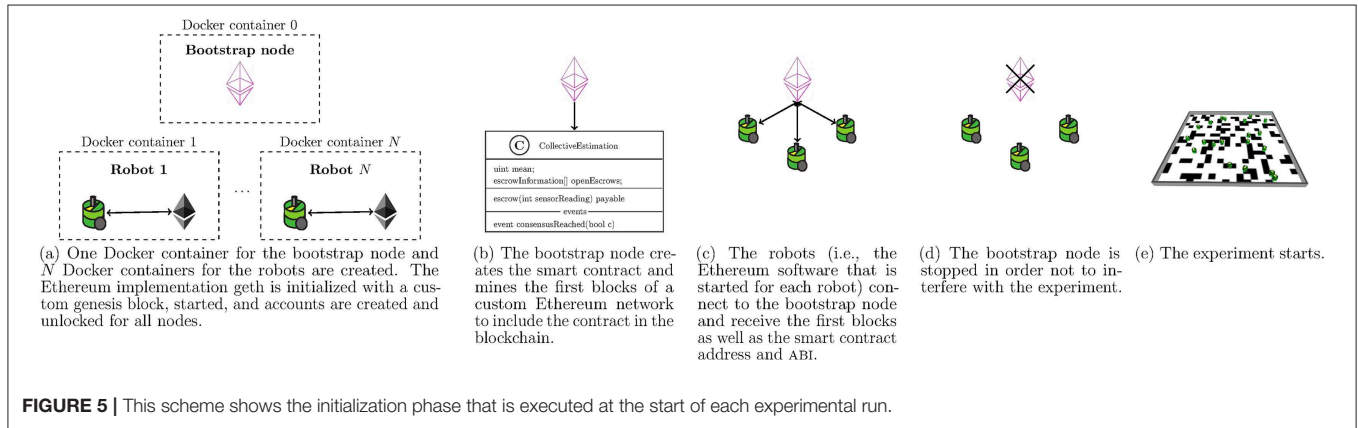
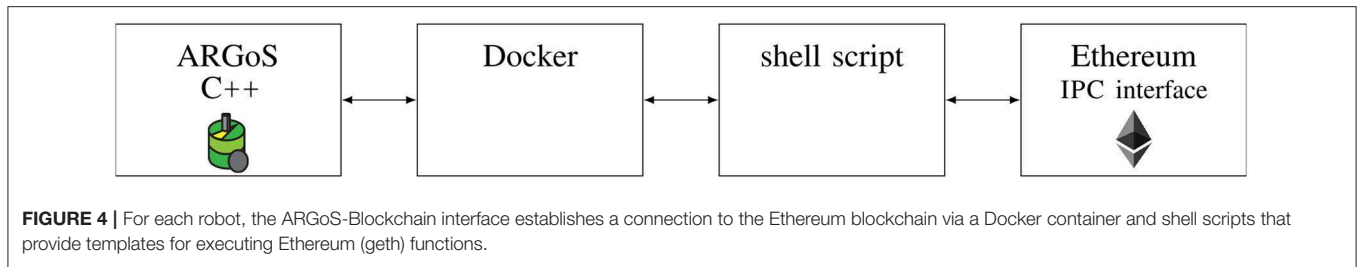
5.1. Setup of the Simulations

We compare three consensus algorithms (LCP, W-MSR, and blockchain) in terms of their general performance and resilience to an increasing number of Byzantine robots. To this end, $N = 20$ robots are used in the robot swarm simulator ARGoS (Pinciroli et al., 2012). The swarm's goal is to estimate the relative frequency of white tiles in a $2 \times 2 \text{ m}^2$ "checkerboard" environment where the floor is covered with B black and W white tiles of size $10 \times 10 \text{ cm}^2$, $B + W = 400$ (Figure 1). The checkerboard environment, obstacle avoidance, and random walk movement routines were developed in earlier work by Valentini et al. (2016a). We replicate their parameters for the random walk and obstacle avoidance routines. Depending on the scenario, the positions of the black and white tiles are either fixed by the experimenter or selected randomly at the beginning of a simulation run. The starting positions of the robots are randomly chosen from a uniform distribution at the beginning of each simulation run. To enable the swarm to aggregate information about the environment, each robot samples its local ground sensor and exchanges information with other robots in their communication range. The experiment is conducted in discrete time steps with one time step corresponding to 1 s. At each time step, a robot i determines if it is above a black or a white tile via its ground sensor. Each robot works in exploration *phases*. We use the subscript notation i, m for variables referring to a robot i in its m th exploration phase. The duration of each exploration phase is $d = 45 \text{ s}$. To obtain a sensor reading, a robot i in its m th exploration phase calculates the ratio $\hat{\rho}'_{i,m}$ between the number of white tiles $\hat{W}_{i,m}$ and the total amount of tiles $\hat{W}_{i,m} + \hat{B}_{i,m}$ it sensed in this exploration phase: $\hat{\rho}'_{i,m} = \frac{\hat{W}_{i,m}}{\hat{W}_{i,m} + \hat{B}_{i,m}} \in [0, 1]$. If the distance between two robots is $< 50 \text{ cm}$, they are in communication range and can exchange information, in accordance with real swarm robotics systems that have only local communication capabilities. This communication range leads to an average degree of connectivity of 2.4 (i.e., one robot is, on average, connected to 2.4 other robots) and yields multiple non-connected clusters that exist almost all the time. For the different approaches, 40 simulation runs (i.e., repetitions) were performed for each value of the independent variable. These are the common characteristics for all three consensus protocols. The peculiarities of the different consensus protocols are given in section 5.2.

5.2. Implementation of the Different Consensus Models

5.2.1. Linear Consensus Protocol

Using the linear consensus protocol (LCP), each robot keeps track of a frequency estimate $\hat{\rho}_{i,m}$ that represents its belief about the relative frequency of white tiles. At the end of the first exploration phase ($m = 0$), the frequency estimate is set to the sensor reading of the first phase: $\hat{\rho}_{i,0} = \hat{\rho}'_{i,0}$. The frequency estimate is then updated at the end of each 45 s exploration phase m by



incorporating the frequency estimates $\hat{\rho}_{j,m-1}$ of the neighbors \mathcal{N}_i (Figure 6):

$$\hat{\rho}_{i,m} = w_{ii}\hat{\rho}'_{i,m} + \sum_{j \in \mathcal{N}_i} w_{ij}\hat{\rho}_{j,m-1}, \quad (1)$$

where $w_{ii} = w_{ij} = \frac{1}{|\mathcal{N}_i|+1}$ is a weight factor, assigning each message an equal weight, as done in related work (e.g., Saulnier et al., 2017). In the phase $m + 1$, a robot i distributes its frequency estimate $\hat{\rho}_{i,m}$ to other robots in communication range, i.e., robots communicate their frequency estimates and not their current sensor readings (the sensor readings fluctuate from phase to phase and consensus achievement would be difficult if these values were used). As in the work by Valentini et al. (2016a), each robot has an identifier and only one message can be received from any specific robot in each phase. In order to store received messages, robots have a buffer size of $M = N - 1 = 19$. If more messages are received, only the last M messages are stored. The buffer size $M = 19$ makes sure that every robot is able to receive a message from every other robot in each exploration phase but small enough so that it represents a mechanism to prevent flooding of the network.

5.2.2. W-MSR

The W-MSR algorithm is a variant of LCP and introduces a means for detecting and discarding outliers. It also uses Equation (1) to obtain a consensus but first performs outlier detection. To do so, the outliers are removed from the set of neighbors. The algorithm requires a design parameter F that should be selected based on the assumed number of Byzantine robots and connectivity of the

network. We set $F = 2$. Then, all received values $\hat{\rho}_{j,m-1}$ larger than $\hat{\rho}'_{i,m}$ are sorted in ascending order. If there are fewer than F values larger than $\hat{\rho}_{i,m}$, all of them are added to the set of outliers \mathcal{O} . Otherwise, the F largest values are considered outliers. The same procedure is applied to all values smaller than $\hat{\rho}'_{i,m}$. To update the frequency estimate, the W-MSR algorithm then uses $\mathcal{N}' = \mathcal{N} \setminus \mathcal{O}$ instead of \mathcal{N} in Equation (1).

5.2.3. Blockchain Approach

The blockchain approach is based on a smart contract that aggregates the sensor readings of the robots into the frequency estimate $\hat{\rho}_t$, while discarding outliers and rewarding robots for contributing to the scenario (Figure 7). To be consistent with the classical approaches, we will use the notation $\hat{\rho}_{i,m}$ to indicate the estimated frequency of white tiles as written in the blockchain of robot i in its m th exploration phase, but will otherwise write $\hat{\rho}_t$ to indicate the frequency estimate in escrow round t (see below for a description of the escrow).

Using the blockchain approach, the robots' sensor information is stored and aggregated using a smart contract at given time intervals (Figure 6). Each robot keeps a local copy of the blockchain; if robots are physically close to each other, they exchange their blockchain information. The setup uses the ARGoS-Blockchain interface described in section 4. In order to simulate the local communication capabilities of real robots, the simulated robots have the ability to connect to each other's Ethereum processes via the Docker container if their distance is smaller than 50 cm; they can then exchange blocks and unconfirmed transactions of the blockchain. To synchronize ARGoS and Ethereum, the experiments were conducted in real time.

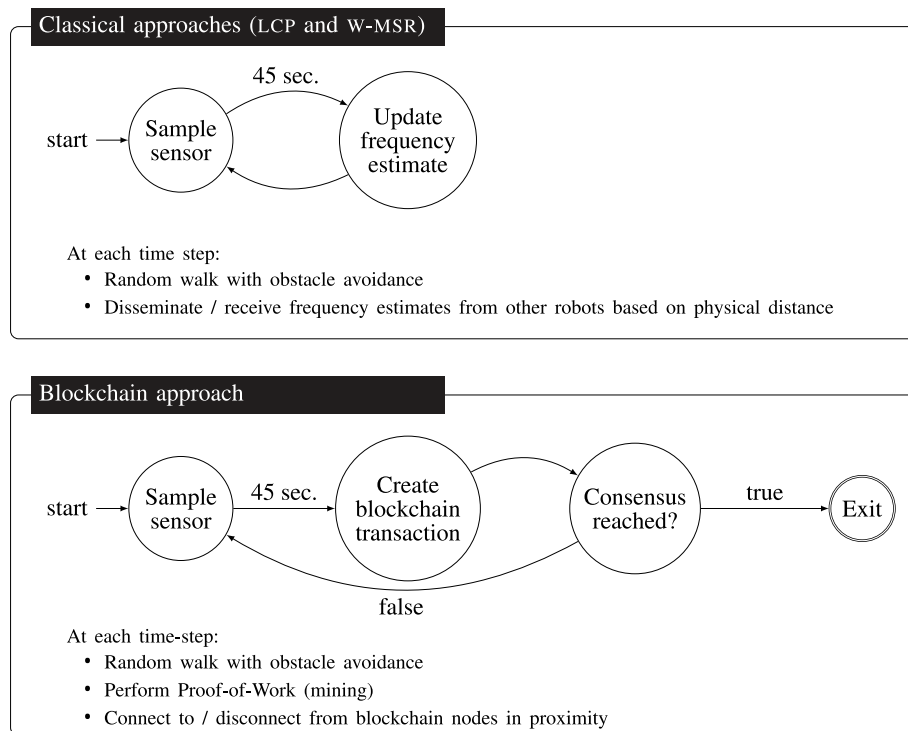


FIGURE 6 | The robots explore the environment using a random walk routine and sample their ground sensors. Using the classical approaches LCP and W-MSR (**top**), they update their frequency estimate every 45 s (i.e., every 45 time-steps) via Equation (1). Using the blockchain approach (**bottom**), every 45 s, the robots create a blockchain transaction that includes their sensor reading. In contrast to the classical approaches, with the blockchain approach, the robots can check whether a consensus has been reached by querying the state (`true` or `false`) of the smart contract event `consensusReached`. If `true`, they enter the exit state and stop creating blockchain transactions. Then, they still perform the random walk and connect to other robots in their proximity to exchange blockchain information.

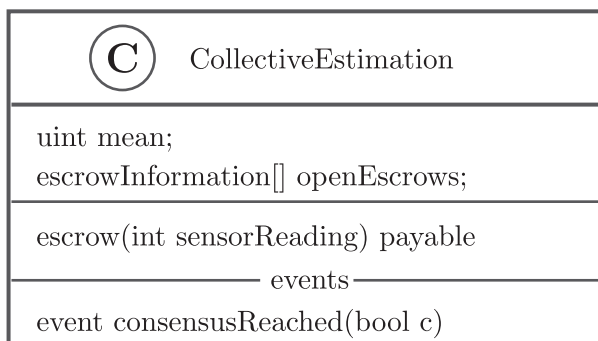


FIGURE 7 | The smart contract keeps track of the frequency estimate $\hat{\rho}_t$ and provides the function `escrow` to send a sensor reading $\hat{\rho}'_{i,m}$. The event `consensusReached` is set to `true` when the frequency estimate does not change more than τ from one escrow round to the next one.

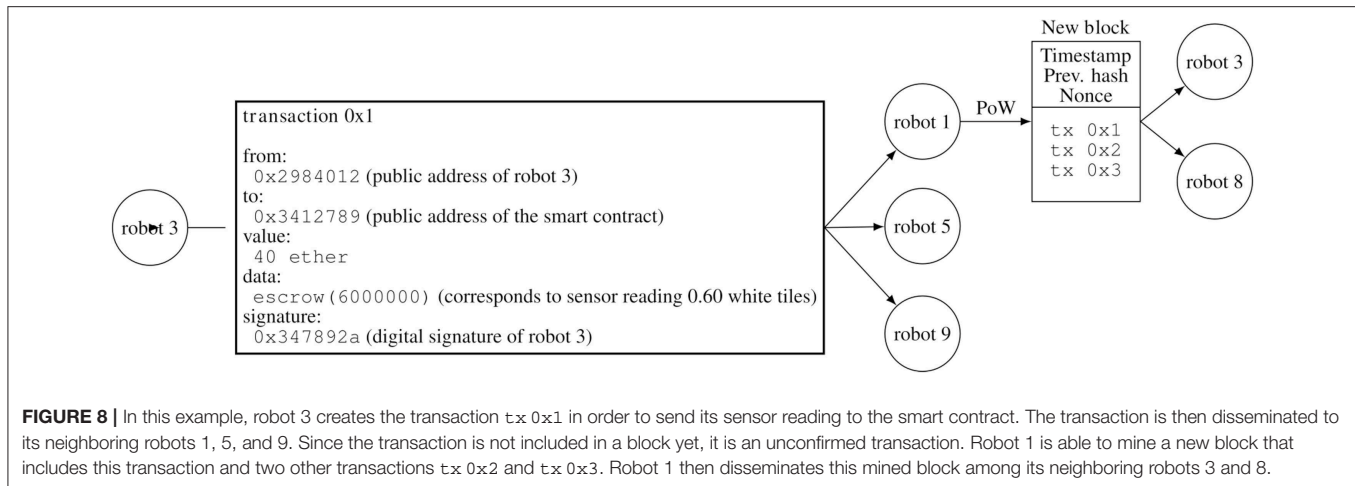
Each robot *mines*, i.e., it performs the Proof-of-Work, from the start to the end of a simulation run. Every time a robot successfully solves a block, it is rewarded by 5 ether (Ethereum's cryptocurrency⁶). In the beginning of each experimental run, all robots have a balance of 0 ether. Since creating blockchain

⁶Since we do not use the main Ethereum network but a custom network maintained by the robots, these ethers have value only within the robot swarm.

transactions requires ether, robots have to mine blocks to gain ether and be able to send transactions to the smart contract. The robots start with 0 ether so that we do not need to identify beforehand which robots will be part of the experiment. This builds a basis for “open robot swarms” (e.g., for citizen science projects) where robots are free to join and leave at any time during an experiment.

We specified an initial difficulty of the mining puzzle in the genesis block, so that the swarm mines approximately one block per second, resulting in 2.25 blocks per robot after 45 s. Therefore, the average balance after 45 s is $2.25 \times 5 \text{ ether} = 11.25 \text{ ether}$. This means that after 45 s most of the time none of the robots have enough ether to submit a transaction. Note that it is possible for the robots to mine empty blocks, i.e., blocks without any transactions, and still get the reward of 5 ether for solving the block.

At the end of each exploration phase m (i.e., after 45 s), each robot sends its sensor reading $\hat{\rho}'_{i,m}$ to the smart contract via the function `escrow(int sensorReading)` (**Figure 7**) and the value gets stored in the list `openEscrows`. That is, to store a sensor reading in the blockchain (**Figure 8**), a robot (i) creates a blockchain transaction which includes its sensor reading in the data part of the transaction, (ii) adds a deposit amount of q ether, (iii) signs this transaction, and (iv) disseminates this transaction among its neighboring robots. The function `escrow` accepts a value between 0.0 and 1.0, which



stands for the sensor reading $\hat{\rho}'_{i,m}$ of the robot i . Since smart contracts in Ethereum accept integer values only, in the actual implementation, all sensor readings are multiplied by 10^7 to simulate rational numbers between 0.0 and 1.0 (e.g., instead of sending 0.30, a robot would send 0.30×10^7). The deposit amount q is intended to limit the number of sensor readings a robot can send. That is, a robot “vouches” for its sensor reading. When the robots send transactions, they do not check whether they possess enough ether or not: in case they do not have enough ether, the transaction is simply discarded by the smart contract. We set $q = 40$ ether, a suitable value as determined in a pilot experiment.⁷

The goal of the escrow is to collect sensor readings and to reward robots that sent meaningful sensor data. As soon as the length of openEscrows is equal to $V = 20$, a new disbursement round t is performed, i.e., outliers are identified and inliers are rewarded. To this end, the difference between $\hat{\rho}_t$ (frequency estimate in the smart contract in disbursement round t) and $\hat{\rho}'_{i,m}$ (sensor readings from the individual escrow transactions) is determined. If the absolute difference $|\hat{\rho}'_{i,m} - \hat{\rho}_t|$ is smaller than a threshold ϵ , the sensor reading is accepted, otherwise it is discarded. Accepted values of $\hat{\rho}'_{i,m}$ are called inliers, discarded ones are called outliers. The value of the mean $\hat{\rho}_t$ is obtained by calculating the mean of all inliers over all escrow rounds t . In every new escrow round, it is updated via a one-pass algorithm to reduce the computational requirements. The execution of the smart contract includes activities such as verifying the validity of the transaction and, when it has received $V = 20$ valid transactions, to compute the mean. The smart contract is executed every time a block is mined (as long as it includes transactions) but the computation of $\hat{\rho}_t$ usually happens with a lower frequency.

In the first round ($t = 0$)—i.e., in the time interval from the beginning of the experiment to the moment in which the smart

contract has received 20 valid transactions—when no frequency estimate $\hat{\rho}_t$ is available yet, all values of $\hat{\rho}'_{i,m}$ are accepted. The value of ϵ is a tuning parameter that influences how much the current mean in the blockchain can change from one round to the other. Decreasing ϵ will increase the sensitivity (the number of Byzantine votes that are correctly identified as outliers), while increasing ϵ will increase the specificity (the number of non-Byzantine votes that are correctly included in the calculation of the current mean). We set $\epsilon = 0.2$, a suitable value as determined in a pilot experiment. The value V is another tuning parameter: lower values lead to earlier results for $\hat{\rho}_t$ (since the value is only updated at the end of an escrow round) but also to an increased risk that the ratio between the number of Byzantine robots and normal robots is high in an escrow round. If the value of V is set too high, the detection of Byzantine robots may start too late and they might have already caused a significant damage and non-Byzantine robots may have to wait long until they get back their deposit amount. We set the list length to $V = 20 = N$ since then, on average, every robot will be represented by one vote in each round.

In order to incentivize robots to take part in the escrow, inliers get a reward r_t in ether. The reward r_t is greater or equal to the escrow value and calculated by distributing the collected ether of the escrow round among the inliers: $r_t = Vq/in_t = 20 \times 40 \text{ ether}/in_t$, where in_t is the number of inliers at round t . Hence, robots can gain ether by mining, thereby improving the network's security, or by sending sensible sensor values, helping to determine the correct frequency of white tiles. This creates an implicit reward mechanism within the swarm that discourages Byzantine robots to operate as such, since sending wrong sensor measures costs cryptotokens.

5.3. Software Availability

The implementation of the presented classical approaches⁸ and blockchain approach⁹ are hosted on GitHub.

⁷In a real-world scenario, determining a suitable price might be difficult. However, there is a simple but effective remedy: instead of sending a transaction every 45 s, robots could send a transaction as soon as they have enough ether to create a transaction. We did not implement this “remedy” because we wanted the three compared approaches to differ in as few aspects as possible.

⁸<https://github.com/Pold87/robot-swarms-need-blockchain-classical>

⁹<https://github.com/Pold87/robot-swarms-need-blockchain>

5.4. Statistics

Let $\mathcal{R} = 1, 2, \dots, N$ be the set of all robots, \mathcal{G} be the subset of non-Byzantine robots (mnemonic: \mathcal{G} for “good”) and \mathcal{B} be the subset of Byzantine robots (mnemonic: \mathcal{B} for Byzantine or “bad”), with $\mathcal{G} \cup \mathcal{B} = \mathcal{R}$ and $|\mathcal{B}| = k$. An asterisk $*$ indicates a randomly selected robot from the set \mathcal{R} and the infinity symbol ∞ indicates that the value is determined at the end of an experimental run. Therefore, $\hat{\rho}_{*,\infty}$ is the estimated frequency of white tiles of one randomly selected robot at the end of an experimental run and $\hat{\rho}_{\mathcal{G},\infty} = \sum_{i \in \mathcal{G}} \hat{\rho}_{i,m} / |\mathcal{G}|$ is the arithmetic mean of the frequency estimate of all non-Byzantine robots at the end of an experiment. The frequency estimate $\hat{\rho}_{\mathcal{B},\infty}$ indicates the mean of the frequency estimate of a run where the number of Byzantine robots was zero (\mathcal{B}_0). We use the median $\hat{\rho}_{\mathcal{B}_0,\infty}$ as a baseline value to compare the performance of the approaches, when the number of Byzantine robots is increased. The baseline values are determined separately for the LCP, W-MSR, and blockchain approaches.

The following statistics are used to compare the performances of the three approaches:

- **Absolute error AE_* .** This statistic is the absolute value of the difference between the actual relative frequency of white tiles ρ and the frequency estimate $\hat{\rho}_{*,\infty}$ of a randomly selected robot at the end of an experimental run: $AE_* = |\rho - \hat{\rho}_{*,\infty}|$. AE_* measures the predictive capacity of the different approaches. For the calculation of AE_* , we randomly select one robot since we assume that a consensus has been reached. Compared to averaging the values of all or several robots, this approach is closer to real-world scenarios where only a single functioning robot might be retrieved after the end of an experiment; additionally, it might be too time-consuming or costly to sample all the robots.
- **Harm.** This statistic measures the amount of harm that Byzantine robots cause to non-Byzantine robots. The idea is that we compute the difference between (i) $AE_{\mathcal{G}}$, that is, the average absolute error of the non-Byzantine robots in presence of Byzantine robots, and (ii) $\widetilde{AE}_{\mathcal{B}_0}$, that is, the median of the average absolute error over all runs with zero Byzantine robots: $\text{harm} = AE_{\mathcal{G}} - \widetilde{AE}_{\mathcal{B}_0}$. Note that to calculate the harm we take on an “omniscient perspective” and assume that we are able to distinguish between Byzantine and non-Byzantine robots. That is, for analysis purposes, here we consider the case when it is possible to retrieve all the robots and identify those that are non-Byzantine after the experiment.
- **Consensus time T_N .** This statistic is the time in seconds until all robots have reached a consensus on a certain estimated frequency of white tiles (see section 6.2.2).

For the blockchain approach, additionally, the following statistic is measured:

- **Blockchain size BC_{MB} .** This statistic indicates the blockchain size in MB of one randomly chosen robot, determined at the end of each experimental run.

For all plots showing the absolute error AE_* and the Harm in the presence of Byzantine robots, we additionally perform

locally estimated scatterplot smoothing (LOESS¹⁰) indicated by blue curves in the graphs. The gray bands around the blue LOESS curve indicate the 95% confidence interval for predictions from the regression. The LOESS curve is intended to make it easier to spot the general trend when the number of Byzantine robots is increased.

6. SIMULATIONS

In this section, we compare the three approaches (LCP, W-MSR, blockchain) in five experiments under different conditions (Table 1). The experiments are structured along the three research questions introduced in section 1 and correspond to the complexity and intelligence of Byzantine robots.

- **Baseline:** Experiment 1 is intended to establish a baseline and does not contain any Byzantine robots. It tests the three different approaches in an environment with randomly distributed tiles. The goal of the experiment is to provide a proof-of-concept and show that the approaches work as intended in standard conditions.
- **Byzantine Robots:** Experiments 2–4 introduce Byzantine robots. While there are many possible Byzantine failures, in this work we study a case where each Byzantine robot disseminates a frequency estimate of $\hat{\rho}_{i,m} = 0.0$ for the classical approaches and accordingly $\hat{\rho}'_{i,m} = 0.0$ for the blockchain approach in all exploration phases m , independent of its actual sensor readings. This choice is motivated by two reasons: (1) a value of 0.0 is the worst-case scenario and maximizes the difference between ρ and $\hat{\rho}_{i,m}$ and (2) it is a failure mode studied in other research (e.g., Gupta et al., 2006). We vary the number of Byzantine robots between 0 and 7.¹¹
- **Sybil attack:** Experiment 5 then introduces clearly malicious Byzantine robots that perform Sybil attacks. The malicious robots still disseminate frequency estimates of $\hat{\rho}_{i,m} = 0.0$ and $\hat{\rho}'_{i,m} = 0.0$. However, they try to send as many messages as possible by creating new identities at every time step. The goal of this experiment is to show that just one malicious robot suffices to let existing approaches fail.

6.1. Comparison in Absence of Byzantine Robots

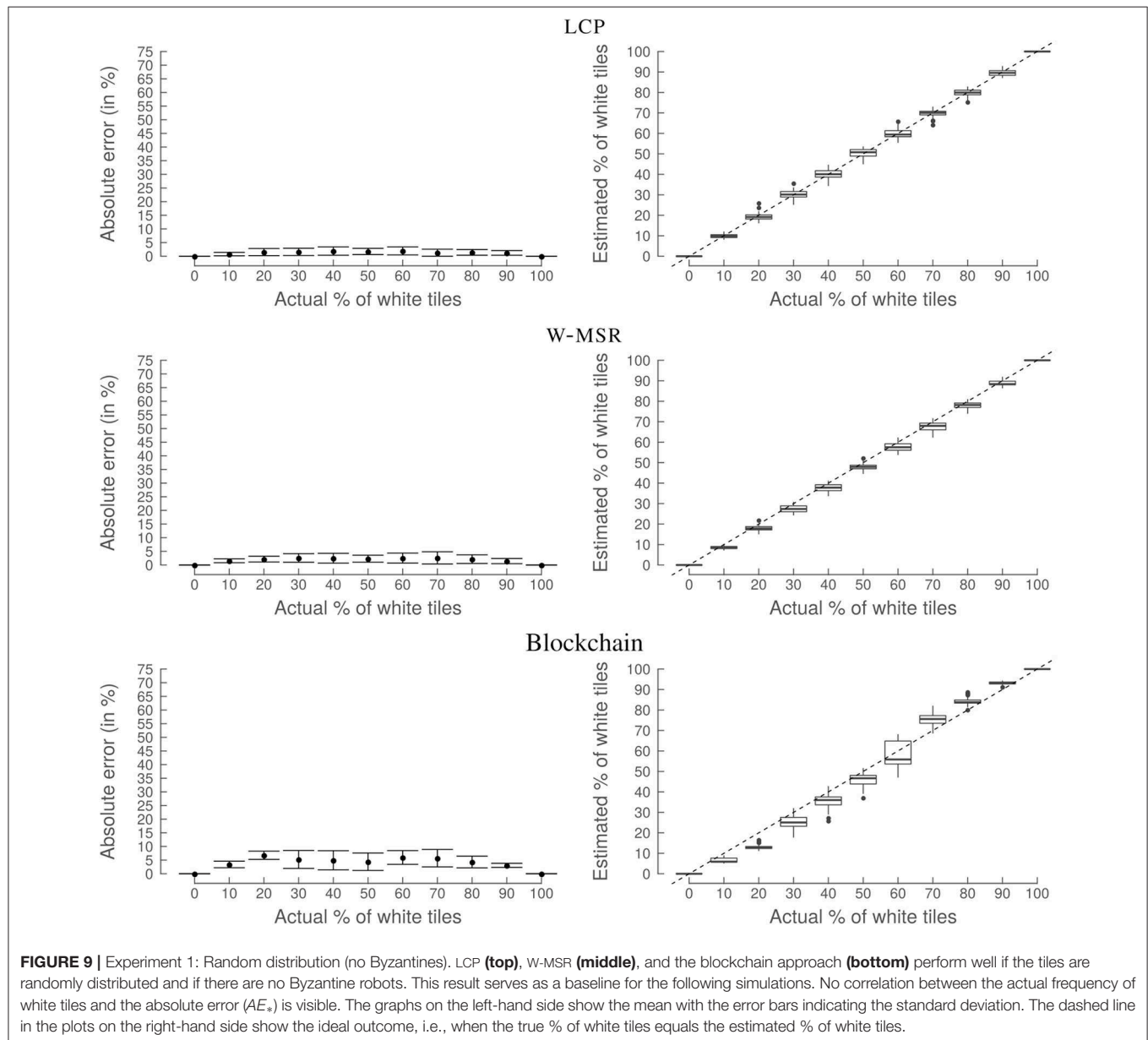
In the first experiment, we compare the values of AE_* for the different approaches without the presence of Byzantine robots. To this end, the percentage of white tiles is increased from 0 to 100% in steps of 10%. A simulation run is stopped after 1,000 seconds. The goal of this experiment is to (1) determine if

¹⁰LOESS smoothing (Jacoby, 2000) is a non-parametric regression method to fit non-linear data. To do so, the LOESS algorithm performs local linear regressions via a weighted sliding-window approach. In other words, for each point in the dataset (the current focal point), it takes a subset of the whole dataset (in our case the 75% nearest neighbors) to calculate the least-squares fit. The higher the distance to the focal point, the lower the weight in the least-squares fit. We used the default settings as provided by the R programming language, described at <https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/loess> (accessed February 6, 2020).

¹¹We set the maximum number of Byzantine robots to 7 since related literature usually considers a maximum of 33 % Byzantine agents.

TABLE 1 | Overview of the experiments.

No.	Experiment	% White tiles	# Byzantines	Tile mixing	Sybil attack	Exit criterion
1	Random distribution (no Byzantines)	0, 10, ..., 100	0	Yes	No	1,000 s
2	Random distribution	75	0, 1, ..., 7	Yes	No	1,000 s
3	Consensus	75	0, 1, ..., 7	Yes	No	Threshold below τ
4	Binary distribution	75	0, 1, ..., 7	No	No	1,000 s
5	Sybil attack	75	0, 1, ..., 7	Yes	Yes	1,000 s



the blockchain-based approach can replace existing approaches, (2) establish a baseline for successive experiments, and (3) see if all approaches are able to deal with a straightforward experimental setup.

Results, Discussion, and Interpretation

The three approaches perform well with a mean absolute error lower than 0.08 (**Figure 9**) and are, therefore, able to successfully perform the desired task. However, the blockchain approach

presents a slightly higher variability and mean absolute error for some values of the actual % of white tiles. This is because the blockchain approach has more random factors—due to the Proof-of-Work and the specific security measure implemented in the smart contract—than the classical approaches. The overall good performance serves as a baseline for the following scenarios.

6.2. Comparison in Presence of Byzantine Robots

In the next three simulations, we study the influence of Byzantine robots (robots that disseminate $\hat{\rho}_{i,m} = 0.0$ for the classical approaches and $\hat{\rho}'_{i,m} = 0.0$ for the blockchain approach) on the performance of the different approaches. The number of Byzantine robots is increased from 0 to 7. The frequency of white tiles in the environment is fixed to 75 %. We chose 75 % because it is in the middle between 50 % and 100 %, i.e., it contains a bias for one color to rule out that a random approach might work.

6.2.1. Byzantine Robots in a Random Environment

In this experiment, the influence of Byzantine robots on the value of AE_* is studied in an environment with randomly distributed tiles. A simulation run is stopped after 1,000 s. The goal of this experiment is to investigate how the Byzantine robots affect the different approaches. With an increasing number of Byzantine robots, we expect LCP to break down fairly quickly due to its lack of security measures. In contrast W-MSR and the blockchain approach should be more resilient as long as the number of Byzantine robots remains low.

Results, discussion, and interpretation

The LCP approach is not designed to be resilient to the presence of Byzantine robots; accordingly, a strong increase in its AE_* can be observed when the number of Byzantine robots increases (Figure 10). In contrast, by design, W-MSR is resilient to the presence of Byzantine robots, as long as their number is low. However, both approaches have a high standard deviation, partially due to the high number of extreme outliers where the AE_* is 75%. This is due to the fact that AE_* is computed by randomly selecting a robot from the swarm. When the number of Byzantine robots increases the probability of selecting a Byzantine robot increases. While different choices of W-MSR's design parameter F would lead to different values for AE_* , the percentage of extreme outliers would stay the same (since the Byzantine robots do not follow the protocol); additionally, in a real-world scenario one would not be able to know whether the selected robot is Byzantine or not.

The blockchain approach is resilient also to a higher number of Byzantine robots. In contrast to the classical approaches, even if a Byzantine robot is selected, the AE_* stays low. This is due to the consensus protocol of the blockchain, i.e., all robots agree on the longest chain and even the Byzantine robots share the same estimate written in the blockchain.

Particularly interesting is the harm value of the LCP. It starts with a median of more than 10% for one Byzantine robot. In other words, the estimated frequency of all non-Byzantine robots is already 10% worse compared to the baseline, if just 5% of the robots (1 out of 20) are Byzantine. The harm can also be negative, in cases when the Byzantine robots help to get closer to the

actual ρ . This is the case for the blockchain approach. Without Byzantine robots, the blockchain approach overestimates the frequency of white tiles due to the implemented security measure: since the smart contract only accepts values within $\hat{\rho}_t - \epsilon < \hat{\rho}'_{i,m} < \hat{\rho}_t + \epsilon$, several $\hat{\rho}'_{i,m}$ values from non-Byzantine robots will be discarded. Therefore, the addition of a small number of Byzantine robots reduces the absolute error and the harm. This is a characteristic of the specific smart contract and different values of ϵ or a different outlier detection method (e.g., taking the standard deviation into account) would lead to different results.

6.2.2. Consensus Agreement in the Presence of Byzantine Robots

In this experiment, the influence of Byzantine robots on the swarm's ability to reach a consensus is studied. The goal of this experiment is to investigate if a swarm can reach a consensus in a fully decentralized way.

For the classical approaches (LCP and W-MSR), we say that a consensus in the swarm has been reached once the absolute difference between the highest $\hat{\rho}_{i,m}$ and the lowest $\hat{\rho}_{j,m}$ in the swarm is smaller than a threshold value τ . However, as soon as there is one “stubborn” Byzantine robot that keeps a constant frequency estimate, consensus of all robots can only be on that value when using the classical approaches. In our case, if the robots would come to a consensus, the only possible value would be 0.0, therefore, the expected absolute error would be 75% for the classical approaches, resulting in a useless frequency estimate of the swarm. For this reason, we show the consensus time for the classical approaches only in the *absence* of Byzantine robots.

For the blockchain approach, consensus is reached, if the frequency estimate between two escrow rounds does not change more than τ , i.e., $|\hat{\rho}_t - \hat{\rho}_{t-1}| < \tau$. The blockchain event `consensusReached` is then set to `true`. At the end of each exploration phase, each robot queries the status of this event. If the status of the event is `true` for all robots, the simulation run is stopped. For this experiment, we use the consensus threshold $\tau = 0.02$.

Results, discussion, and interpretation

The top row in Figure 11 shows the comparison of the three approaches in the absence of Byzantine robots. All approaches perform well and are able to reach a consensus in a reasonably short amount of time. The consensus time of the W-MSR and blockchain approaches is higher than the baseline LCP approach. Hence, there is a trade-off between consensus time in the absence of Byzantine robots and the level of security an approach provides.

The bottom row in Figure 11 shows the absolute error and consensus time of the blockchain approach. The consensus time rises slightly when the number of Byzantine robots increases. Similarly, the absolute error also increases, but the mean of the AE_* remains at about 20% even with seven Byzantine robots.

The blockchain-controlled swarm could reach a decentralized consensus, even in the presence of Byzantine robots. Therefore, it is autonomous and resilient, while the classical approaches are not. In addition—even without Byzantine robots—it is difficult for the classical approaches to determine whether each robot actually agrees on a certain value. Note that the classical

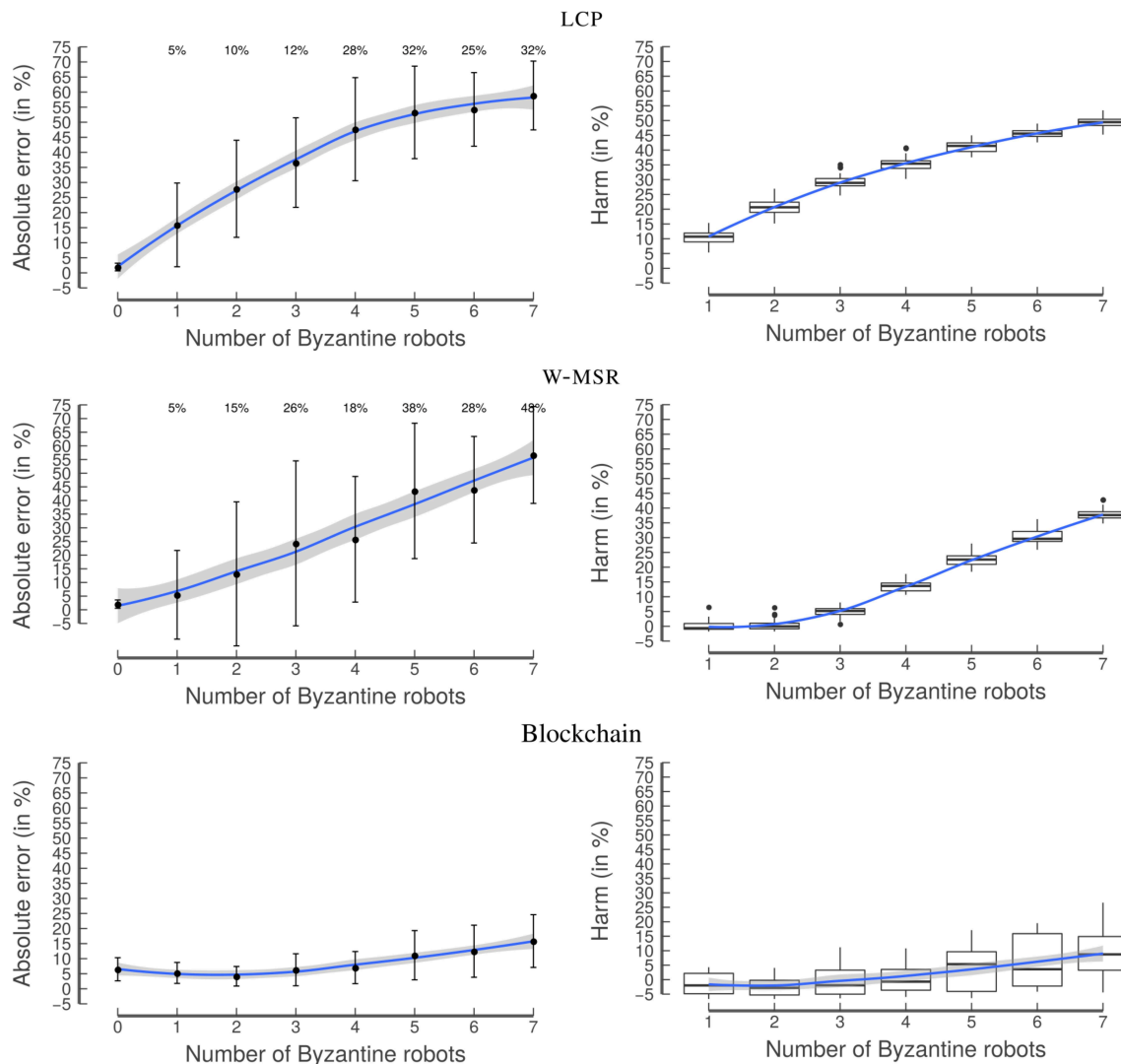


FIGURE 10 | Experiment 2: Random distribution. When the number of Byzantine robots increases, LCP's performance (**top**) quickly deteriorates and the frequency of extreme outliers becomes high (the percentages at the top of each graph correspond to the frequency with which Byzantine robots were selected when calculating AE_*). Therefore, with the classical approaches, one is always exposed to the risk of getting a completely wrong result, even if there is just one Byzantine robot in the swarm. W-MSR (**middle**) is able to manage a few Byzantine robots but its AE_* quickly increases when there are more than three of them. The blockchain approach (**bottom**) is largely unaffected by the increasing number of Byzantine robots, and does not contain extreme outliers. The graphs on the left-hand side show the mean with the error bars indicating the standard deviation. The blue line is obtained by locally estimated scatterplot smoothing (LOESS), the gray band around the blue line shows the 95% confidence interval for predictions from the LOESS regression.

approaches could be extended, so that robots in the swarm send a consensus signal to their neighbors when they have reached convergence; however, this signal would be prone to Byzantine robots sending a negative consensus signal. In practice, an external observer might be needed but this observer would represent a single point of failure and in some cases it might even be impossible to set it up. In contrast, in the case of the blockchain approach, the consensus determination is done *on-chain* (i.e., via a blockchain-based smart contract) without any external observer.

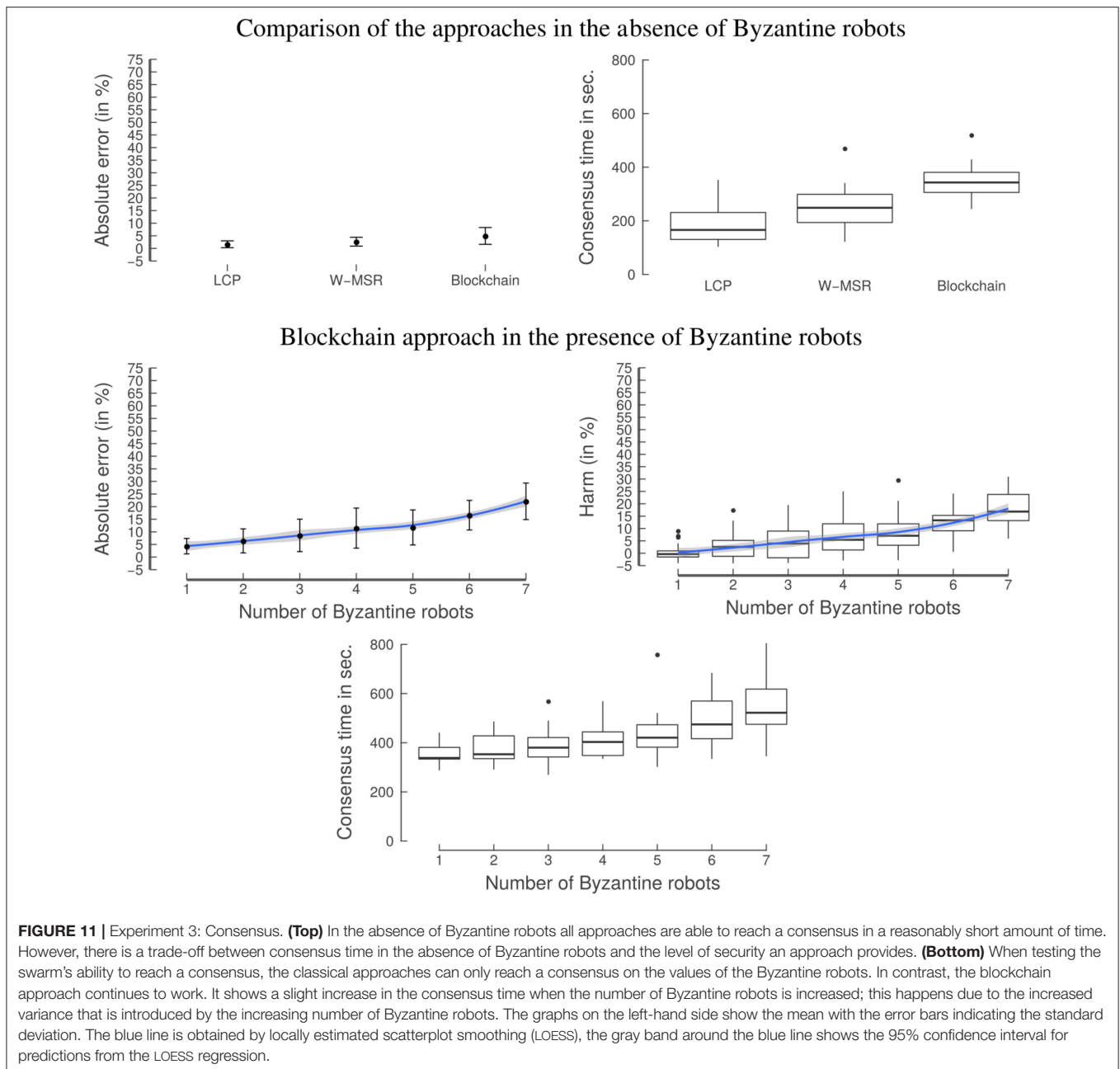
6.2.3. Byzantine Robots in a Binary Environment

In this experiment, the influence of Byzantine robots on the value of AE_* is studied in an environment with a fixed

distribution of tiles (**Figure 12**). Using the fixed distribution, the tiles in the left part of the environment are black (25%), while those in the right part are white (75%). A simulation run is stopped after 1,000 s. The goal of this experiment is to investigate whether the modified distribution of tiles makes the detection of outliers more difficult since also non-Byzantine robots will get extreme sensor readings of $\hat{\rho}'_{i,m} = 0.0$ and $\hat{\rho}'_{i,m} = 1.0$.

Results, Discussion, and Interpretation

While LCP's AE_* quickly increases with an increasing number of Byzantine robots, the W-MSR approach is able to manage a few Byzantine robots, starting with a relatively high AE_* of 10% (**Figure 13**).



When no Byzantine robots are part of the swarm, LCP performs better than W-MSR and the blockchain approach. This is because of the security measures implemented in W-MSR and in the blockchain approach, which have difficulties in distinguishing between the values generated by the Byzantine and by the non-Byzantine robots. However, in contrast to W-MSR, the blockchain's performance remains approximately constant, even for a rather high number of Byzantine robots. The harm distribution is similar to Experiment 2.

These results show that there is no "one size fits all" of consensus protocols; instead, there is a trade-off between adding security measures to approaches and their ability to perform

well under all circumstances. However, in real-world scenarios, we will almost certainly have to deal with Byzantine robots, therefore, using the blockchain approach is still warranted.

6.3. Comparison in Presence of Sybil Attacks

In the last experiment, we study the case in which Byzantine robots perform a Sybil attack. The goal of this experiment is to investigate how decentralized swarms can deal with robots that forge multiple identities. The tiles are randomly distributed and a simulation run is stopped after 1,000 s.

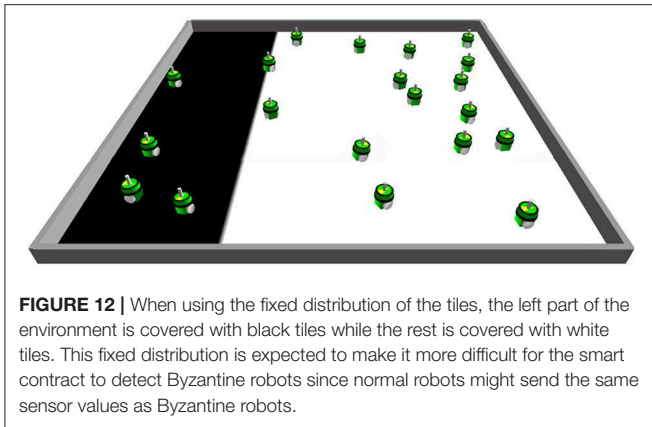


FIGURE 12 | When using the fixed distribution of the tiles, the left part of the environment is covered with black tiles while the rest is covered with white tiles. This fixed distribution is expected to make it more difficult for the smart contract to detect Byzantine robots since normal robots might send the same sensor values as Byzantine robots.

To perform a Sybil attack, the Byzantine robots are programmed as follows. In the classical approaches, every Byzantine robot creates a new identity at every time step and uses it to disseminate its sensor readings. In the blockchain approach, robots do not create new identities since these identities would not have any ether; therefore, the Sybil attack would be prevented automatically. We could have programmed Byzantine robots to first create new public addresses (i.e., identities) and distribute their ether among these addresses but since the public addresses are not used in the identification of outliers, this was not deemed necessary. Additionally, this would most likely weaken the Sybil attack, since first distributing the tokens would slow down the process. Instead, in the blockchain approach, a Byzantine robot sends as many transactions as possible. However, the limiting factor is that sending transactions costs cryptotokens, that is, robots have to send 40 ether every time they send an escrow transaction that contains their sensor reading (section 5.2.3).

Results, Discussion, and Interpretation

As expected, the classical approaches have high values for AE_* and harm as soon as one robot in the swarm is able to perform a Sybil attack (Figure 14). In stark contrast, in the blockchain approach, the Sybil attack is not successful since the 40 ether robots have to deposit to send a transaction prevents the robots from creating a high number of transactions. In other words, the robots cannot “spam” or “flood” the network with transactions since they would quickly run out of ether. The robots also cannot steal the identity of other robots (spoofing attack) due to digital signatures. Therefore, the blockchain approach stays resilient, even in the presence of a relatively high number of Byzantine robots. Based on these results, one of the main advantages of this approach is visible: the blockchain is able to introduce *scarcity* into a decentralized swarm, making the system more secure.

7. GENERAL DISCUSSION

In this work, we set out to study whether robot swarms need blockchain technology. To this end, we considered the open research problem of consensus reaching in robot swarms for the

general case of Byzantine robots and the more specific case of Sybil attacks. To answer the three research questions listed in the introduction of this article, we used a collective estimation task and compared the blockchain approach to existing consensus protocols. Our simulation results support a positive answer to our research questions: in the absence of Byzantine robots, consensus could be reached as effectively with blockchain-based smart contracts as with existing consensus protocols in robot swarms (RQ 1); the use of smart contracts indeed mitigates the influence of Byzantine robots in robot swarms (RQ 2); and Sybil attacks were prevented when using the blockchain approach (RQ 3). Below, we discuss the implications and limitations of our research.

7.1. Implications

The results of our experiments can be generalized in two ways: across use cases and across platforms. We showed that it is possible to implement meta-controllers with blockchain-based smart contracts. In our experiments, a meta-controller (i) aggregated the sensor readings from the individual robots, (ii) performed simple, yet effective outlier detection to manage Byzantine robots, and (iii) determined if a consensus was reached in the swarm, even in the presence of Byzantine robots.

The provided use case was intended to be a simple and easy to understand example of how a smart contract can be used in swarm robotics. Therefore, we used one of the simplest outlier detection methods. As our goal is to provide a proof-of-concept for blockchain-coordinated robot swarms, we did not strive for the best performance by fine-tuning algorithm parameters. For example, the approach could be extended and improved with more sophisticated outlier detection methods. Since smart contracts are Turing-complete, any outlier detection method is in principle implementable; in practice, however, one should choose a lightweight algorithm with a low run-time. Another aspect to consider is the operability of the approach in a dynamic environment. In the current implementation, the smart contract obtains a rough estimate in the first escrow round and then narrows down the collective estimate; a sudden change in the environment (e.g., the color of all the tiles is suddenly inverted) could lead to a dead-end situation, where all sensor readings in future escrow rounds are discarded by the outlier detection mechanism. However, in an improved version of the smart contract, one could, for example, always accept a minimum number of sensor readings per escrow round—even if they are outliers—to prepare the algorithm for dynamic environments. It is important to note that there is no need for adapting the robots’ controllers when changing the outlier detection method in the smart contract.

Although we selected a specific scenario and task (consensus reaching in collective estimation), this result is promising for the field of swarm robotics in general: using smart contracts as meta-controllers might facilitate the implementation of various other existing and novel swarm robotics applications. To list concrete examples, besides the presented collective decision-making scenario, we believe a blockchain-based approach might be useful in:

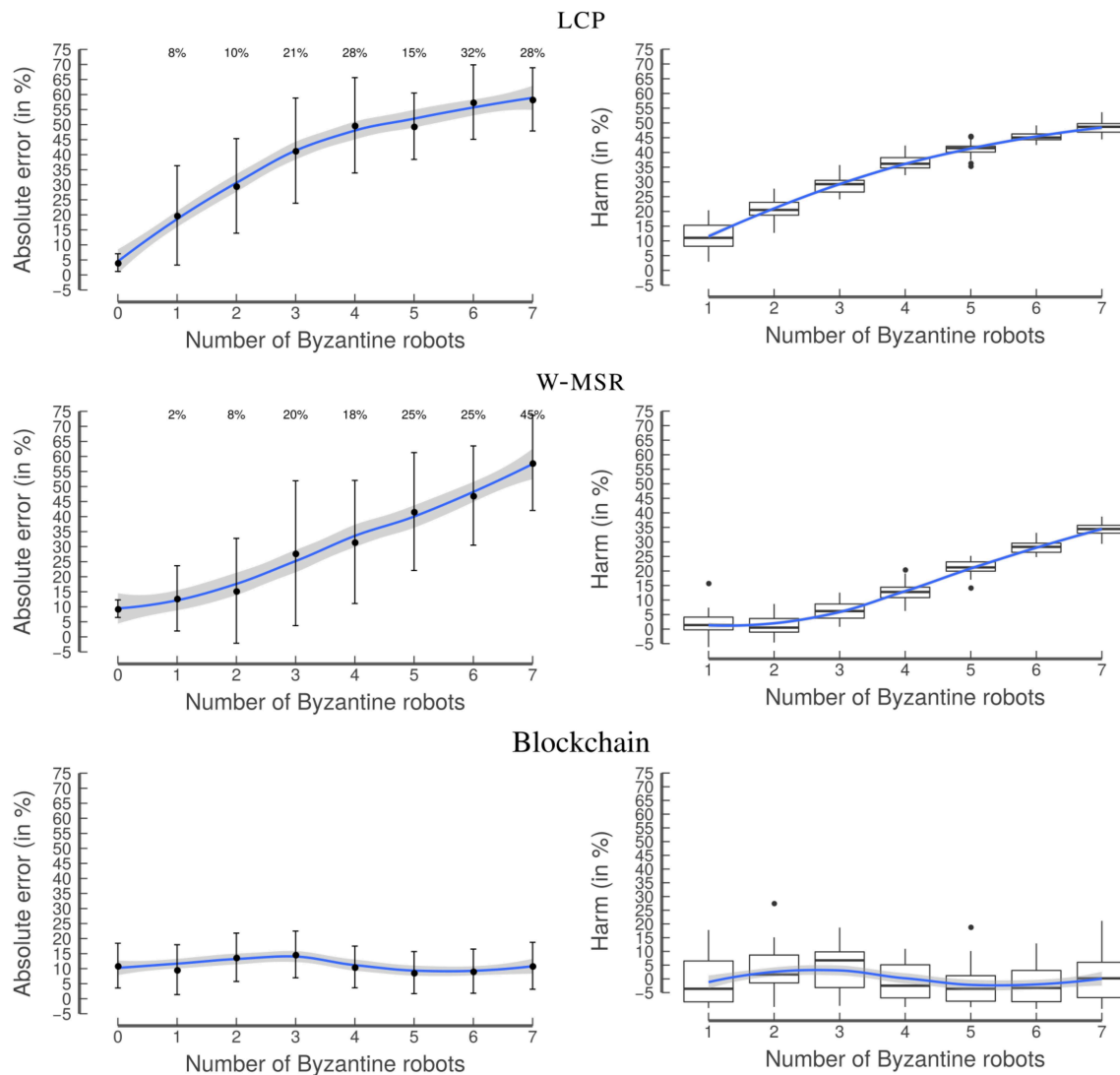


FIGURE 13 | Experiment 4: Binary distribution. Similar to the random distribution, LCP (top) shows a steep increase in AE_* , W-MSR (middle) can handle a few Byzantine robots, and the blockchain approach (bottom) is also resilient to a higher number of Byzantine robots. The graphs on the left-hand side show the mean with the error bars indicating the standard deviation. The blue line is obtained by locally estimated scatterplot smoothing (LOESS), the gray band around the blue line shows the 95% confidence interval for predictions from the LOESS regression.

- *task-allocation scenarios*: e.g., in an area exploration scenario, a smart contract could identify unexplored areas and send control commands to different robots to explore these areas;
- *reputation management*: the tamper-proof nature of the blockchain allows for maintaining reputation values for the different robots;
- *lightweight machine learning algorithms*: e.g., a smart contract could serve as a database for sensor data and train a classification algorithm;
- *collective mapping*: parts of a map could be stored and aggregated in a smart contract;
- *robot-to-robot economies*: e.g., auction-based approaches, where the auction is executed via a smart contract;
- *robot-to-human economies*: e.g, people could pay robots for executing a task (monetization of jobs, leading to *robot as a service*) or, vice versa, people could offer rewards for the completion of a task.

In addition to considering other use cases, it is also possible to consider swarms composed of entities that are not robots. In this sense, this work can be seen as a stepping-stone for swarms composed of people, Internet-of-Things devices, and/or vehicles.

A blockchain is tamper-proof due to its decentralized consensus protocol that is able to maintain scarce resources in decentralized systems. In our research, we showed that these scarce “cryptotokens,” i.e., immutable units of exchange stored in the blockchain, can be used to prevent Sybil attacks in open

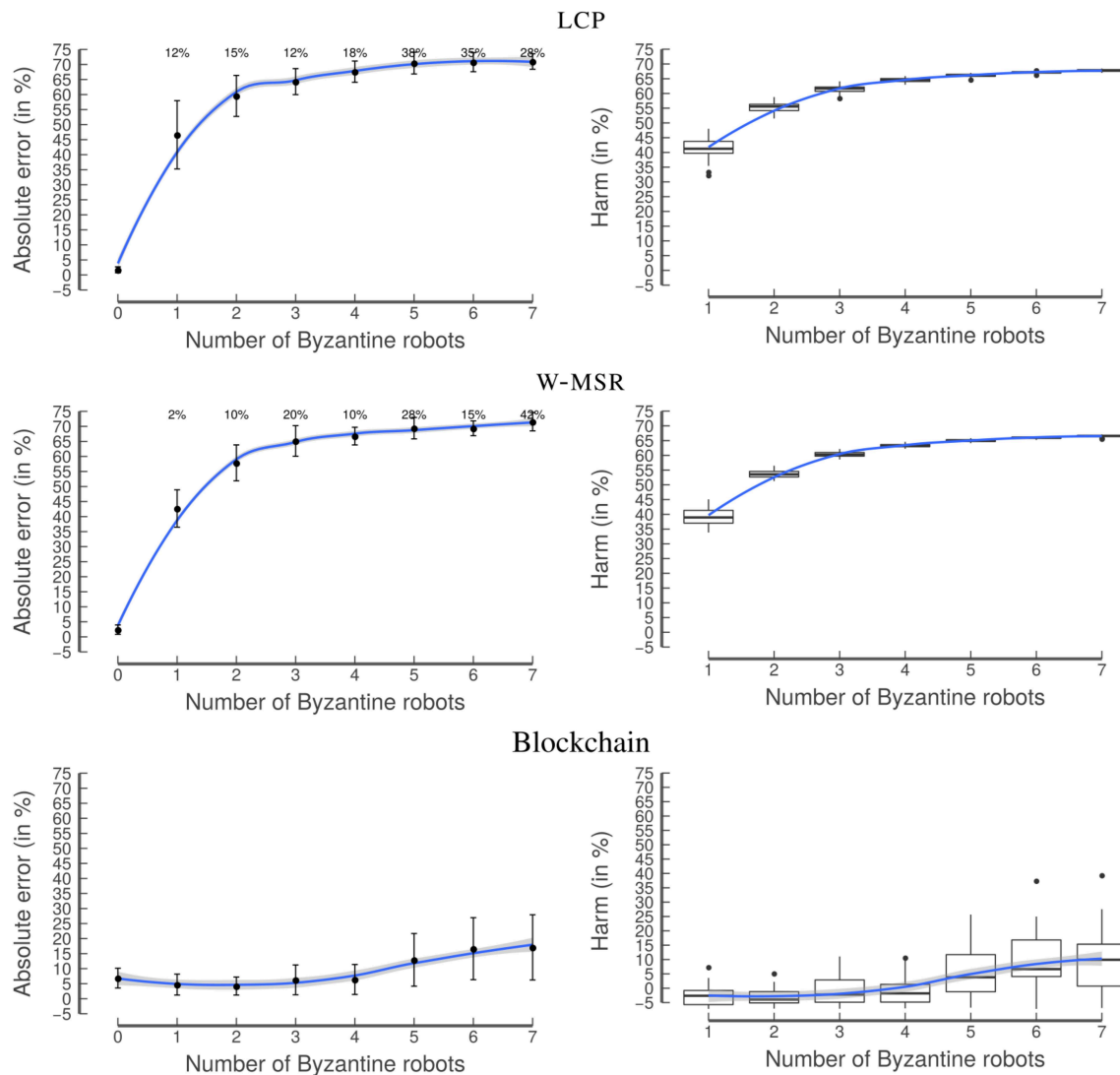


FIGURE 14 | Experiment 5: Sybil attack. In the comparison of the approaches, it is clearly visible that LCP (top) and W-MSR (middle) fail in the presence of even one Byzantine robot performing a Sybil attack. In contrast, the blockchain approach (bottom) is able to prevent these attacks by limiting the number of transactions that can be included in the blockchain. The graphs on the left-hand side show the mean with the error bars indicating the standard deviation. The blue line is obtained by locally estimated scatterplot smoothing (LOESS), the gray band around the blue line shows the 95% confidence interval for predictions from the LOESS regression.

robot swarms. A swarm is *open* when entities are free to join (e.g., because it turns out that the mission is too complex to be solved by a smaller swarm) and leave the swarm at any moment in time (e.g., because of a hardware failure). Sending a message via a blockchain is only possible when the sender spends some amount of cryptotokens. Hence, the number of messages a robot can send is limited and Sybil attacks can be prevented. This is of the utmost importance for many swarm robotics applications where a Sybil attack would undermine the swarm performance. For example, in voting scenarios without Sybil attack protection, an attacker would be able to achieve the majority; and in sensor fusion scenarios, an attacker would be able to gravely bias the swarm estimate. These attacks do not require sophisticated programming skills and are hard to prevent in decentralized

systems (Borisov, 2006). The most used means of preventing such attacks are centralized cryptographic authentication or password authentication. In our case, this would have meant that at the beginning of a simulation run, each robot would have received a list of public keys that are seen as trusted entities and would only have accepted a message from another robot if the message was signed by one of the trusted robots. However, this would entail the common disadvantages of centralized systems, such as the presence of a single point of failure at the moment when the list of public keys is created and distributed and reduced flexibility, since every robot must be identified before deployment and adding robots at run time would not be possible. Therefore, basing the approach on centralized cryptographic authentication would restrict the applicability to *closed* robot swarms.

Finally, a blockchain serves as a tamper-proof audit log and keeps track of all relevant information from all robots over time. In real-world applications it may happen that only a single robot can be retrieved (e.g., only one robot might be physically reachable, or retrieving robots might be very expensive)¹². However, the information written in its blockchain may be sufficient to reconstruct the complete course of the experiment. This information can be post-processed, e.g., outliers could even be detected after the end of the experiment. In addition, any other irregularities can be spotted and analyzed, e.g., for digital forensics.

7.2. Limitations

Our results clearly showed that the blockchain-based consensus protocol outperforms existing consensus protocols when Byzantine robots are present and that it is even needed when wishing to reach consensus in a decentralized manner under Sybil attack. While we can conclude that robot swarms are better off with blockchain technology, certain constraints need to be considered at the design stage before choosing to work with blockchain-controlled robot swarms.

A first possible issue is the fact that blockchains can introduce delays. Transactions first have to be mined to be considered by the smart contracts. Therefore, if fast reactions to messages are required, blockchains are not advisable. Instead, blockchains should be used for security-relevant data and should be combined with traditional local processing to yield hybrid approaches. Therefore, it is important to determine which information is security-relevant and should be processed via smart contracts (“on-chain”) and which information can be processed locally by single robots (“off-chain”).

Another possible issue is connected with blockchain technology’s storage requirements. This was however not the case in our experiments, where the size of an escrow blockchain transaction was 148 Bytes and the total size of the blockchain (including auxiliary files) reached on average 6.8 MB after 1,000 s. During these 1,000 s, on average, 350 transactions were stored in the blockchain. To further test scalability, we conducted experiments with a run-time of 24 h with 20 robots. After the 24 h, the total size of the blockchain reached on average 33 MB. The blockchain size grows linearly after an initialization phase of approximately 6 h during which approximately one block is created per second; in the beginning, the network needs to adapt to the hash power in the network; after 6 h, one block is created approximately every 15 s. This time interval is the default in Ethereum, and could be changed if necessary. If we hypothesize robots with 16 GB of storage capacity—this is within the capacity of state-of-the-art swarm robotic platforms, such as the Pi-puck robot (Millard et al., 2017)—the storage would last for approximately 485 days.

Another aspect of scalability is the influence of the robot swarm size on the blockchain size. Adding more robots to the swarm might increase the blockchain size because a larger swarm

might create more transactions. In the following calculation, we assume that 1,000 robots create 50 times more transactions than 20 robots and that each robot creates a transaction every 45 s. With these 1,000 robots, the upper limit for the estimated blockchain size would be 1.5 GB after 24 h. Please note that this calculation is just a rough approximation and that the study of scalability has other aspects that should be taken into account in future research, such as: (i) with a larger swarm size, it might suffice to create a transaction after longer intervals, reducing in this way the overall dimension of the blockchain; (ii) transactions could be aggregated or preprocessed before sending them to the blockchain; and (iii) since the PoW algorithm adapts to the hash power in the network, the number of mined blocks is largely independent of the number of robots, therefore, the space requirements for a greater number of robots will grow sublinearly.

In this article, we used a PoW-based consensus protocol. In contrast to popular opinion, PoW does not require sophisticated hardware and does not become necessarily harder over time. The difficulty of the mining puzzle depends on the total hash power in the network. Less powerful hardware leads to lower hash power. From a theoretical point of view, it would be possible to mine on a Kilobot (Rubenstein et al., 2014), which has an 8 MHz processor. In addition, it has been demonstrated that a variety of single board processors with ARM processors (e.g., the Raspberry Pi) are able to mine and run Ethereum nodes¹³. If, however, an intruder can outperform the hash power of the remaining robots (51% attack), it can change the order of the transactions and decide whether or not transactions should be included in the blockchain. Therefore, the higher the hash power of the network, the more difficult it is to perform a 51% attack. In this article, we used 2.0 GHz and 1.8 GB of RAM so that Ethereum works “out-of-the-box,” as explained in section 4. In order to let Ethereum run on robots with more limited hardware such as those that we have recently acquired in our lab, we have created a modified version of Ethereum’s source code¹⁴. With these modifications, Ethereum, including PoW, runs on the Pi-puck robots in our lab. By changing the initial difficulty specified in the genesis block, it is possible to establish a direct mapping for the time it takes to perform the PoW calculations from our simulations to the physical hardware.

A powerful intruder cannot forge signatures or change the logic implemented in a smart contract. Therefore, it depends on the context whether a PoW-based consensus protocol is adequate. If no powerful intruder is expected to enter the swarm (e.g., in an underwater exploration), PoW can be suitable: as long as the majority of robots acts according to the protocol,

¹³<http://ethembedded.com/>, accessed on September 17, 2019.

¹⁴The necessary modifications can be found at <http://iridia.ulb.ac.be/supp/IridiaSupp2019-009/Ethereum-on-Pi-puck/>. Ethereum’s default mining algorithm—an extension of the basic PoW algorithm, as for example used in Bitcoin—includes a memory-hard problem to make it resistant to specialized mining hardware using ASICs. This is done via the generation of a data structure called DAG which requires more than 1.0 GB of RAM (for a description of the DAG, see <https://github.com/ethereum/wiki/wiki/ethash-dag>, accessed on February 16, 2020). With the described modifications, the RAM requirements can be reduced to a few MB.

¹²If one has, however, the possibility to choose between different robots, one may select the robot with the longest blockchain to make sure that the chain is selected where the highest number of participants contributed to the Proof-of-Work.

the data in the blockchain can be trusted. In robot swarm deployments, one might be concerned that the computational overhead required by PoW might lead to battery drain. However, preliminary results (not discussed in this article) show that the power consumption due to the blockchain mining activity is low and compatible with experimentation with a swarm of Pi-puck robots.

If the swarm is operating in an environment where reliable global communication is possible, PoW does not need to be run on the hardware of the robots. In this case, a custom blockchain network maintained by the robots is not necessary. Instead, a smart contract could be used in the main Ethereum network. Since the main Ethereum network is maintained by a decentralized network of computers, it does not pose a single point of failure. However, such a scenario would change some other aspects (e.g., the entry conditions for new robots) and would possibly have a stronger focus on economics (e.g., attacks would become expensive in terms of the “main Ethereum network” cryptocurrency, that has a certain market price), so we will leave it for future work. The scope of this article is strictly limited to swarm robotics to avoid any confusion with centralized multi-robot systems.

7.3. Future Work

In this article, we studied attacks at the collective estimation level by sending deceitful data. However, there is a difference between attacks at the collective estimation level and attacks at the blockchain level. Some attacks that can pose problems to decentralized systems, such as replay attacks, are naturally prevented by blockchain technology. Yet, there are potential blockchain-level attacks in robot swarms: for example, clustering of malicious robots to perform a majority attack. In order to prevent majority attacks, a flocking algorithm may guarantee a certain degree of connectivity and help to avoid local robot clusters that have different blockchain forks. As an additional procedure to manage blockchain forks, the number of confirmations (i.e., the number of blocks after the block number that contains a certain transaction) can serve as a metric indicating how probable it is that a transaction stays in a specific block.

The robustness of the blockchain approach to much sparser connectivity is an open research topic that we plan to address in future research. As described in section 2.1, transactions stay valid and can be included in the blockchain after days of disconnectivity or after a blockchain fork gets discarded (they then become unconfirmed transactions again that can be included in later blocks). However, the longer the robot clusters stay disconnected, the higher the risk that they base a decision on a blockchain fork that is not the longest blockchain. There are several strategies to address this issue. One option is to increase the average time between mined blocks (block time) via a different difficulty setting. This will introduce delays but reduce the risk that decisions are based on non-final information. Further possibilities are aggregation algorithms to guarantee a certain connectivity; or “messenger robots” that can move faster (e.g., UAVs) and bring together different blockchain information. The robot that we are currently planning to use (the Pi-puck)

has a Wi-Fi speed of up to 72 Mbps. Therefore, if a robot in the studied scenario would join the swarm after 20 min, it could download the blockchain within a few seconds from other robots. In future research, we will measure the relationship between the time two components of the swarm were disconnected and the time it takes to re-synchronize the blockchain across the disconnected robots afterwards.

In future work, we plan to transfer the system to heterogeneous robot swarms where some of the robots might have very different computational capabilities. In such a heterogeneous robot swarm, the overhead of blockchain technology could be delegated to the more powerful robots. For example, a swarm of smaller Kilobots could report back to larger Pi-puck robots at certain intervals. The Pi-puck robots could store the blockchain and perform the PoW, while the Kilobots just create transactions.

Another option to bring blockchain technology to robots of any size is to use a different blockchain framework. In the last couple of years, blockchain technology has experienced dramatic development. While at the start of this research work Ethereum was the only fully-developed blockchain-based smart contract platform, there are now more than a dozen smart contract platforms. These frameworks differ, among other aspects, in terms of their computational requirements, consensus protocol, scalability, robustness, speed, and use cases. The nature of, for example, public-key cryptography, transactions, and smart contracts, is largely independent of the used consensus protocol. Therefore, our work can serve as a basis for studying other blockchain frameworks, such as, Hyperledger Sawtooth¹⁵, Cardano¹⁶, and Tezos¹⁷ in the context of robot swarms. By means of these blockchain frameworks, we intend to compare alternatives to the *Proof-of-Work*-based consensus protocol on both the physical robots and via the ARGoS-Blockchain interface in future work. We plan to study *Proof-of-Stake* (already implemented in some existing blockchain protocols), *Proof-of-Sensing* (only robots that can produce a certain sensory output can send or validate transactions), or even *Proof-of-physical-Work* (only robots that can prove that they have performed physical work, such as collecting an item can send or validate transactions).

8. CONCLUSIONS

In this article, our goal was to compare consensus protocols used in swarm robotics with regard to their resilience to Byzantine robots. We showed that existing consensus protocols can easily fail in the presence of Byzantine robots. With the developed ARGoS-blockchain interface, we provide a framework for secure robot swarm coordination via blockchain-based smart contracts as “meta-controllers.” Blockchain technology makes sure that every robot runs the same code, that the code is executed exactly as specified, that the robots come to a consensus regarding the outcome of the execution, and that

¹⁵<https://sawtooth.hyperledger.org/> (accessed September 12, 2019).

¹⁶<https://www.cardano.org/en/home/> (accessed on September 12, 2019).

¹⁷<https://tezos.com/> (accessed September 12, 2019).

there is not a single point of failure. Blockchains prevent Sybil attacks via their scarce cryptocurrency that limits the number of transactions a robot can send. Additionally, the blockchain is able to securely store critical events. This decentralized log can then be used to evaluate the quality of experiments and to spot irregularities.

Blockchain-controlled robot swarms must meet certain computational and memory requirements. Compared to Internet-based blockchain networks, in robot swarms, the computational capacities are limited, the delays can be much longer, and failing entities are more probable due to rough environmental conditions or flat batteries. While we discussed these characteristics, we do not question the fact that there are still many open challenges for blockchain-based swarm robotics. Nevertheless, we are convinced that the synthesis of these two technologies offers unprecedented possibilities and that the various challenges can gradually be addressed. In this article we have shown that blockchain-based smart contracts are a promising and versatile tool to address security issues in swarm robotics. If we ever want robot swarms to be deployed in the real world, we need to start preparing them to the possible presence of Byzantine robots: the work we have presented is a first step in this direction.

REFERENCES

- Beal, J. (2016). Trading accuracy for speed in approximate consensus. *Knowl. Eng. Rev.* 31, 325–342. doi: 10.1017/S0269888916000175
- Borisov, N. (2006). “Computational puzzles as Sybil defenses,” in *Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing (P2P '06)* (Los Alamitos, CA: IEEE Press), 171–176.
- Brown, A., Franken, P., Bonner, S., Dolezal, N., and Moross, J. (2016). Safecast: successful citizen-science for radiation measurement and communication after Fukushima. *J. Radiol. Protect.* 36, S82–S101. doi: 10.1088/0952-4746/36/2/S82
- Buterin, V. (2014). *A Next-Generation Smart Contract and Decentralized Application Platform. Ethereum Project White Paper*. Technical Report. Available online at: <https://github.com/ethereum/wiki/wiki/White-Paper> (Accessed July 18, 2019).
- Castelló Ferrer, E. (2016). The blockchain: a new framework for robotic swarm systems. *arXiv:1608.00695v3*. doi: 10.1007/978-3-030-02683-7_77
- Christensen, A. L., O’Grady, R., Birattari, M., and Dorigo, M. (2008). Fault detection in autonomous robots based on fault injection and learning. *Auton. Robots* 24, 49–67. doi: 10.1007/s10514-007-9060-9
- Christensen, A. L., O’Grady, R., and Dorigo, M. (2009). From fireflies to fault-tolerant swarms of robots. *IEEE Trans. Evol. Comput.* 13, 754–766. doi: 10.1109/TEVC.2009.2017516
- Crosby, M., Pattanayak, P., Verma, S., and Kalyanaraman, V. (2016). Blockchain technology: beyond bitcoin. *Appl. Innov.* 2:71. doi: 10.1109/ICCECOME.2018.8658518
- Dorigo, M., Trianni, V., Şahin, E., Groß, R., Labella, T. H., Baldassarre, G., et al. (2004). Evolving self-organizing behaviors for a swarm-bot. *Auton. Robots* 17, 223–245. doi: 10.1023/B:AURO.0000033973.24945.f3
- Douceur, J. R. (2002). “The Sybil attack,” in *1st International Workshop on Peer-to-Peer systems, Vol. 2429 of Lecture Notes in Computer Science*, eds P. Druschel, F. Kaashoek, and A. Rowstron (Berlin; Heidelberg: Springer), 251–260.
- Elhage, N., and Beal, J. (2010). “Laplacian-based consensus on spatial computers,” in *Proceedings of the 9th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2010)* (Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems), 907–914.
- Fernandes, M., and Alexandre, L. A. (2019). Robotchain: using Tezos technology for robot event management. *Ledger* 4(Suppl. 1). doi: 10.5195/ledger.2019.175

DATA AVAILABILITY STATEMENT

The datasets generated for this study can be found in the IRIDIA—Supplementary Information (ISSN: 2684-2041) at <http://iridia.ulb.ac.be/supp/IridiaSupp2019-009/>.

AUTHOR CONTRIBUTIONS

All authors contributed to the conceptualization of this research and the setup of the experiments. VS implemented the software and conducted and analyzed the experiments. In addition, he wrote the first draft of this manuscript. EC and MD gave critical feedback, revised the article, and contributed to the final manuscript.

FUNDING

VS and MD acknowledge support from the Belgian F.R.S.-FNRS and from the FLAG-ERA project RoboCom++. VS additionally acknowledges support from the Office of Naval Research Global under the Visiting Scientists Program No. 18-8-003. EC acknowledges support from the Marie Skłodowska-Curie actions (EU project BROS—DLV-751615).

- Ferrante, E., Turgut, A. E., Huepe, C., Stranieri, A., Pincioli, C., and Dorigo, M. (2012). Self-organized flocking with a mobile robot swarm: a novel motion control method. *Adapt. Behav.* 20, 460–477. doi: 10.1177/1059712312462248
- Gil, S., Kumar, S., Mazumder, M., Katabi, D., and Rus, D. (2017). Guaranteeing spoof-resilient multi-robot networks. *Auton. Robots* 41, 1383–1400. doi: 10.1007/s10514-017-9621-5
- Guerrero-Bonilla, L., Prorok, A., and Kumar, V. (2017). Formations for resilient robot teams. *IEEE Robot. Autom. Lett.* 2, 841–848. doi: 10.1109/LRA.2017.2654550
- Gupta, V., Langbort, C., and Murray, R. M. (2006). “On the robustness of distributed algorithms,” in *Proceedings of the 45th IEEE Conference on Decision and Control* (Piscataway, NJ: IEEE Press), 3473–3478.
- Higgins, F., Tomlinson, A., and Martin, K. M. (2009). “Survey on security challenges for swarm robotics,” in *Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems* (Piscataway, NJ: IEEE Press), 307–312.
- Jacoby, W. G. (2000). Loess: a nonparametric, graphical tool for depicting relationships between variables. *Elect. Stud.* 19, 577–613. doi: 10.1016/S0261-3794(99)00028-1
- Lamport, L., Shostak, R., and Pease, M. (1982). The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.* 4, 382–401.
- LeBlanc, H. J., Zhang, H., Koutsoukos, X., and Sundaram, S. (2013). Resilient asymptotic consensus in robust networks. *IEEE J. Select. Areas Commun.* 31, 766–781. doi: 10.1109/JSAC.2013.130413
- Lopes, V., and Alexandre, L. A. (2019). “Detecting robotic anomalies using robotchain,” in *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC 2019)* (Piscataway, NJ: IEEE Press), 1–6.
- McAbee, A., Tummala, M., and McEachen, J. (2019). “Military intelligence applications for blockchain technology,” in *Proceedings of the 52nd Hawaii International Conference on System Sciences* (Honolulu, HI: ScholarSpace), 6031–6040.
- Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux J.* 2014. Available online at: <https://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment>

- Millard, A. G., Joyce, R., Hilder, J. A., Fleşeriu, C., Newbrook, L., Li, W., et al. (2017). "The pi-puck extension board: a Raspberry Pi interface for the e-puck robot platform," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Los Alamitos, CA: IEEE Press), 741–748.
- Millard, A. G., Timmis, J., and Winfield, A. F. T. (2014). "Towards exogenous fault detection in swarm robotic systems," in *Towards Autonomous Robotic Systems - Proceedings of TAROS 2013 - 14th Annual Conference, Vol. 8069 of Lecture Notes in Computer Science* (Cham: Springer), 429–430.
- Montes de Oca, M. A., Ferrante, E., Scheidler, A., Pinciroli, C., Birattari, M., and Dorigo, M. (2011). Majority-rule opinion dynamics with differential latency: a mechanism for self-organized collective decision-making. *Swarm Intell.* 5, 305–327. doi: 10.1007/s11721-011-0062-z
- Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Technical Report. Available online at: <https://bitcoin.org/bitcoin.pdf> (Accessed August 11, 2018).
- Nishida, Y., Kaneko, K., Sharma, S., and Sakurai, K. (2018). "Suppressing chain size of blockchain-based information sharing for swarm robotic systems," in *Proceedings of the Sixth International Symposium on Computing and Networking Workshops (CANDARW 2018)* (Los Alamitos, CA: IEEE Press), 524–528.
- Olfati-Saber, R., and Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* 49, 1520–1533.
- Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., et al. (2012). ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell.* 6, 271–295. doi: 10.1007/s11721-012-0072-5
- Primiero, G., Tuci, E., Tagliabue, J., and Ferrante, E. (2018). "Swarm attack: a self-organized model to recover from malicious communication manipulation in a swarm of simple simulated agents," in *Swarm Intelligence - Proceedings of ANTS 2018 - Eleventh International Conference*, eds M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. Reina, and V. Trianni (Cham: Springer), 213–224.
- Reina, A., Dorigo, M., and Trianni, V. (2014). "Collective decision making in distributed systems inspired by honeybees behaviour," in *Proceedings of the 13th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2014)*, eds A. Lomuscio, P. Scerri, A. Bazzan, and M. Huhns (Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems), 1421–1422.
- Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M., and Trianni, V. (2015). A design pattern for decentralised decision making. *PLoS ONE* 10:e0140950. doi: 10.1371/journal.pone.0140950
- Roumeliotis, S. I., Sukhatme, G. S., and Bekey, G. A. (1998). "Sensor fault detection and identification in a mobile robot," in *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*, Vol. 3 (New York, NY: IEEE Press), 1383–1388.
- Rubenstein, M., Ahler, C., Hoff, N., Cabrera, A., and Nagpal, R. (2014). Kilobot: a low cost robot with scalable operations designed for collective behaviors. *Robot. Auton. Syst.* 62, 966–975. doi: 10.1016/j.robot.2013.08.006
- Saldaña, D., Prorok, A., Sundaram, S., Campos, M. F. M., and Kumar, V. (2017). "Resilient consensus for time-varying networks of dynamic agents," in *Proceedings of the American Control Conference (ACC)* (Piscataway, NJ: IEEE Press), 252–258.
- Sargeant, I., and Tomlinson, A. (2016). "Maliciously manipulating a robotic swarm," in *Proceedings of ESCS'16 - The 14th International Conference on Embedded Systems, Cyber-Physical Systems, & Applications* (Bogart, GA: CSREA Press), 122–128.
- Saulnier, K., Saldaña, D., Prorok, A., Pappas, G. J., and Kumar, V. (2017). Resilient flocking for mobile robot teams. *IEEE Robot. Autom. Lett.* 2, 1039–1046. doi: 10.1109/LRA.2017.2655142
- Schmickl, T., Thenius, R., Moeslinger, C., Radspieler, G., Kernbach, S., Szymanski, M., et al. (2009). Get in touch: cooperative decision making based on robot-to-robot collisions. *Auton. Agents Multi-Agent Syst.* 18, 133–155. doi: 10.1007/s10458-008-9058-5
- Soysal, O., and Sahin, E. (2005). "Probabilistic aggregation strategies in swarm robotic systems," in *Proceedings of the 2005 IEEE Swarm Intelligence Symposium (SIS 2005)* (Piscataway, NJ: IEEE Press), 325–332.
- Strobel, V., Castelló Ferrer, E., and Dorigo, M. (2018). "Managing Byzantine robots via blockchain technology in a swarm robotics collective decision making scenario," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018)*, eds M. Dastani, G. Sukthankar, E. André, and S. Koenig (Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems), 541–549.
- Strobel, V., and Dorigo, M. (2018). "Blockchain technology for robot swarms: a shared knowledge and reputation management system for collective estimation," in *Swarm Intelligence - Proceedings of ANTS 2018 - Eleventh International Conference*, Vol. 11172 of *Lecture Notes in Computer Science*, eds M. Dorigo, M. Birattari, C. Blum, A. L. Christensen, A. Reina, and V. Trianni (Cham: Springer), 425–426.
- Szabo, N. (1997). Formalizing and securing relationships on public networks. *First Monday* 2. doi: 10.5210/fm.v2i9.548
- Tarapore, D., Christensen, A. L., and Timmis, J. (2017). Generic, scalable and decentralized fault detection for robot swarms. *PLoS ONE* 12:e182058. doi: 10.1371/journal.pone.0182058
- Tarapore, D., Lima, P. U., Carneiro, J., and Christensen, A. L. (2015). To err is robotic, to tolerate immunological: fault detection in multirobot systems. *Bioinspir. Biomim.* 10:016014. doi: 10.1088/1748-3190/10/1/016014
- Tarapore, D., Timmis, J., and Christensen, A. (2019). Fault detection in a swarm of physical robots based on behavioral outlier detection. *IEEE Trans. Robot.* 35:1–7. doi: 10.1109/TRO.2019.2929015
- Valentini, G., Brambilla, D., Hamann, H., and Dorigo, M. (2016a). "Collective perception of environmental features in a robot swarm," in *Swarm Intelligence - Proceedings of ANTS 2016 - Tenth International Conference*, Vol. 9882 of *Lecture Notes in Computer Science* (Cham: Springer), 65–76.
- Valentini, G., Ferrante, E., and Dorigo, M. (2017). The best-of-n problem in robot swarms: formalization, state of the art, and novel perspectives. *Front. Robot. AI* 4:9. doi: 10.3389/frobt.2017.00009
- Valentini, G., Ferrante, E., Hamann, H., and Dorigo, M. (2016b). Collective decision with 100 Kilobots: speed versus accuracy in binary discrimination problems. *Auton. Agents Multi-Agent Syst.* 30, 553–580. doi: 10.1007/s10458-015-9323-3
- Winfield, A. F. T., and Nembrini, J. (2006). Safety in numbers: fault tolerance in robot swarms. *Int. J. Modell. Identif. Control* 1, 30–37. doi: 10.1504/IJMIC.2006.008645
- Xiao, L., Boyd, S., and Lall, S. (2005). "A scheme for robust distributed sensor fusion based on average consensus," in *The Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)* (Piscataway, NJ: IEEE Press), 63–70.
- Zikratov, I., Maslennikov, O., Lebedev, I., Ometov, A., and Andreev, S. (2016). "Dynamic trust management framework for robotic multi-agent systems," in *Proceedings of the 12th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN 2016)*, and the 5th Conference on Internet of Things and Smart Spaces (ruSMART 2016), eds O. Galinina, S. Balandin, and Y. Koucheryavy (Cham: Springer), 339–348.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling editor declared a past co-authorship with one of the authors MD.

Copyright © 2020 Strobel, Castelló Ferrer and Dorigo. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Sparse Robot Swarms: Moving Swarms to Real-World Applications

Danesh Tarapore^{1*}, Roderich Groß² and Klaus-Peter Zauner¹

¹ School of Electronics and Computer Science, University of Southampton, Southampton, United Kingdom, ² Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, United Kingdom

Robot swarms are groups of robots that each act autonomously based on only local perception and coordination with neighboring robots. While current swarm implementations can be large in size (e.g., 1,000 robots), they are typically constrained to working in highly controlled indoor environments. Moreover, a common property of swarms is the underlying assumption that the robots act in close proximity of each other (e.g., 10 body lengths apart), and typically employ uninterrupted, situated, close-range communication for coordination. Many real world applications, including environmental monitoring and precision agriculture, however, require scalable groups of robots to act jointly over large distances (e.g., 1,000 body lengths), rendering the use of *dense* swarms impractical. Using a dense swarm for such applications would be invasive to the environment and unrealistic in terms of mission deployment, maintenance and post-mission recovery. To address this problem, we propose the *sparse* swarm concept, and illustrate its use in the context of four application scenarios. For one scenario, which requires a group of rovers to traverse, and monitor, a forest environment, we identify the challenges involved at all levels in developing a sparse swarm—from the hardware platform to communication-constrained coordination algorithms—and discuss potential solutions. We outline open questions of theoretical and practical nature, which we hope will bring the concept of sparse swarms to fruition.

Keywords: swarm robotics, multirobot systems, field robotics, forest robots, sparse coupling, communication networks, information propagation, long-range radio

OPEN ACCESS

Edited by:

Eliseo Ferrante,
Vrije Universiteit Amsterdam,
Netherlands

Reviewed by:

Dario Albani,
Italian National Research Council, Italy
Giulia De Masi,
Zayed University,
United Arab Emirates

*Correspondence:

Danesh Tarapore
d.s.tarapore@soton.ac.uk

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 20 December 2019

Accepted: 19 May 2020

Published: 02 July 2020

Citation:

Tarapore D, Groß R and Zauner K-P
(2020) Sparse Robot Swarms: Moving
Swarms to Real-World Applications.
Front. Robot. AI 7:83.
doi: 10.3389/frobt.2020.00083

1. INTRODUCTION

Swarm robotics takes inspiration from observed behaviors of collective systems in nature (Camazine et al., 2003) to develop large-scale teams of robots with limited individual capabilities; the collective behavior emerging from the self-organized interactions between the many robots of a swarm allow it to solve complex tasks (Beni, 2004; Sahin, 2004). To date, robot swarms have been demonstrated to solve tasks such as aggregation (Gauci et al., 2014), coordinated movement (Virágh et al., 2014), transportation of objects (Wang and Schwager, 2016), self-assembly (Rubenstein et al., 2014; Mathews et al., 2017), collective construction of structures (Werfel et al., 2014), and decentralized consensus formation (Schmickl and Crailsheim, 2008; Valentini et al., 2016).

Despite the variety of movement-centric and simple cognitive tasks that robot swarms have been demonstrated to perform (Bayındır, 2016), they continue to function largely as demonstration platforms in carefully controlled laboratory environments (Schranz et al., 2020), unable to transition to realistic application scenarios due to the following challenges:

Difficulties maintaining a high-density swarm: A common feature of existing swarm robotic systems is the underlying assumption that the robots of the swarm act in close proximity of each other. Inter-robot distances of existing swarms are typically around 1–10 body lengths, both in indoor (Rubenstein et al., 2014; Pickem et al., 2017), and outdoor (Duarte et al., 2016; Zoss et al., 2018) environments. *Densely packed robot swarms* are inspired by social insect colonies, and rely on inter-robot physical interactions to complete their task. However, in employing such swarms in real-world outdoor applications encompassing large areas, the end-user faces a number of challenges involving the deployment and maintenance of such large numbers of robots during the mission. The recovery of the swarm post-mission is also problematic, particularly considering the high environmental cost of unrecovered robots. Furthermore, densely packed swarms are more likely to physically disrupt the other mission-participants, such as emergency workers in search and rescue operations.

Constraints on inter-robot communication: In most robot swarms, inter-robot coordination is reliant on an uninterrupted access to situated, close-range communication of coordination messages between robots (Duarte et al., 2016; Mathews et al., 2017, 2019; Garattoni and Birattari, 2018; Albani et al., 2019). However, in real-world scenarios robot swarms may face a number of challenges in exchanging coordination messages across the swarm. Swarms would be required to share communication channels with other participants in a mission, with high-bandwidth wireless channels most likely being reserved for human operators. Additionally, due to regulatory imposed channel-specific limitations on the communication duty-cycle (i.e., the proportion of time the transmitter is sending messages) (Semtech, 2015; Bor and Roedig, 2017), the robots of the swarm may also expect significant latency in receiving coordination messages.

Restricted mobility and low endurance of robot platforms: Most commercially available swarm robot platforms are designed to be operated over short distances (i.e., limited endurance) in carefully controlled indoor laboratory environments. This is particularly the case for swarms of ground robots that are typically constrained to operate on smooth, leveled surfaces such as table-tops (Mondada et al., 2009; Chamanbaz et al., 2017; Jones et al., 2018). Furthermore, low-cost outdoor platforms typically offer low autonomy and endurance, and are not thoroughly tested, compared to more costly alternatives.

In summary, despite the desirable characteristics of robustness and flexibility observed in collective systems in nature (Camazine et al., 2003), robot swarms inspired by such systems remain ill suited for realistic application scenarios. In mimicking the densities and coordination strategies of swarms in nature, swarm robotics faces a number of technological challenges relating to materials and their fabrication, power-efficiency, and battery-technologies for developing small-scale robots of a swarm that are compliant and autonomous in manners similar to their biological counterparts (Yang et al., 2018). Therefore, for robot swarms to be employed in realistic application scenarios, swarm technologies need to be reconceptualized.

In this paper, we propose the concept of *sparse swarms*, where the group of robots interact while (i) not being in close proximity to each other, and/or (ii) it is not possible for information to rapidly propagate within the group. Sparse swarms could be particularly relevant in application scenarios, where the robots are operating in the order of 1,000 body lengths apart under sporadic low-bandwidth communication constraints. In such scenarios, the robots would be likely be required to coordinate their activities via informational interactions rather than physical interactions.

2. RELATED CONCEPTS

This section notes similarities and differences between sparse swarms and two related concepts, cloud robotics and multirobot systems.

2.1. Cloud Robotics

In the domain of cloud robotics, robots separated by large distances perform some tasks, for example, grasping objects, while storing and sharing task-critical information over a “cloud” (Beetz et al., 2011; Kehoe et al., 2015; Wan et al., 2016). This is realized via machine-to-cloud (M2C) and/or machine-to-machine (M2M) communications (Hu et al., 2012). In both cloud robotics and sparse swarms, robots may rely on long-range interactions, for example, to share and learn from each others’ experiences. However, while cloud-linked robots work on independent tasks in different environments, sparse swarm robots work on a common task in a shared environment, which requires them to coordinate their activities. Moreover, cloud-linked robots rely on costly external infrastructure—Internet connections providing high-bandwidth, low-latency communication with cloud services—which may not be available to sparse swarms deployed in real-world scenarios, for example, outdoors. The robots in a typical sparse swarm scenario are also likely to be less expensive than those in a typical cloud robotics scenario.

2.2. Multirobot Systems

While any robot swarm can be considered a multirobot system, the former term is usually preferred where a system comprises a relatively homogeneous group of robots, typically a dozen or more, which are unable to solve a given task efficiently on their own, but coordinate their activities, by exploiting only information that they can locally obtain, in the absence of global infrastructure (Sahin, 2004). With sparse swarms we consider groups of robots that are more sparsely distributed than present robot swarms, and even most multirobot systems (Chamanbaz et al., 2017). The high cost of currently available outdoor multirobot platforms prevents their adoption in robot swarms¹. Moreover, many implementations of outdoor multirobot systems lack a fully decentralized, fault-tolerant control architecture, with the robots receiving instructions from

¹Note that we are not postulating to reduce the number of robots in swarms—merely their density.

a central planning/coordination node (Tardioli et al., 2016; Weinstein et al., 2018).

Some studies have focused on multirobot systems operating in communication-constrained environments (Amigoni et al., 2017; Tardioli et al., 2019). One approach is using some robots to physically deliver information to within communication range of other robots (Ducatelle et al., 2014; Cesare et al., 2015). Another approach is using some robots to form multi-hop communication chains, allowing for rapid propagation of information beyond the communication range of individual robots (Nouyan et al., 2009; Tardioli et al., 2010; Pei et al., 2013; Luo et al., 2019). Yet another approach is for the robots to reestablish contact, for example, periodically, at a priori known locations (Hollinger and Singh, 2012; Kantaros and Zavlanos, 2017) or using search (Banfi et al., 2018; Vandermeulen et al., 2018). Some of these approaches rely on a priori knowledge regarding how well robots can communicate between any two points in the environment (Amigoni et al., 2017; Banfi et al., 2018; Vandermeulen et al., 2018), which makes their application in real-world scenarios challenging.

3. CONCEPTUALIZING A SPARSE SWARM

In the following, we describe two alternative characterizations of the sparse swarm concept. In both cases, we consider a swarm of n robots, $S = \{1, 2, \dots, n\}$.

3.1. Constraints on Inter-robot Proximity

In a sparse swarm, it would be costly for the robots to get into close proximity of each other (e.g., 10 body lengths away). To formalize this idea, we examine the swarm from a given time step, $k_0 \geq 0$, during the mission, for example, its start, $k_0 = 0$. We refer to the swarm as *sparse* at time step k_0 if the following condition is satisfied by a typical robot, $i \in S$:

$$\begin{aligned} \text{cost}_i(\text{"move to nearest neighbor"}, k_0) &>> \\ \text{cost}_i(\text{"perform typical operation"}, k_0), \end{aligned} \quad (1)$$

where $>>$ is defined as "at least one order of magnitude greater than," and cost_i is a function that defines the *cost* for robot i to perform a given task at a given time. The cost could reflect the time taken, or energy expended, to complete the task. It would depend on the robot's capabilities and the environment the swarm resides in. What constitutes a "typical" operation would depend on the application scenario. For example, task "perform typical operation" could involve collecting a physical sample, or moving to the next waypoint. Task "move to nearest neighbor" could involve moving directly to the robot's nearest neighbor, or moving along a path of minimal cost.

Equation (1) suggests that for the typical robot in a sparse swarm, it may be prohibitively expensive to get into close proximity of another robot. The definition is sufficiently flexible to allow for occasional close encounters among some members of the swarm.

3.2. Constraints on Inter-robot Coupling

In a sparse swarm, it would not be possible for information to propagate rapidly to all of its members. To formalize this idea, we examine the swarm from a given time step, $k_0 \geq 0$, during the mission. Let $\mathbf{x}_i[k]$ denote the state of robot $i \in S$ at time step k . A robot's state could reflect its external configuration (e.g., pose) as well as its internal configuration (e.g., behavioral state, battery level). Let $\mathbf{z}_i[k]$ denote the measurements that robot $i \in S$ obtains at time step k . Let $\tilde{\mathbf{z}}_i^{(j)}[k]$ denote the corresponding measurements that robot i would obtain had robot j not been present in the environment at time step k , and had all modifications that robot j made to the environment on or after time step k_0 been discarded. By default, we assume that a robot's state transition function is affected by noise. Let $P(\mathbf{x}_i[k])$ denote the state distribution of robot i at time step $k \geq k_0$. For $k > k_0$, let $A[k]$ be the $n \times n$ matrix with

$$A_{ij}[k] = \begin{cases} 1, & P(\mathbf{x}_i[k] | \mathbf{x}_i[k-1], \mathbf{x}_j[k-1], \mathbf{z}_i[k-1], \mathbf{z}_j[k-1]) \\ & \neq P(\mathbf{x}_i[k] | \mathbf{x}_i[k-1], \tilde{\mathbf{z}}_i^{(j)}[k-1]); \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Term $P(\mathbf{x}_i[k] | \mathbf{x}_i[k-1], \mathbf{x}_j[k-1], \mathbf{z}_i[k-1], \mathbf{z}_j[k-1])$ represents the conditional probability distribution of the state of robot i at time step k when the states and measurements of robots i and j are known at time step $k-1$. It may depend on additional information, such as the environment, which is not explicitly represented here. Term $P(\mathbf{x}_i[k] | \mathbf{x}_i[k-1], \tilde{\mathbf{z}}_i^{(j)}[k-1])$ represents the corresponding distribution under the assumption that robot j and all of its modifications made to the environment since time step k_0 are currently discarded. If such "removal" of robot j would influence the conditional state distribution of robot i at time step k , the corresponding element of the matrix, $A_{ij}[k]$, is 1, otherwise 0. Matrix A hence describes the possible interactions between all pairs.² The couplings are directional. In other words, $A_{ij}[k] = 1$ does not imply $A_{ji}[k] = 1$. We assume that $A_{ii} = 1$ for all i , as robot i , once removed, would no longer have a well-defined state.

For $\tau \in \{1, 2, \dots\}$, let

$$D(\tau) = \prod_{k=k_0+1}^{k_0+\tau} A[k]. \quad (3)$$

In other words, matrix $D(\tau)$ is a product of matrices, which models the dependencies between pairs of robots within time period τ , starting from k_0 . Intuitively we consider all robots to be fully independent at time k_0 , that is, we discard the whole history of interactions up to time k_0 . Note that if a robot i influenced robot j , and robot j influenced robot l thereafter, then robot i influenced robot l as well.

²Note that if the states of two robots are correlated this would not necessarily imply that an interaction took place. It could be that both robots independently discovered a same environmental feature.

Let

$$\tau_{min} = \operatorname{argmin}_{\tau} (D(\tau) \text{ is not sparse}), \quad (4)$$

where a matrix is considered sparse if half or more of its elements are zero. In other words, τ_{min} reflects the time it takes for information to propagate within the swarm. In particular, it denotes the earliest time after which robot i could have influenced robot j for the majority of pairs, $(i, j) \in S \times S$. In the following, we assume that τ_{min} is finite. If $\tau_{min} = \infty$, we would not refer to S as a swarm.

We refer to the swarm as *sparse* at time step k_0 if

$$\tau_{min} = \Omega(n), \quad (5)$$

that is, if τ_{min} is “at least as large as a constant times $[n]$ for all large n ” (Knuth, 1976). In other words, the time it takes for information to propagate grows at least linearly with the number of robots in the swarm.

A broad range of interactions can be captured using Equation (3). If the state of a robot described its position, an interaction could involve one robot pushing another robot, whether deliberately, or not. An interaction could involve one robot approaching a second robot, unless the presence of the second robot did not inform the choice of motion of the first. An interaction could involve a robot changing its state due to receiving a message by another robot. An interaction could involve a robot changing its state due to encountering a modification to the environment that was made by another robot. This latter form of interaction is commonly referred to as stigmergic communication.

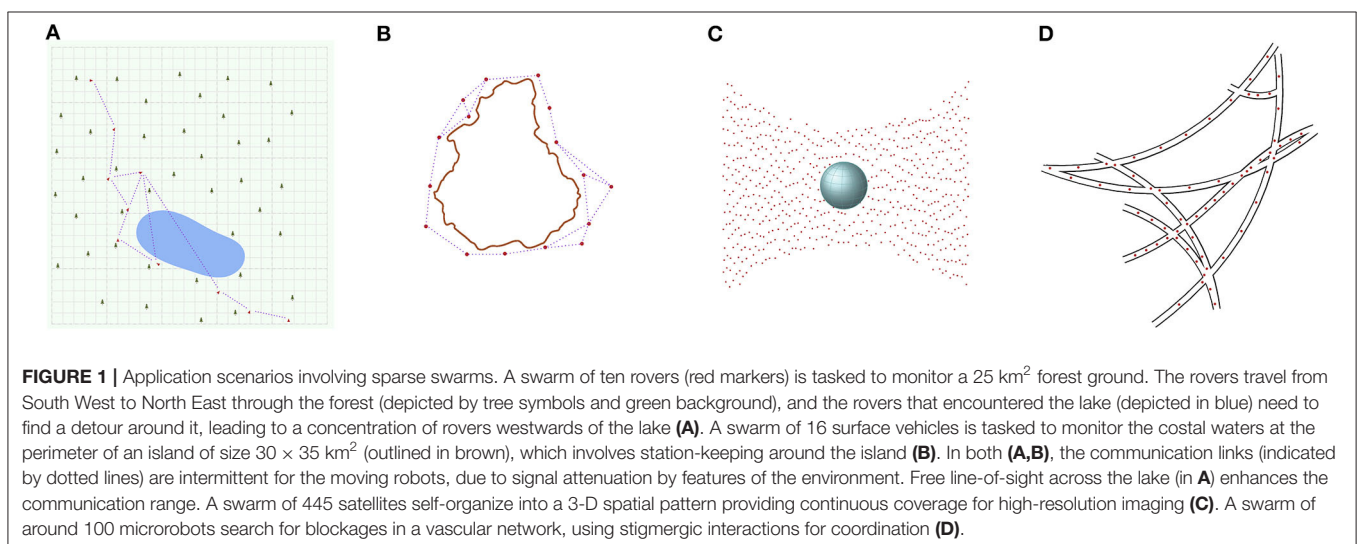
The above two criteria are meant to complement each other. Where a swarm system is investigated in a concrete situation, the *constraints on inter-robot proximity* criterion can be used, taking into account the costs for a typical operation and that to reach the nearest neighbor. Where a swarm system is investigated over an infinite set of situations, involving groups of arbitrary size, the *constraints on inter-robot coupling* criterion can be used. This

allows to evaluate the $\Omega(n)$ expression, which cannot be evaluated for swarms of constant size.

Figure 1 illustrates the sparse swarm concept in four concrete situations, reflecting a range of application scenarios. In the first scenario, a group of 10 ground rovers operate in a squared forest region of side length 5 km. A typical operation for a ground rover may be to extract and store a sample of soil, which may take 30 s of time. The (median) distance to its nearest neighbor is 261 m. Assuming a terrain that allows the robot to move with an average speed of 0.2 m/s, the (median) time to move to the nearest neighbor would be 1,305 s. In the second scenario, a group of 16 unmanned surface vessels monitor the perimeter of an island of size 35 km North-to-South and 30 km East-to-West. A typical operation for a surface vehicle may be to maintain its position along the perimeter, which would require significantly less energy than that required for the vehicle to sail a (median) distance of 6.8 km to its nearest neighbor. Such station-keeping mission scenario for a swarm of surface vehicles may also be extrapolated to 3-D for underwater, aerial and space environments, requiring larger sized swarms that are still sparse. Given the constraints on inter-robot proximity, above groups could be considered sparse swarms. Moreover, they are characterized by predominantly linear inter-robot communication networks. As the swarms would have to encompass larger environments, they would have to be proportionally larger in size, and the time it takes for information to propagate within the swarm would increase linearly with the number of robots. This would thus satisfy our constraint for inter-robot coupling for sparse swarms. An interesting scenario are in-body applications where using a dense swarm of robots may be too invasive. Instead, a sparse swarm of microrobots could be used, for example, to explore the vascular network for blockages. In such applications, the microrobots may coordinate their response using stigmergic interactions.

4. FOREST APPLICATION SCENARIO

In this section, we discuss the aforementioned forest application scenario, including the associated challenges in realizing the



concept of a sparse swarm. Consider a sparse swarm of 10 terrestrial robots (i.e., rovers) tasked with monitoring a large tract of 25 km² forest ground (**Figure 1A**). The robots are deployed at one end of the forest equidistant to each other, and are tasked with sweeping through the forest in a quasi line-formation; quasi as the robots are traversing on uneven terrain and are consequently unable to maintain a constant velocity across the swarm. The proposed scenario allows us to assess the following challenges: (i) the mechanical design of platform hardware in terms of its capability to efficiently traverse difficult terrain; (ii) algorithms for terrain perception and robot locomotion over difficult terrain; (iii) the selection of a long-range inter-robot communication technology for a forest environment; and (iv) the design of the decentralized coordination strategies for the sparse swarm. We detail these challenges and introduce our ongoing work to address them.

4.1. Robot Platform Design

Although different forest environments may present different sets of requirements, the latter have typically the following ones in common: (i) the ability of the rover to progress fast through simple terrain; (ii) the ability to either overcome or avoid obstacles in its path, and (iii) long endurance. The need to traverse long distances requires energy efficient mobility, which is easiest achieved by rolling. For practicality, our interest is in rovers that are small enough to fit in a backpack. The size of obstacles the rover will be able to overcome is accordingly limited. Furthermore, the overall cost of each rover should be low enough that sizeable swarms are practical. In the context of these constraints the robot platform needs to address the challenges of mobility and communication.

4.1.1. Mechanical Design for Mobility

A rover that is well-adapted to the forest environment will provide a good trade-off between the ability to climb over obstructions to avoid detours at the reduced endurance that results from the extra weight of this climbing ability. We use an iterative design strategy where data on energy consumption

and mobility is gathered by teleoperated prototype platforms (**Figures 2A–E**). In addition to the on-board data collection, telemetry provides real-time feedback during such test runs to improve our understanding of what obstacles can be tackled by a particular rover design, what are suitable approaches to do so, and what is the energy expended for a particular path.

4.1.2. Hardware for Communication

Communication is the only form of direct interaction that is considered here. It needs to be scalable to many units and work over long range even with antennas located close to the ground. In many application scenarios the intra-swarm communication cannot be prioritized over other services. For radio communication these requirements point to limits in the frequency spectrum and transmission power that in combination with the range requirement lead to low channel capacity. Such a low capacity channel could be established over satellite communication or over text messages transmitted in a mobile phone network. However, a solution that does not rely on infrastructure is preferred, both from a cost and from an availability perspective. In the field of sensor networks and internet of things ultra-high-frequency radio technologies have recently come to the fore that aim for long range communication with low power requirements, scalability to several thousands nodes, and low hardware cost. One of these technologies, called long-range radio (LoRa), is particularly attractive in the present context of rovers. Our preliminary exploration of the suitability of LoRa for rovers operating at forest ground indicate that several hundred meters communication range is realistic at about 60 bytes per second. This is the case, even in the highly attenuating forest environment and with the ground plane effect inherent in a low antenna position (tip of antenna 17 cm above ground).

4.2. Locomotion on Difficult Terrain

Navigating off-trail in a forest environment is a challenging task and an open problem in the area of field robotics (Yang et al., 2018). The robots are required to assess their traversability on a priori unknown terrains in their proximity, relying solely on

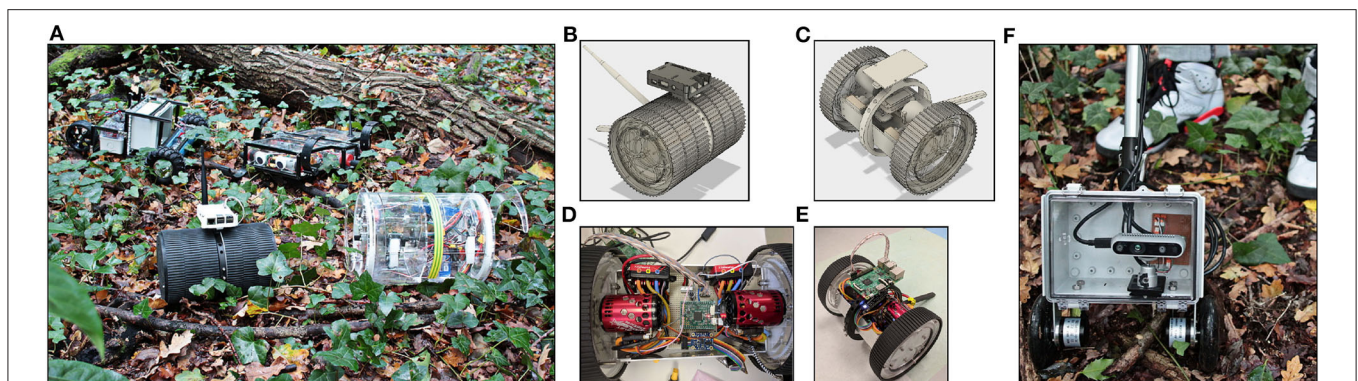


FIGURE 2 | Hardware platforms for the forest environment. Aside from four-/six-wheel drive, and tracks, other locomotion concepts are also investigated for their suitability on forest ground (**A**). The torque available to platforms with brushless motors (e.g., **B–E**) is helpful for tackling the ubiquitous small obstacles typical for this environment. Additional data for computer vision development is collected with a manual rig (**F**). Depth and color images are recorded with a global shutter camera (D435i, www.intel.com) to a laptop in a backpack. Meta data is collected from the camera's inertial measurement unit, rotary encoders on the wheels, and a GPS. A mobile phone mounted on the telescopic push rod gives remote access to the laptop.

onboard sensors under varying lighting and weather conditions, where GPS signal localization may not always be available. The problem is made further difficult by the varying nature of traversability; the traversability of a robot on a terrain depends not only on the innate characteristics of the terrain, but also on the dynamics of interaction between the robot and the terrain, which itself is susceptible to change (e.g., from a thick layer of mud stuck on the left side of a six-wheeled robot, or a damaged leg sustained by a quadruped robot).

Many studies have investigated terrain traversability for robot navigation algorithms in off-road environments, pioneered by the DARPA PerceptOR (Krotkov et al., 2007) and later the DARPA Learning Applied to Ground Vehicles (LAGR) programs (Huang et al., 2009a,b). The approaches developed for terrain traversability analysis use exteroceptive sensory information such as geometry-based and appearance-based features, as well as proprioceptive sensory information (Papadakis, 2013), and typically employ near-to-far type of learning algorithms (Bagnell et al., 2010) to predict traversable terrain for the robot. However, the robots employed in such off-road situations are relatively large (e.g., the DARPA LAGR vehicle was over 1 m in length and weighed around 100 kg), and equipped with expensive sensors such as radar, 2D lidar and multiple stereo cameras for off-road navigation (Jackel et al., 2006; Zhou et al., 2012; Milella et al., 2015; Santamaria-Navarro et al., 2015). In comparison, our small-scale low-cost robots running off-trail in the forest are faced with bigger challenges: Almost everything is an obstacle, and due to their small size the robots are much more likely to topple over. The development of computationally inexpensive computer vision and machine learning algorithms for the robots to efficiently locomote over a priori unknown terrains is part of our ongoing effort to realize our sparse swarm.

In addressing the traversability challenge we are in the process of developing a forest environment RGBD data-set, using a two-wheel mobile sensor platform (**Figure 2F**). The platform comprising an Intel D435i depth camera including an IMU, left and right wheel encoders, and GPS, is to be pushed manually along various off-trail “paths.” Our developed data-set is to be employed to train a depth estimation model, to predict depth with RGB image data from a monocular camera.

4.2.1. Terrain Traversability for a Single Robot

Using the depth-estimation model, the robots of the sparse swarm are required to learn closed-loop policies to efficiently traverse across different terrains. Forest terrain the robot may have to overcome include wet leaves on the forest floor, ditches with varying inclinations, muddy tracks and fallen tree branches. Challenges involved in learning locomotion behaviors for such terrain include investigating suitable representations for a closed-loop policy, characterizing metrics to estimate success of a policy in overcoming terrain, and accounting for progress between trials in evaluating multiple policies episodically on the robot. Trial-and-error based algorithms for rapid behavior adaptation (e.g., see Cully et al., 2015) appear to be a promising approach to begin addressing these challenges.

4.2.2. Collaborative Learning Across the Swarm

The available LoRa communication channel may be employed by the swarm for collaborative learning of traversable terrain in the forest environment. In such a transfer learning scenario, the robots of the swarm share information on their experiences traversing different terrains. Information shared may comprise metrics providing situational information on robot-terrain interaction, for instance energy consumption statistics, and the stability of the robot in traversing the terrain. Policies employed by robots to traverse terrain may also be shared, for recipient robots to bootstrap their exploration of new locomotion behaviors to adapt to changes in their proximal terrain. Additionally, in forest environments, some a priori unknown terrains may be unsafe for the robot to traverse over. The discovery of such terrains by the swarm may be accomplished by learning with “deliberative” catastrophic failure. Herein, the swarm may vote for one or a few robots to attempt to traverse over potentially hazardous terrain and share the resulting traversability information generated with the rest of the swarm.

4.3. Coordination in Communication-Constrained Environments

In the forest application scenario, the robots assume a linear formation that moves across a defined region. In a simple linear formation, the robots would occupy equidistant points on a line segment; each robot, bar the ones at the two ends, would have two neighbors. An alternative linear formation would place the robots alternately onto two parallel lines such that they form equilateral triangles; each robot within the formation would have four equidistant neighbors. To ease deployment, the robots could determine their order within the formation at run-time, for example, using their unique identifiers. While linear formations lend themselves for tasks such as coordinated search and coverage (Durham et al., 2012; Kolling et al., 2018), our scenario is particularly challenging, because the robots will be unable to interact with their neighbors for most of the time. Moreover, they do not know in advance the terrain to be encountered. This makes it difficult to predict individual progress. Some robots may have to take a detour of several hundreds of meters after discovering that a floodplain ahead of them is not traversable. To cope efficiently with these challenges, the robots need to move even when having had no recent contact with any neighbor. Yet, they should prevent the overall formation from becoming disconnected indefinitely. The robots could generate waypoints, and use the potential field method to approach them while avoiding obstacles. New way points could be suggested in an attempt to move the formation forward, and to repair it. The robots would use beliefs regarding their neighborhood, that is, which robots are present and their locations. Algorithms that allow robots to reestablish contact with lost, and potentially immobilized, members of the group could be considered (Banfi et al., 2018; Vandermeulen et al., 2018).

5. DISCUSSION

In this perspective article, we have highlighted the challenges that prevent most swarm robotic systems from transitioning to real-world applications. At present, robot swarms typically operate in highly controlled indoor laboratory environments. They are frequently interacting with each other, which is facilitated by their spatial proximity (e.g., 10 body lengths). Consequentially, such swarms are impractical for many real-world applications, in particular those, requiring the robots to act jointly over large distances (e.g., 1,000 body lengths). To address this problem, we have proposed the *sparse swarm* concept, which focuses on robot swarms that self-organize despite severe constraints regarding inter-robot proximity and coupling. Moreover, we have illustrated its use in a forest application scenario.

The sparse swarm concept opens up a number of theoretical questions. While sparse swarms are robot swarms, they are subject to additional constraints on inter-robot proximity and inter-robot coupling. A question to investigate is how the performance for a given swarm changes as these constraints are progressively enforced. A related question is how the minimal number of robots to exhibit self-organization changes as the constraints increase. For example, will swarms degenerate once the time for information to propagate is no longer polynomially bounded with the number of robots? Another question relates to the types of interactions. Where members of sparse swarms interact solely via non-situated communication, can they still spatially organize, for example, by sharing information on how to interact with the environment? And given the lack of spatial proximity, would the members of sparse swarms be required to encounter a similar set of environmental features (which could be empty) to exhibit self-organization? A further question relates to whether sparse swarms could be realized at all scales, with their members ranging in size from hundreds of meters (e.g., fleets of container ships) to micrometers (e.g., robot swarms within the human body).

For a sparse robot swarm to solve real-world problems in a land, sea, air or space environment, the individual robots are likely to require a high degree of autonomy and the ability to travel and to communicate over long distances. Depending on the environment and the task at hand the practical challenges to achieving the required capabilities differ. In environments that allow for energy harvesting (e.g., consider solar-powered

aerial drones or autonomous sailboats), endurance is not limited by power, but by the device life-time. As a consequence of the much increased deployment time across the sparse swarm, rare events can no longer be ignored. For such a system, what general strategies that broaden the ability of a system to recover from unforeseen situations (e.g., Cully et al., 2015) can be developed? Moreover, in many sparse swarm scenarios the channel capacity for communicating within the swarm is severely restricted (e.g., robots operating under water). How can the mismatch between the amount of data available from local sensors and the amount of data that can be received from others robots be reconciled for effective learning?

In conclusion, directly mimicking the densities and associated coordination strategies of natural swarms may be impractical for applications that require groups of robots to cover outdoor areas that are very large relative to their own size. We postulate that for these applications, swarm technologies need to be reconceptualized for robots to coordinate over large distances. Such coordination without any physical inter-robot interaction would require higher autonomy from individual robots of the swarm. Robots of the swarm would also require to traverse large distances to complete their mission, thus requiring low-cost, high-endurance hardware platforms. With this perspective article, we invite the robotics community to address the various challenges to bring sparse swarms to fruition.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

The work described was funded in part by an EPSRC New Investigator Award (EP/R030073/1) to DT and by the ECS Centre for Internet of Things and Pervasive Systems to K-PZ.

ACKNOWLEDGMENTS

The authors are grateful for the contributions of their students, in particular D. Jones (LoRa), J. Curran, J. Curry, T. Darlison, F. De Neve, M. Hunter, K. Jalundhwala D. Malyszko, R. Menezes, M. Oakley, Y. Yiangou (robots), C. Niu (vision).

REFERENCES

- Albani, D., Manoni, T., Arik, A., Nardi, D., and Trianni, V. (2019). "Field coverage for weed mapping: toward experiments with a UAV swarm," in *International Conference on Bio-inspired Information and Communication* (Pittsburgh, PA: Springer), 132–146. doi: 10.1007/978-3-030-24202-2_10
- Amigoni, F., Banfi, J., and Basilio, N. (2017). Multirobot exploration of communication-restricted environments: a survey. *IEEE Intell. Syst.* 32, 48–57. doi: 10.1109/MIS.2017.4531226
- Bagnell, J. A., Bradley, D., Silver, D., Sofman, B., and Stentz, A. (2010). Learning for autonomous navigation. *IEEE Robot. Autom. Mag.* 17, 74–84. doi: 10.1109/MRA.2010.936946
- Banfi, J., Basilio, N., and Amigoni, F. (2018). Multirobot reconnection on graphs: problem, complexity, and algorithms. *IEEE Trans. Robot.* 34, 1299–1314. doi: 10.1109/TRO.2018.2830418
- Bayındır, L. (2016). A review of swarm robotics tasks. *Neurocomputing* 172, 292–321. doi: 10.1016/j.neucom.2015.05.116
- Beetz, M., Civera, J., D'Andrea, R., Elfving, J., and Galvez-lopez, D. (2011). RoboEarth: a world wide web for robots. *IEEE Trans. Robot. Autom.* 6, 69–82. doi: 10.1109/MRA.2011.941632
- Beni, G. (2004). "From swarm intelligence to swarm robotics," in *International Workshop on Swarm Robotics* (Berlin: Springer), 1–9. doi: 10.1007/978-3-540-30552-1_1

- Bor, M., and Roedig, U. (2017). "LoRa transmission parameter selection," in *2017 13th International Conference on Distributed Computing in Sensor Systems DCOSS* (Los Alamitos, CA: IEEE Computer Society). doi: 10.1109/DCOSS.2017.10
- Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Bonabeau, E., and Theraula, G. (2003). *Self-Organization in Biological Systems*. Princeton, NJ: Princeton University Press.
- Cesare, K., Skeele, R., Yoo, S.-H., Zhang, Y., and Hollinger, G. (2015). "Multi-UAV exploration with limited communication and battery," in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA: IEEE), 2230–2235. doi: 10.1109/ICRA.2015.7139494
- Chamanbaz, M., Mateo, D., Zoss, B. M., Tokić, G., Wilhelm, E., Bouffanais, R., et al. (2017). Swarm-enabling technology for multi-robot systems. *Front. Robot. AI* 4:12. doi: 10.3389/frobt.2017.00012
- Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature* 521:503. doi: 10.1038/nature14422
- Duarte, M., Costa, V., Gomes, J., Rodrigues, T., Silva, F., Oliveira, S. M., et al. (2016). Evolution of collective behaviors for a real swarm of aquatic surface robots. *PLoS ONE* 11:e0151834. doi: 10.1371/journal.pone.0151834
- Ducatelle, F., Di Caro, G. A., Förster, A., Bonani, M., Dorigo, M., Magnenat, S., et al. (2014). Cooperative navigation in robotic swarms. *Swarm Intell.* 8, 1–33. doi: 10.1007/s11721-013-0089-4
- Durham, J. W., Franchi, A., and Bullo, F. (2012). Distributed pursuit-evasion without mapping or global localization via local frontiers. *Auton. Robots* 32, 81–95. doi: 10.1007/s10514-011-9260-1
- Garattoni, L., and Birattari, M. (2018). Autonomous task sequencing in a robot swarm. *Sci. Robot.* 3:eat0430. doi: 10.1126/scirobotics.eat0430
- Gauci, M., Chen, J., Li, W., Dodd, T. J., and Groß, R. (2014). Self-organized aggregation without computation. *Int. J. Robot. Res.* 33, 1145–1161. doi: 10.1177/0278364914525244
- Hollinger, G. A., and Singh, S. (2012). Multirobot coordination with periodic connectivity: theory and experiments. *IEEE Trans. Robot.* 28, 967–973. doi: 10.1109/TRO.2012.2190178
- Hu, G., Tay, W. P., and Wen, Y. (2012). Cloud robotics: architecture, challenges and applications. *IEEE Netw.* 26, 21–28. doi: 10.1109/MNET.2012.6201212
- Huang, W., Grudic, G., and Matthies, L. (2009a). Special issue: special issue on LAGR program, part I. *J. Field Robot.* 26, 1–113. doi: 10.1002/rob.v26:1
- Huang, W., Grudic, G., and Matthies, L. (2009b). Special issue: special issue on LAGR program, part II. *J. Field Robot.* 26, 115–240. doi: 10.1002/rob.20280
- Jackel, L. D., Krotkov, E., Perschbacher, M., Pippine, J., and Sullivan, C. (2006). The DARPA LAGR program: goals, challenges, methodology, and phase i results. *J. Field Robot.* 23, 945–973. doi: 10.1002/rob.20161
- Jones, S., Studley, M., Hauert, S., and Winfield, A. F. T. (2018). A two teraflop swarm. *Front. Robot. AI* 5:11. doi: 10.3389/frobt.2018.00011
- Kantaros, Y., and Zavlanos, M. M. (2017). Distributed intermittent connectivity control of mobile robot networks. *IEEE Trans. Autom. Control* 62, 3109–3121. doi: 10.1109/TAC.2016.2626400
- Kehoe, B., Patil, S., Abbeel, P., and Goldberg, K. (2015). A survey of research on cloud robotics and automation. *IEEE Trans. Autom. Sci. Eng.* 12, 398–409. doi: 10.1109/TASE.2014.2376492
- Knuth, D. E. (1976). Big omicron and big omega and big theta. *SIGACT News* 8, 18–24. doi: 10.1145/1008328.1008329
- Kolling, A., Kleiner, A., and Carpin, S. (2018). Coordinated search with multiple robots arranged in line formations. *IEEE Trans. Robot.* 34, 459–473. doi: 10.1109/TRO.2017.2776305
- Krotkov, E., Fish, S., Jackel, L., McBride, B., Perschbacher, M., and Pippine, J. (2007). The DARPA PerceptOR evaluation experiments. *Auton. Robots* 22, 19–35. doi: 10.1007/s10514-006-9000-0
- Luo, W., and Sycara, K. (2019). "Minimum k-connectivity maintenance for robust multi-robot systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau: IEEE), 7370–7377.
- Mathews, N., Christensen, A. L., O'Grady, R., Mondada, F., and Dorigo, M. (2017). Mergeable nervous systems for robots. *Nat. Commun.* 8:439. doi: 10.1038/s41467-017-01622-0
- Mathews, N., Christensen, A. L., Stranieri, A., Scheidler, A., and Dorigo, M. (2019). Supervised morphogenesis: exploiting morphological flexibility of self-assembling multirobot systems through cooperation with aerial robots. *Robot. Auton. Syst.* 112, 154–167. doi: 10.1016/j.robot.2018.11.007
- Milella, A., Reina, G., and Underwood, J. (2015). A self-learning framework for statistical ground classification using radar and monocular vision. *J. Field Robot.* 32, 20–41. doi: 10.1002/rob.21512
- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klapotcz, A., et al. (2009). "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions (IPCB: Instituto Politécnico de Castelo Branco)*, 59–65.
- Nouyan, S., Groß, R., Bonani, M., Mondada, F., and Dorigo, M. (2009). Teamwork in self-organized robot colonies. *IEEE Trans. Evol. Comput.* 13, 695–711. doi: 10.1109/TEVC.2008.2011746
- Papadakis, P. (2013). Terrain traversability analysis methods for unmanned ground vehicles: a survey. *Eng. Appl. Artif. Intell.* 26, 1373–1385. doi: 10.1016/j.engappai.2013.01.006
- Pei, Y., Mutka, M. W., and Xi, N. (2013). Connectivity and bandwidth-aware real-time exploration in mobile robot networks. *Wireless Commun. Mobile Comput.* 13, 847–863. doi: 10.1002/wcm.1145
- Pickem, D., Grotfelter, P., Wang, L., Mote, M., Ames, A., Feron, E., et al. (2017). "The Robotarium: a remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Singapore: IEEE), 1699–1706. doi: 10.1109/ICRA.2017.7989200
- Rubenstein, M., Cornejo, A., and Nagpal, R. (2014). Programmable self-assembly in a thousand-robot swarm. *Science* 345, 795–799. doi: 10.1126/science.1254295
- Sahin, E. (2004). "Swarm robotics: From sources of inspiration to domains of application," in *International Workshop on Swarm Robotics* (Berlin: Springer), 10–20. doi: 10.1007/978-3-540-30552-1_2
- Santamaria-Navarro, A., Teniente, E. H., Morta, M., and Andrade-Cetto, J. (2015). Terrain classification in complex three-dimensional outdoor environments. *J. Field Robot.* 32, 42–60. doi: 10.1002/rob.21521
- Schmickl, T., and Crailsheim, K. (2008). Trophallaxis within a robotic swarm: bio-inspired communication among robots in a swarm. *Auton. Robots* 25, 171–188. doi: 10.1007/s10514-007-9073-4
- Schranz, M., Umlauf, M., Send, M., and Elmenreich, W. (2020). Swarm robotic behaviors and current applications. *Front. Robot. AI* 7:36. doi: 10.3389/frobt.2020.00036
- Semtech (2015). *LoRa Modulation Basics*. Application Note AN1200.22, Semtech Corporation, Camarillo, CA.
- Tardioli, D., Mosteo, A. R., Riazuelo, L., Villarroel, J. L., and Montano, L. (2010). Enforcing network connectivity in robot team missions. *Int. J. Robot. Res.* 29, 460–480. doi: 10.1177/0278364909358274
- Tardioli, D., Riazuelo, L., Sicignano, D., Rizzo, C., Lera, F., Villarroel, J. L., et al. (2019). Ground robotics in tunnels: keys and lessons learned after 10 years of research and experiments. *J. Field Robot.* 36, 1074–1101. doi: 10.1002/rob.21871
- Tardioli, D., Sicignano, D., Riazuelo, L., Romeo, A., Villarroel, J. L., and Montano, L. (2016). Robot teams for intervention in confined and structured environments. *J. Field Robot.* 33, 765–801. doi: 10.1002/rob.21577
- Valentini, G., Ferrante, E., Hamann, H., and Dorigo, M. (2016). Collective decision with 100 kilobots: speed versus accuracy in binary discrimination problems. *Auton. Agents Multi Agent Syst.* 30, 553–580. doi: 10.1007/s10458-015-9323-3
- Vandermeulen, I., Groß, R., and Kolling, A. (2018). "Re-establishing communication in teams of mobile robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE), 7947–7954. doi: 10.1109/IROS.2018.8594460
- Virág, C., Vászárhelyi, G., Tarcai, N., Szörényi, T., Somorjai, G., Nepusz, T., et al. (2014). Flocking algorithm for autonomous flying robots. *Bioinspir. Biomimet.* 9:025012. doi: 10.1088/1748-3182/9/2/025012
- Wan, J., Tang, S., Yan, H., Li, D., Wang, S., and Vasilakos, A. V. (2016). Cloud robotics: current status and open issues. *IEEE Access* 4, 2797–2807. doi: 10.1109/ACCESS.2016.2574979
- Wang, Z., and Schwager, M. (2016). Force-Amplifying N-robot Transport System (Force-ANTS) for cooperative planar manipulation without communication. *Int. J. Robot. Res.* 35, 1564–1586. doi: 10.1177/0278364916667473
- Weinstein, A., Cho, A., Loianno, G., and Kumar, V. (2018). Visual inertial odometry swarm: An autonomous swarm of vision-based quadrotors. *IEEE Robot. Autom. Lett.* 3, 1801–1807.
- Werfel, J., Petersen, K., and Nagpal, R. (2014). Designing collective behavior in a termite-inspired robot construction team. *Science* 343, 754–758. doi: 10.1126/science.1245842

- Yang, G.-Z., Bellingham, J., Dupont, P. E., Fischer, P., Floridi, L., Full, R., et al. (2018). The grand challenges of science robotics. *Sci. Robot.* 3:ear7650. doi: 10.1126/scirobotics.aar7650
- Zhou, S., Xi, J., McDaniel, M. W., Nishihata, T., Salesses, P., and Iagnemma, K. (2012). Self-supervised learning to visually detect terrain surfaces for autonomous robots operating in forested terrain. *J. Field Robot.* 29, 277–297. doi: 10.1002/rob.21417
- Zoss, B. M., Mateo, D., Kuan, Y. K., Tokić, G., Chamanbaz, M., Goh, L., et al. (2018). Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies. *Auton. Robots* 42, 1669–1689. doi: 10.1007/s10514-018-9702-0

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Tarapore, Groß and Zauner. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.



Adaptive Foraging in Dynamic Environments Using Scale-Free Interaction Networks

Ilja Rausch*, Pieter Simoens and Yara Khaluf

IDLab - Department of Information Technology, Ghent University—IMEC, Ghent, Belgium

OPEN ACCESS

Edited by:

Nicolas Bredeche,
Université Pierre et Marie Curie,
France

Reviewed by:

Alan Gregory Millard,
University of Lincoln, United Kingdom
Amine Boumaza,
UMR7503 Laboratoire Lorrain de
Recherche en Informatique et ses
Applications (LORIA), France
Payam Zahadat,
University of Graz, Austria

*Correspondence:

Ilja Rausch
Ilja.Rausch@UGent.be

Specialty section:

This article was submitted to
Multi-Robot Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 22 December 2019

Accepted: 02 June 2020

Published: 09 July 2020

Citation:

Rausch I, Simoens P and Khaluf Y
(2020) Adaptive Foraging in Dynamic
Environments Using Scale-Free
Interaction Networks.
Front. Robot. AI 7:86.
doi: 10.3389/frobt.2020.00086

Group interactions are widely observed in nature to optimize a set of critical collective behaviors, most notably sensing and decision making in uncertain environments. Nevertheless, these interactions are commonly modeled using local (proximity) networks, in which individuals interact within a certain spatial range. Recently, other interaction topologies have been revealed to support the emergence of higher levels of scalability and rapid information exchange. One prominent example is scale-free networks. In this study, we aim to examine the impact of scale-free communication when implemented for a swarm foraging task in dynamic environments. We model dynamic (uncertain) environments in terms of changes in food density and analyze the collective response of a simulated swarm with communication topology given by either proximity or scale-free networks. Our results suggest that scale-free networks accelerate the process of building up a rapid collective response to cope with the environment changes. However, this comes at the cost of lower coherence of the collective decision. Moreover, our findings suggest that the use of scale-free networks can improve swarm performance due to two side-effects introduced by using long-range interactions and frequent network regeneration. The former is a topological consequence, while the latter is a necessity due to robot motion. These two effects lead to reduced spatial correlations of a robot's behavior with its neighborhood and to an enhanced opinion mixing, i.e., more diversified information sampling. These insights were obtained by comparing the swarm performance in presence of scale-free networks to scenarios with alternative network topologies, and proximity networks with and without packet loss.

Keywords: swarm robotics, foraging, collective decision-making, scale-free networks, dynamic environments, adaptive swarm

1. INTRODUCTION

The efficiency of the information sharing mechanisms used by individuals during group decision processes determines to a large extent the fitness of the group decision. In nature, collective systems consist of a high number of individuals living in large and unknown environments, and needing to perform complex tasks to survive. Among the many examples of collective decision-making is choosing a new site to build their home (Richardson et al., 2018), or deciding among a number of foraging patches (Michelena et al., 2009). Despite the high diversity of tasks, uncertainty and complexity are common features. Hence, individuals apply information pooling to mitigate uncertainty and increase decision accuracy (Conradt, 2011). Achieving efficient opinion sampling

depends to a large extent on the network topology that defines the interaction structure and opinion sharing of these individuals (Khaluf et al., 2018; Rausch et al., 2019b). The use of such network is fundamental for collective decision-making. It is generally exploited at two stages of the process (i) when spreading information on one or multiple stimuli that are initially perceived by a limited number of individuals that are able to trigger the collective decision process—e.g., a predator attack—; and (ii) when spreading the individuals' opinions or choices to achieve consensus (Khaluf et al., 2019a).

In artificial systems such as swarm robotics, collective decision-making is mostly designed in static environments (Bayındır, 2016), where options and their qualities are defined at the beginning and do not change over time. In these studies the focus is mainly on the design of efficient voting mechanisms that enable a high level of decision coherence within the shortest time possible (Khaluf et al., 2018). Alternatively, other studies were addressing the design of decision strategies that tackle the accuracy vs. speed trade-off (Valentini, 2017)—i.e., taking longer time to gather enough information and making more accurate decisions vs. exploiting the available information and taking the decision as soon as possible. In both cases, the speed of converging on a decision is a fundamental goal in the design of decision-making. The decision speed strongly depends on the interaction topology the individuals are part of, to spread stimuli or opinions during the decision-making process. Interactions in collective systems are frequently modeled using local (i.e., proximity) communication, where the neighborhood of an individual is defined spatially based on their interaction range, i.e., interacting with all peers within the individual's communication radius. Nevertheless, other interaction models such as scale-free networks were revealed in several real-world examples (Albert and Barabási, 2002; Holme, 2019). A comprehensive review on scale-free phenomena in a more general context can be found in Khaluf et al. (2017a). In various works, scale-free networks enable scalable, fast and efficient information transfer. For example, in Goh et al. (2001), authors showed how the betweenness centrality scales with the scale-free exponent. Other works showed how the ultrasmall diameter of the scale-free networks contributes to their efficiency in information transmission (Cohen and Havlin, 2003; Thivierge, 2014). Finally, scale-free topologies were studied in natural collective systems such as in Cavagna et al. (2010). In this work, the authors studied starlings flocks and suggest that collective response to predator's attacks may be achieved through scale-free behavioral correlations. Based on these studies, we extend the application of scale-free networks to artificial swarms in order to investigate the role these networks can play in improving a swarm's collective decision-making process.

A key aspect of scale-free networks is the presence of hubs—i.e., nodes with a comparably high connectivity degree—(Albert et al., 2000; Albert and Barabási, 2002). Hubs represent a small percentage of the network nodes, however, their high connectivity leads to a small network diameter. This facilitates efficient communication by enabling any two random nodes to share information over only few hops, resulting in fast information transfer (Cohen and Havlin, 2003). In this paper,

we exploit this critical feature of scale-free networks to help collective systems to faster respond to changes in dynamic environments. In dynamic environments, conditions change over time and hence, the collective system needs to adapt its behavior within a short period of time in order to survive. We refer to this as the collective response time. In our study, this is the time required for the group to collectively change the intensity of its foraging activities as a response to a change in the availability of the food items.

Among many examples of collective tasks in natural systems, we select *foraging* (Liu et al., 2007) and perform our study using a simulated population of swarming robots. Foraging is a complex task used by many species to retrieve food to their homes, but beyond that it is a metaphor for many real-world robotics tasks such as search and rescue, retrieve materials for collective construction and others. In foraging, individuals (robots) need to continuously make a decision between staying at their base or leaving to forage for food items. A large body of literature has been dedicated to investigate foraging in artificial systems such as swarm robotics. These studies have addressed various research questions such as the foraging performance under the influence of physical robot interference (Lerman and Galstyan, 2002; Khaluf et al., 2016), the multi-foraging task (Campo and Dorigo, 2007)—i.e., the foraging for different types of items—or consensus achievement (Hoff et al., 2013; Khaluf et al., 2017b). Additionally, some studies have focused on how to optimize the task allocation in foraging using cost functions (Pini et al., 2013; Khaluf et al., 2019b). Also how to investigate simple probabilistic models that rely on the foraging success probability in achieving an efficient foraging behavior (Pinciroli et al., 2012). Other studies have gone further to investigate whether the performance of swarms in the foraging tasks bears a particular characteristic distribution (e.g., a power law) for any of its time or space features (Khaluf and Dorigo, 2016; Rausch et al., 2019a). Despite this intensive research effort, foraging of robot swarms in dynamic environments and the influence of different interaction models are still not well understood. However, these questions are paramount, given the prevalence of scale-free phenomena in real-world systems and admitting that most real environments are dynamic. Therefore, in this paper, we focus on the fundamental question of how the integration of a scale-free interaction structure may influence the collective response of simulated swarms to changes in food density within the foraging environment. We approach this question by analyzing the speed and coherence of the collective response to those changes. We begin with defining the robot (microscopic) and the swarm (macroscopic) behaviors in sections 2.1, 2.3, respectively. The details on generating scale-free networks from local neighborhoods are given in section 2.2. In section 2.4, we describe the experimental setup. Thereafter, in section 3 we compare the collective response of the swarm in presence and absence of scale-free interactions. We discuss our findings that suggest that the use of scale-free interactions can be advantageous due to (i) reduced correlations between a robot's decisions and those of its spatial neighbors and (ii) enhanced information spread through long-range interactions and frequent rewiring of communication links. These insights are obtained by comparing

the influence of scale-free networks to scenarios with alternative random networks as well as scenarios that include packet loss. Conclusions are drawn in section 4.

2. METHODS

2.1. Robot Behavior

Robots are placed in an arena that is divided into two areas: the nest and the foraging environment. Inspired by the behavior observed in harvester ants *Pogonomyrmex barbatus* (Schafer et al., 2006; Pinter-Wollman et al., 2013), each robot can switch between two essential states: *resting* and *foraging*. In the foraging state, the robot attempts to find a food item inside the foraging environment by performing a pseudo-random walk. In particular, the robot moves on a straight line until it encounters another robot or an obstacle (e.g., a wall), in which case a *collision avoidance* maneuver is initiated. By executing this maneuver, the robot attempts to move in the direction of least physical interference, as sensed by its proximity sensors. After executing the collision avoidance maneuver, the robot goes back to its standard motion following a straight line. When the robot encounters a food item, it collects this item and retrieves it back to the nest where the robot rests for a given period of time θ_r .

In the resting state, the robot remains inside the nest, which is the only area where communication with other robots can take place. This is inspired by several natural systems, in which the communication occurs mainly inside the nest or the hive (Liu et al., 2007; Seeley et al., 2012; Reina et al., 2015; Valentini et al., 2016). This approach accommodates two relevant properties of foraging systems: (i) it is common that the foraging environment is significantly larger than the nest area, and hence, individual encountering rates outside the nest are negligibly low. (ii) Due to the high density of individuals inside the nest there is a high likelihood of interaction between individuals that have explored different parts of the foraging environment, and hence a more diversified sample of information about the environment can be collected.

Robots can communicate only with neighbors that are within a direct line of sight, sharing their individual experiences. This is a continuous process—i.e., each robot broadcasts at every time step its previous experience (success or failure in finding a food item) until it switches again to the foraging state. Continuous communication activity is a required choice of the experiment design to research the role of network topology in the emergent behavior (Rausch et al., 2019a).

All robots, in our study, are identical and each robot is a probabilistic finite state machine. In particular, a robot's behavior is shaped by two switching probabilities that describe at every time step the robot's likelihood to switch from foraging to resting ($P_{f \rightarrow r}$) or the opposite ($P_{r \rightarrow f}$). These probabilities are updated differently at the robot's resting and foraging states. At the foraging state, the switching probabilities are updated using the robot's foraging experience. The impact of this experience on the robot's decision-making is given by the set of two individual cues $\{i_f, i_r\} \in \mathbb{R}_0^+ \times \mathbb{R}_0^+$. More specifically, the cue i_f defines a numerical value by which the probability to switch from resting to foraging ($P_{r \rightarrow f}$) is increased when the robot has

experienced foraging success—i.e., a discovered food item during the latest foraging attempt. The same value is used to decrease this switching probability in case of a failed foraging attempt, i.e., when the robot has spent a specific time (θ_f) foraging without finding a food item. The cue i_r updates the robot's switching probability from foraging to resting ($P_{f \rightarrow r}$) in a manner that is inverse to i_f . Besides updating the switching probabilities at the foraging state, the robot updates those while resting. This update is performed using the experience received from the robot's neighbors and is numerically given by two social cues $\{s_f, s_r\} \in \mathbb{R}_0^+ \times \mathbb{R}_0^+$. The social cue s_f is used to update the switching probability from resting to foraging (i.e., $P_{r \rightarrow f}$) by increasing (decreasing) $P_{r \rightarrow f}$ when the robot's neighbors report primarily on successful (failed) foraging attempts. Whereas, s_r is used to update the switching probability from foraging to resting (i.e., $P_{f \rightarrow r}$), inversely to s_f . In the following we define how the switching probabilities are updated at every simulation step (as described in Rausch et al., 2019a; to prevent divergence, both probabilities were truncated between zero and one):

$$P_{r \rightarrow f}(t+1) = P_{r \rightarrow f}(t) + \delta_\eta(t)s_f + \delta_\phi(t)i_f \quad (1)$$

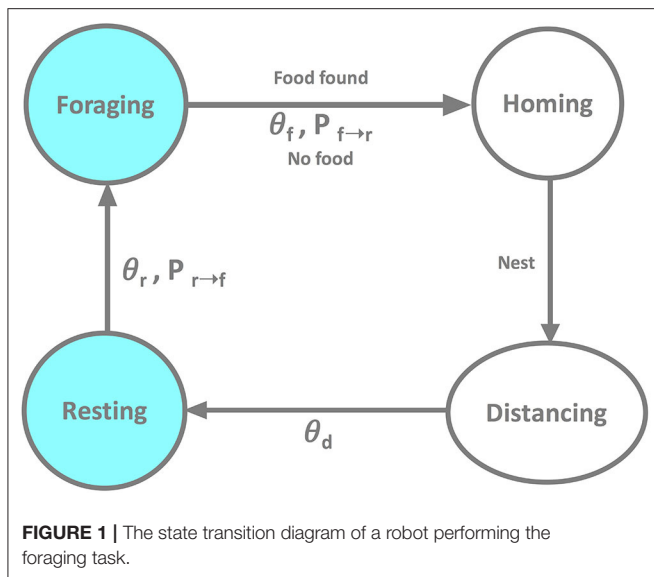
$$P_{f \rightarrow r}(t+1) = P_{f \rightarrow r}(t) - \delta_\eta(t)s_r - \delta_\phi(t)i_r, \quad (2)$$

where $\delta_\eta(t)$ is the difference between the successful and the failed foraging attempts communicated to the robot by its neighbors. Hence, it has a positive sign when there are more successful attempts than failed ones and a negative sign otherwise. Consequently, the former increases the switching probability from resting to foraging and the latter increases the switching probability from foraging to resting. $\delta_\eta(t) = 0$ if the robot is not resting. Additionally, the robot's individual experience during a foraging attempt that starts at t_f is defined as follows:

$$\delta_\phi(t) = \begin{cases} +1, & \text{at } t_{if} \\ 0, & \text{if } t_f < t \leq t_f + \theta_f \text{ \& no item is found} \\ -1, & \text{if } t > t_f + \theta_f \text{ \& the robot is still foraging} \end{cases} \quad (3)$$

where t_{if} is the (unique) time step at which the robot finds an item while in foraging state. While in the foraging state, the robot may find an item at any time $t_f < t_{if}$ (i.e., it could also happen that $t_f + \theta_f < t_{if}$). After finding an item, i.e., subsequently to t_{if} , the robot leaves the foraging state. If no item is found and the foraging time crosses the threshold θ_f , then $\delta_\phi(t) = -1$. This increases $P_{f \rightarrow r}(t)$ at every time step $t > t_f + \theta_f$, guaranteeing that the robot will probabilistically leave the foraging state at some t , even without finding an item. $\delta_\phi(t) = 0$ outside of the foraging state.

The robot behavior is illustrated in **Figure 1** using a state diagram. It includes the following states: (i) foraging: after having spent at least θ_r time steps resting, the robot switches with probability $P_{r \rightarrow f}$ from resting to foraging. It attempts to search the foraging area for a food item to retrieve to the nest. If the robot fails to find a food item within a predefined time θ_f , it switches with probability $P_{f \rightarrow r}$ to homing; (ii) homing: in this transitional state the robot returns to the nest, with $\delta_\eta(t) = 0$ and $\delta_\phi(t) = 0$; as soon as the robot reaches nest, it switches



to distancing; (iii) distancing: having returned to the nest, the robot searches for an empty spot in the nest where it can rest; similar to the homing state, distancing is a transitional state with $\delta_\eta(t) = 0$ and $\delta_\phi(t) = 0$; distancing terminates after θ_d time steps and the robot switches to resting; (iv) resting: subsequent to distancing the robot rests for at least θ_r time steps after which it switches with probability $P_{r \rightarrow f}$ to foraging. A resting robot broadcasts “success” (or “failure”) to its neighbors if the latest foraging attempt was successful (or not), respectively. If the robot failed to leave the nest in state (i), it has no information about the foraging environment and, thus, does not broadcast any message. Throughout the entire experiment, the robot performs collision avoidance maneuvers if other robots or walls enter its proximity sensors’ range (not shown in **Figure 1** for better readability).

2.2. Robot Scale-Free Communication Network

In this section, we describe the design and implementation of the algorithm that leads to a scale-free robot communication network. An implementation of this algorithm in C++ is publicly available online¹ (Rausch et al., 2020). The generation of a scale-free network from local neighborhoods is an iterative process, where at each time step t the robot communication is updated according to the following procedure:

1. Identify all connected components (CCs) in the resting swarm using depth-first-search. A CC is the maximal set of nodes (robots), where each two nodes are connected through a finite path. The CCs are initially derived from the spatial networks in which the robots are neighbors if they are within each other’s communication radius.
2. Generate the scale-free network topology within a CC using preferential attachment (Albert and Barabási, 2002) as summarized in Algorithm 1. This algorithm is largely inspired by previously proposed approaches (Li and Chen, 2003; Jiang

et al., 2014). We begin by selecting a sink node $v_{s,0}$ which is the node with the highest number of neighbors within its spatial proximity—i.e., within the initial radius of $r_s = 1.25$ m. Within this r_s , each spatial neighbor $v_{s,i}$ is linked to $v_{s,0}$, creating an initial sink network G_s . Next, we increase r_s by 0.2 m. Due to this increase, new nodes v_{new} enter r_s . Each v_{new} is connected to any v_s following preferential attachment. In a preferential attachment process, the higher the degree of node v_s compared to the sum of all node degrees within G_s , the more likely is v_{new} to connect to v_s . After all v_{new} were added to G_s , r_s is increased again by 0.2 m. This process continues until G_s is of the same size as CC.

3. Repeat 2. for every CC in the swarm.

In Algorithm 1, N_{sink} is the size of the sink network G_s , in terms of the number of nodes. Similarly, N_{CC} is the size of the selected connected component; d_s is the degree of node v_s , and $\sum_i d_i$ is the sum over all degrees in the sink-network. Note that the robot communication approaches the scale-free network topology only for large enough CC. However, due to the relatively small area of the nest the robots had a high tendency to self-aggregate into a giant connected component.

To test how successful Algorithm 1 was in generating a scale-free topology, we recorded the degree distributions at $t = 10$ of 1,000 simulation runs. At $t = 10$ the large majority of robots was still resting inside the nest, providing us with at least one large CC. Scale-free networks are characterized by the power law degree distribution. Thus, we tested whether our recorded degree distributions follow the power law using previously established statistical methods (Clauset et al., 2009; Broido and Clauset, 2019; Rausch et al., 2019a). Essentially, this statistical analysis is a highly rigorous power law fitting procedure that consists of three critical steps: (i) testing whether the shape of the distribution is due to random fluctuations, i.e., testing the *goodness-of-fit* given by a p -value. We proceed to the next step only if $p < 0.1$, otherwise the power law fit is considered unreliable. (ii) As the power law behavior is commonly found at the tail of the distribution, we proceed to the third step only if the data that is fit the power law behavior represents at least 10% of all data points. (iii) Finally, we compare the power law fit to other common distributions (such as the exponential or the log-normal) that may also tend to resemble a linear shape on a log-log scale (which is characteristic for the power law) (Clauset et al., 2009; Alstott et al., 2014). This is done by considering the log-likelihood ratio of each pair of distributions, which has a negative value if the distribution we compare the power law to is a significantly better fit. Consequently, the hypothesis that the data is power law distributed is not rejected only if this log-likelihood ratio is positive and only if we did not reject it at steps (i) and (ii). The result of the testing procedure can be captured by a numeric value to categorize whether the support for the hypothesis is not present, weak, moderate or strong (for more details see Rausch et al., 2019a). The test results for Algorithm 1 have shown a statistically sound support for the power law distribution in 76% of tests (we ran 1,000 tests), suggesting that Algorithm 1 was considerably successful in creating scale-free networks.

¹<https://osf.io/48b9h/>

Algorithm 1: Pseudo-code for the implementation of the preferential attachment, executed at each time step.

```

initialize a sink network  $G_s \subseteq CC$ :
Choose a node  $v \in CC$  with the highest degree and connect it
to its spatial neighbors within the radius  $r_s$  around  $v$ .
while  $N_{sink} < N_{cc}$  do
  increase  $r_s$ 
  for each  $v_{new} \in CC$  do
    for each  $v_s \in G_s$  do
      create a bi-directional link between  $v_{new}$  and  $v_s$ 
      with probability  $P_{BA} = d_s / \sum_i d_i$ 
      if no link created then
         $P_{BA} \leftarrow P_{BA} + d_s / \sum_i d_i$ 
      end
    end
  end
end

```

Alternatively, one can use Algorithm 1 to construct networks with a degree distribution that is less skewed than power law and more symmetric around the mean degree, i.e., networks that resemble more closely the well-known small-world networks. To this end, one can simply replace the preferential attachment component $d_s / \sum_i d_i$ by a real number.

2.3. Swarm Behavior

At the swarm level, the foraging behavior emerges as a result of complex interactions between the robots as well as between robots and their environment. As mentioned above, we evaluate this performance in dynamic environments, in which the food density is subject to single and periodic changes. The quality of the emergent performance is evaluated with respect to the swarm response (adaptivity) to the changing number of items in the foraging environment. In particular, we define the swarm performance with respect to (i) the speed of the swarm's collective response, and (ii) the number of retrieved items. The collective response is quantified using the number of resting robots at any time step. For instance, in case of a sudden high availability of food items an ideal swarm's response would be to allocate more robots to the foraging state shortly after the increase in the number of food items is detected.

We borrow the term of settling time from control theory to measure the time of the swarm's collective response, referred to as the convergence time—i.e., the time the swarm needs to adapt the number of resting/foraging robots to any change in the items density. The settling time is defined as the time elapsed from the moment of applying a particular stimulus (i.e., changing the items' density) to the time the system output (i.e., number of robots N_{rest} that are in the resting state) reaches and remains within a specified margin of error. Hence, the time to convergence is computed as in the following:

$$t_{conv} = \inf\{S\},$$

$$\text{where } S = \{t : |Fn(N_{rest}(t)) - Fn(N_{rest}(t_{steady}))| < \zeta\}, \quad (4)$$

TABLE 1 | Robot and arena parameters.

Parameter	Value
ROBOT PARAMETERS	
Physical avoidance range	0.1 m
Communication range	1.25 m
Maximum moving speed	1 m/s
Minimum resting time θ_r	100 s
Minimum unsuccessful foraging time θ_f	500 s
Minimum distancing time θ_d	100 s
Individual cues i_r, i_f	0.01
Social cues s_r, s_f	{0.01, 0.25, 0.99}
ARENA PARAMETERS	
Total area of the arena A	$50 \times 50 \text{ m}^2$
Area of the Nest A_n	$10 \times 50 \text{ m}^2$
Area of the Foraging environment A_f	$40 \times 50 \text{ m}^2$
Number of robots N_{robots}	950
Number of items N_{items}	30 or 300
Total experiment duration T	10^4 ts

where $\inf\{S\}$ is the greatest lower bound of the set S , and the set S includes all time steps t at which the difference between the transformed number of resting robots at a specific time step $N_{rest}(t)$ and the transformed number of resting robots at the steady state $N_{rest}(t_{steady})$ is smaller than a threshold ζ . In our study we set $\zeta = 0.1$. Here, t_{steady} is the time step at which the system reaches its steady state. To compute the time to convergence, we use the matlab tool STEPINFO², that first applies $Fn(\dots)$ to transform the input into a continuous representation. This transformation was used for N_{rest} .

Finally, in addition to the convergence time, we investigate the swarm performance in terms of the number of retrieved items. The number of retrieved items is strongly related to the time to convergence, since a faster convergence implies a higher efficiency in retrieving items. We compute this performance measure using the cumulative sum of the items retrieved over time.

2.4. Simulation Setup

We ran the simulations using ARGOS³, a well-established physics-based simulator for swarm robotics (Pinciroli et al., 2012). The values of particular parameter settings that can be used to reproduce our simulations and results are listed in **Table 1**. Additionally, the reader is encouraged to find our project on the Open Science Framework⁴ (Rausch et al., 2020) to download the development sources and run the simulations.

Figure 2 displays snapshots from simulations with proximity (**Figure 2A**) and scale-free (**Figure 2B**) networks. The square-shaped arena is of the size $L \times L$ ($L = 50 \text{ m}$) and consists of the nest $A_n = 10 \times 50 \text{ m}^2$ (gray colored floor in **Figure 2**) in addition to the foraging environment $A_f = 40 \times 50 \text{ m}^2$ (white

²<https://www.mathworks.com/help/control/ref/stepinfo.html>

³<http://www.argos-sim.info/>

⁴<https://osf.io/48b9h/>

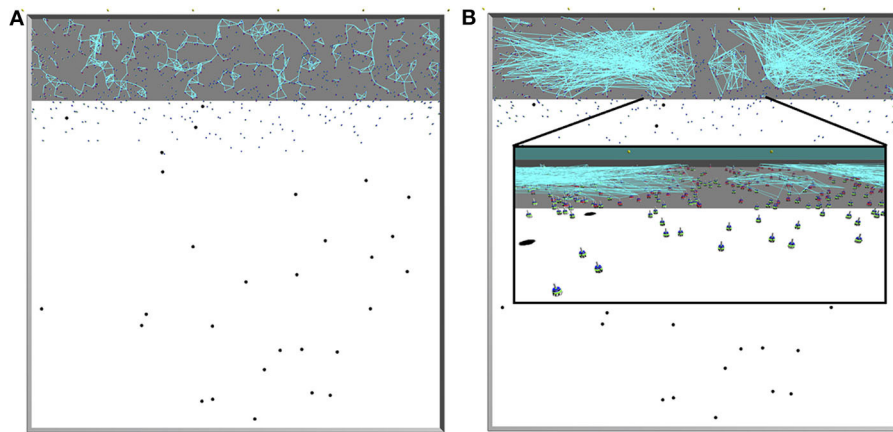


FIGURE 2 | Illustrations of the arena taken from ARGoS simulations. Gray area: nest; white area: foraging environment; black dots: items; blue objects: Footbots; light-blue lines: communication (range-and-bearing) links. Top views onto the entire arena; the communication network is constructed in **(A)** using spatial network given by the local robot interactions, and in **(B)** using Algorithm 1; the inset shows a close-up view on the robots. In all figures, the communication links are formed *only for resting robots inside the nest*, as in our experiments moving robots neither broadcast nor listen to any messages. Therefore, it can happen that although a robot is within the communication range of another, no communication link is established between the two.

in **Figure 2**). Inside the foraging environment, food items are uniformly distributed. When a robot brings a food item to the nest, a new food item appears at a random location within the foraging environment, preventing item depletion that might lead the foraging activity to halt.

The robots are able to rapidly leave or return to the nest thanks to the phototaxis behavior. For that purpose, light beacons are installed on one side of the nest, opposite to the foraging environment (yellow dots at the top of **Figure 2A** or **Figure 2B**). Robots are repelled from the lights whenever they need to leave the nest, and attracted to the lights to return to the nest. The swarm consists of N_{robots} homogeneous robots (we use Footbots; Bonani et al., 2010). Robots are equipped with probabilistic controllers, which tune their behavior to forage or rest based on the above mentioned probabilities (i.e., $P_{r \rightarrow f}$ and $P_{f \rightarrow r}$).

To implement the proposed networks (i.e., scale-free and proximity), we utilize the range-and-bearing medium (that includes sensor and actuator) provided in ARGoS. However, this communication medium is used differently for the two networks. In the case of proximity networks, the communication range of the range-and-bearing medium is set to 1.25 m (as we can see in **Table 1**). In the case of the scale-free networks, at each time step, we first obtain the connected components using the spatial proximity network, where the robots communicate via the range-and-bearing medium within a radius of 1.25 m. In the same time step, for each of these connected components, we create a scale-free network in which the connections can span over the entire length of the nest, if the connected component spans over that area. Thus, the resulting scale-free networks can include much longer ranges than 1.25 m. For implementing such a communication topology in real-world swarms, it is possible to apply other communication systems than the range-and-bearing medium, such as other radio communication technologies (e.g., the well-established wifi Li et al., 2008), shared memory

(Bayındır, 2016) or promising concepts such as the augmented reality for Kilobots (ARK) (Reina et al., 2017).

3. RESULTS AND DISCUSSION

The goal of this study is to investigate the influence of the scale-free topology on the collective performance and response of a swarm foraging in a dynamic environment. The dynamics of the environment is modeled in terms of single and periodic changes in the food density. In robot swarms, the interaction among individuals is mostly modeled using local communications, where each robot has a limited communication range. The communication range is usually much smaller than the dimension of the world. The robot's neighborhood is defined as the set (or a subset) of robots that is located within its communication range. In this study, besides local interactions, we make use of the well-known preferential attachment mechanism (applied in Algorithm 1, see section 2.2) to construct a scale-free topology that accelerates information sharing. Hence, we investigate whether it may improve the efficiency of the swarm collective response to environmental dynamics.

As mentioned above, we define the collective response in terms of the number of resting robots and measure it as the change in this number over time. In our experiments, initially, the entire swarm is in the resting state. In the following, a transient period begins, during which the swarm displays oscillations at the group level. First, almost all robots begin foraging during the first 500 time steps (ts)—Note that a simulated time step is one second, with one tick per second. Within the subsequent ≈ 500 ts most of the swarm individuals come back to the nest and switch to resting. Even though such collective behavior oscillates over several following time periods—due to the probabilistic nature of the robot controller—the coherence increases rapidly and the swarm converges on a relatively stable number of resting

robots. The duration of this transient period is mostly shorter than $5 \cdot 10^3$ ts, after which we begin our measurements. Finally, based on preliminary results, we set the swarm size to $N = 950$, which balances physical interference with swarm performance and delivers a sufficiently large number of samples for statistically sound analysis.

We use two experimental settings. In the first setting, after the system converges on a number of resting robots N_{rest} (number of foraging robots is then $N_{forg} = N - N_{rest}$), a single external stimulus is applied. This stimulus represents an increase in the number of food items N_{items} by the factor of 10 (from 30 to 300 items) at a particular time point t_{crit} . In the second experimental settings, we challenge the swarm further by applying a periodic change in the density of the food items, hence the benefit of a quicker response becomes clearer. The periodic change is applied over periods of 2500 ts and can be of two types, either increasing or decreasing the number of food items N_{items} , always by a factor of 10.

In each of the two experimental settings, two interaction networks are implemented, proximity network (emerging from local interactions), and scale-free network (generated using preferential attachment). As mentioned above, for the construction of scale-free networks, the connected components of the robots resting at the nest site are used to impose the network topology. Over these networks the robots exchange specific information about their success or failure of the latest foraging attempt seeking an accurate estimation of the current situation in the foraging environment.

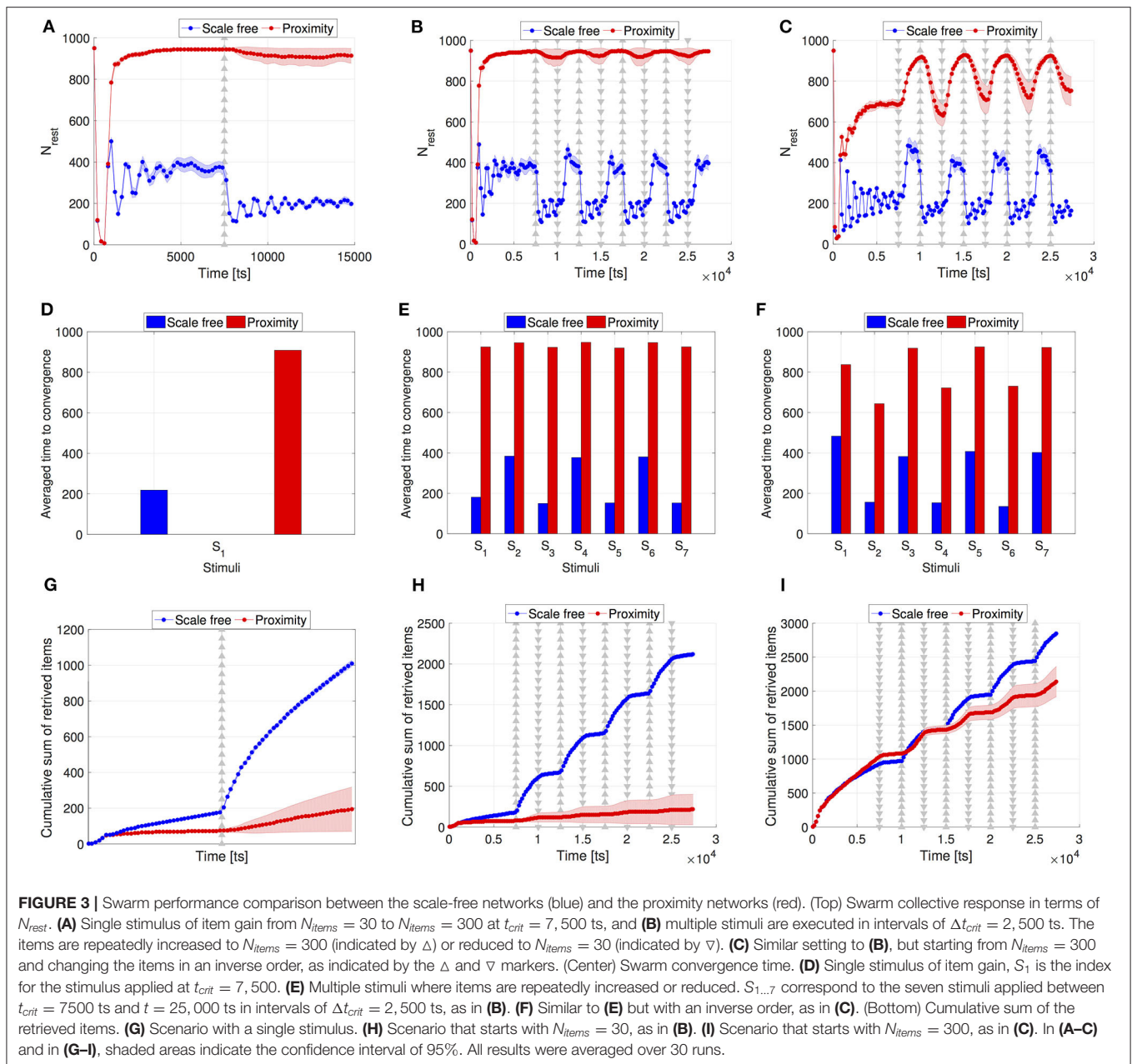
According to our experiments, there are two main cases, in which the influence of the communication topology is negligible. These are (i) small social cues (i.e., with s_f and s_i values smaller than 0.01), and (ii) small number of resting robots N_{rest} . The first case is straightforward, as the social cues decrease, the impact of the information obtained from other robots decreases, and hence the impact of the interaction network on the emergent dynamics vanishes. The second case is associated with the particular implementation of the scale-free communication network in the nest. Since the construction of this network relies on the connected components present in the nest at every time step, small numbers of resting robots result in scaling down the size of such connected components and hence topological contribution becomes negligible. Therefore, as we aim to investigate the influence of the interaction network on the emerging dynamics, we consider those cue configurations in which the social feedback of the robot's neighborhood has a distinguishable role in shaping its decision. This is achieved by setting the social cues to have a clear advantage over the individual cues—i.e., $s_f \gg i_f$, $s_r \gg i_r$. For an extensive discussion on the impact of cue values on swarm behavior in a similar settings of the foraging task the interested reader is referred to Liu et al. (2007) and Rausch et al. (2019a). For the reasons mentioned above, we set the cue values to $s_f = 0.25$, $s_r = 0.25$, $i_f = 0.01$, $i_r = 0.01$. Nevertheless, further below we will additionally compare our results to those obtained with more extreme values of the social cues, i.e., $s_f = 0.01$, $s_r = 0.01$ and $s_f = 0.99$, $s_r = 0.99$.

The plots in **Figure 3** depict results obtained over 30 runs. They compare the emergent collective response of the swarm

to a single stimulus (i.e., change in food density) as well as to multiple stimuli when individuals interact locally in comparison to interacting via scale-free topologies. Firstly, our results reveal a clear impact of the network structure on the robot activation level across all types of stimuli (i.e., increasing or decreasing food item density). This is illustrated through the number of resting robots being considerably smaller when using the scale-free network as opposed to the proximity network throughout the entire simulation time (see **Figures 3A,B**). Proximity networks in **Figure 3B** show a non-adaptive swarm behavior that is largely due to the very low number of foraging robots. When there are too few foraging robots, the system tends to approach a global absorbing state in which robots cease to switch to foraging. In case of proximity networks in **Figure 3B**, this tendency toward the global resting state is due to the initial low density in food items (i.e., $N_{items} = 30$). Low N_{items} leads to a large number of failed attempts to find and retrieve them. Consequently, this increases $P_{f \rightarrow r}$ up to its maximum $P_{f \rightarrow r} = 1$, pushing the robots to keep resting. Thus, the subsequent increase in items to $N_{items} = 300$ is not sensed by the swarm. As an example, this behavior is evident at $t = 7,500$ ts when N_{rest} did not decrease in response to the increasing N_{items} .

Therefore, it is important to consider the robustness of the swarm behavior to initial conditions, prior to the external stimulus. To this end, we inverted the changes of N_{items} , starting with $N_{items} = 300$, reducing it to $N_{items} = 30$ at $t = 7,500$ ts, then increasing it back to $N_{items} = 300$, etc... Under this specific setting, foraging robots have a higher likelihood to find items than when the initial item density is as low as $N_{items} = 30$. Consequently, the returning robots broadcast a larger number of “success” messages, increasing the robots’ probability to switch to foraging ($P_{r \rightarrow f}$). **Figure 3C** shows that this configuration of the initial conditions led to an adaptive swarm behavior for the case of proximity networks. This adaptive behavior comes with a reduced time to convergence (see **Figure 3F** vs. **Figure 3E**) and a significantly higher number of retrieved items (see **Figure 3I** vs. **Figure 3H**). Nevertheless, with scale-free networks the collective response not only remained more rapid but also appeared to be more robust to the initial conditions of the system, as the trajectory of N_{rest} in **Figure 3C** is qualitatively similar to **Figure 3B**. Nevertheless, the scale-free networks display higher fluctuations of N_{rest} compared to the relatively coherent decision achieved when using proximity networks (**Figures 3A–C**). This is due to the high impact that a single hub can have on a large population of the swarm.

The key contribution of the network topology is reflected in the time the swarm requires to build up its collective response. When using scale-free networks, hubs—i.e., robots with an exceptionally high connectivity degree—help accelerate the information propagation in two manners: (i) due to their high connectivity degree, their individual experience is shared with a large number of robots within one time step. (ii) Their presence creates a shorter average path of the network compared to proximity networks, which allows any two robots to exchange information over a smaller number of hops (i.e., within fewer time steps). As mentioned above, we use the settling time defined in Equation (4) to compute the swarm’s convergence time



after each stimuli—i.e., change in the items' density. **Figure 3D** shows the time it took the swarm to converge to a steady number of resting/foraging robots after increasing the items at the foraging area from $N_{items} = 30$ to $N_{items} = 300$ at time step $\Delta t_{crit} = 7,500$. **Figures 3E,F** show the same measure for the repeated stimuli of items increase and decrease, starting from $N_{items} = 30$ (**Figure 3E**) and $N_{items} = 300$ (**Figure 3F**). In all three findings, **Figures 3D–F**, we can notice the significantly shorter convergence time when robots in the nest are communicating using the scale-free network in comparison to the proximity network. These results suggest a higher level of swarm adaptivity to dynamic environments under scale-free communications. Furthermore, as shown in **Figures 3G–I**,

using scale-free networks the cumulative sum of the retrieved items is either considerably higher from the beginning or at the later stages of the experiment, compared to the scenarios with proximity networks.

An important aspect to notice is the physical division between the site at which the information is to harvest (i.e., the foraging environment), and the site at which the information is to exchange (i.e., the nest). Usually, the communication speed is considerably higher than motion speed. However, specifically in the foraging scenario, the communication speed is limited by the motion speed, since it is necessary for the robot to travel across the foraging environment to reach the nest, where it can start communicating. One of the clear consequences of this important

remark is that even for the case of scale-free networks where the collective response is accelerated, there is a considerably faster swarm reaction to an increase in the food density compared to the reaction to a decrease (see the blue line in **Figure 3B**). Before the increase of food items, there were few foraging robots. Those robots consumed time to return to the nest, switch to resting, inform their neighbors about their foraging experience, and, ultimately, convince more robots to leave the nest in case of a successful foraging attempt. For scale-free networks this resulted in a rapid activation of resting robots. Differently, collective reaction slowed down when the environmental change was a decrease in food items. This behavior can be explained as follows: the large number of robots foraging while the food density was high experienced the drop in the food density through their failed foraging attempts. Upon returning to the nest, these robots led to considerably higher crowding at the nest entrance. This prolonged the time that the robots needed to enter the nest and start communicating. Moreover, the higher N_{rest} the higher the likelihood that there is one, giant, connected component inside the nest, spanning over a large number of robots. If such a network is scale-free, the hubs have a high chance of influencing many robots to switch to foraging. By contrast, a low N_{rest} often led to fragmented networks, reducing the influence of hubs, lowering the number of switching robots and, thus, slowing down the collective response compared to a high N_{rest} . Hence, the collective response time—even when using scale-free networks—is longer when there are many robots foraging.

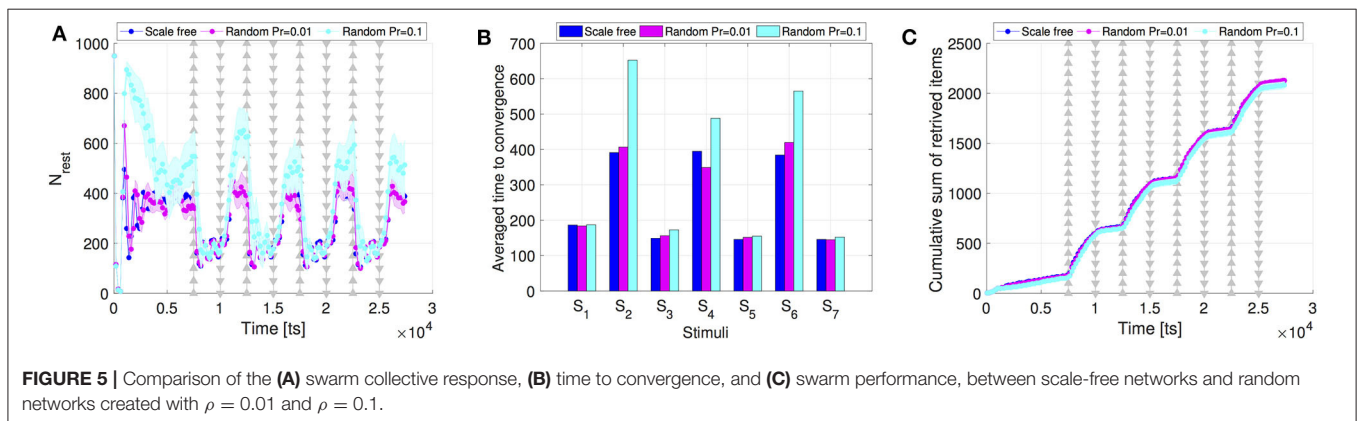
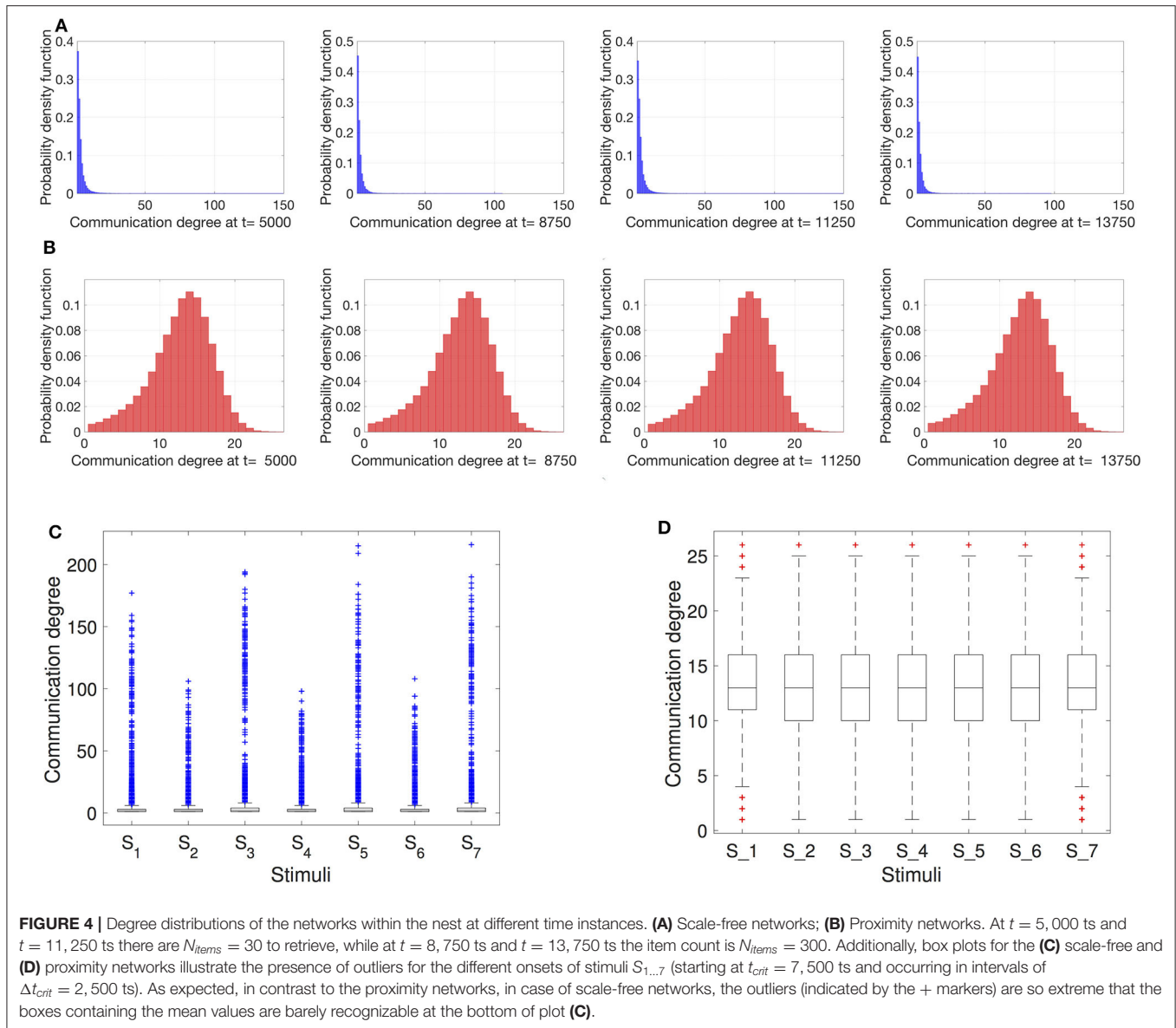
To obtain a closer look at the interaction network topology, we can analyze the degree distributions of the resting robots interacting inside nest. We draw the degree distributions for different time steps that are selected when the item density was both high (i.e., 300 items) and low (i.e., 30 items). As we can see in **Figure 4A**, scale-free networks strongly resemble a power-law distributed degree for all time steps at which the networks are recorded. Similar consistent is the degree distribution of the proximity networks in **Figure 4B** for all tested time steps. However, the degree distribution here appears closer to a Gaussian distribution which is more symmetrical around the mean than the scale-free network and has fewer outliers. To get a clearer look at the outliers, in **Figures 4C,D**, we show the communication degree using boxplots. For the scale-free networks the density of outliers is notably large, the most extreme among those are the hubs in the network. We can also notice a clear trend of a higher number of hubs when the number of resting robots N_{rest} is higher due to low N_{items} . This density of outliers changes periodically between the external stimuli S_i together with N_{rest} . In the case of proximity networks, the boxplots show a relatively low density of outliers and negligible changes with S_i .

Additionally, it is worthwhile considering the effect of rewiring on the collective response. As elaborated in section 2.2, Algorithm 1 is applied at every time step as the robots are in motion. However, because Algorithm 1 has a stochastic component, the resulting network at time step t is very likely to be different from $t - 1$. Such dynamic rewiring increases the probability that two remote robots share a link. Consequently,

a random robot is more likely to obtain information from spatially uncorrelated sources, i.e., it obtains a sample that is more representative of the swarm opinion. This resembles the common “random mixing” paradigm often found in swarm robotics, stating that an encounter probability between two robots is the same for any pair of robots. Thus, the adaptive behavior that follows from using Algorithm 1 could be largely attributed to this rewiring-induced opinion mixing.

To examine whether this may indeed be the case, we ran simulations with a modified version of Algorithm 1 where we replaced the preferential attachment component $d_s / \sum_i d_i$ by a real number $\rho \in \{0.01, 0.1\}$. Note that while this modification aims at altering the network topology, the resulting alternative networks are still regenerated at each time step, similar to scale-free networks, i.e., the notion of rewiring is preserved. The results are shown in **Figure 5**. The similarity to the scale-free networks scenario is particularly striking for $\rho = 0.01$. When N_{rest} is low, it becomes difficult to separate a scale-free network (where the degrees are power law distributed) from a small-world network (where the degree distribution is much less skewed, i.e., more symmetric around the mean value). Therefore, for low N_{rest} the impact of the preferential attachment component in Algorithm 1 can be well-approximated by a constant such as $\rho = 0.01$. More importantly, it shows that the strong effect that dynamic rewiring has on swarm adaptivity and collective response.

A feature that frequently occurs in realistic communication is the packet loss. It occurs when a robot fails to receive a message broadcast by a neighbor, due to radio-frequency interference or due to overflow of a robot’s receiver queue. We implemented packet loss events by allowing the robots to ignore incoming messages with probability p_{pl} . **Figure 6** shows the results for the proximity and scale-free networks with $p_{pl} \in \{0.1, 0.5\}$. Surprisingly, the swarm adaptivity considerably improves in case of proximity networks, while with scale-free networks the swarm remains more robust to the influence of the packet loss. Higher probabilities of packet loss appears to shorten the time to convergence and slightly increase the number of collected items. One possible explanation for this behavior could be that by probabilistically ignoring incoming messages the robots become to some extent able to reduce the correlation between their behavior and that of their spatial neighbors. Synthetically generated networks, such as the scale-free networks considered in this study, represent an extreme case of such spatial decorrelation. In contrast, in proximity networks and absence of packet loss, spatial correlations are very high, leading to feedback mechanisms that reduce sensitivity to new information. The presence of packet loss appears to create a middle ground that bolsters the adaptive behavior at the swarm level. However, we only tested two values of p_{pl} and it is possible that for $p_{pl} > 0.5$ inverse effects could be observed. Finally, when resting state can be associated with low energy consumption, the behavior of the system in the presence of here considered p_{pl} may demonstrate a high level of efficiency, in terms of increasing N_{rest} while preserving the high number of retrieved items. Nevertheless, as mentioned above, the detailed investigation of the influence of packet loss is beyond the scope of the current study and future research is needed to confirm the generality of our



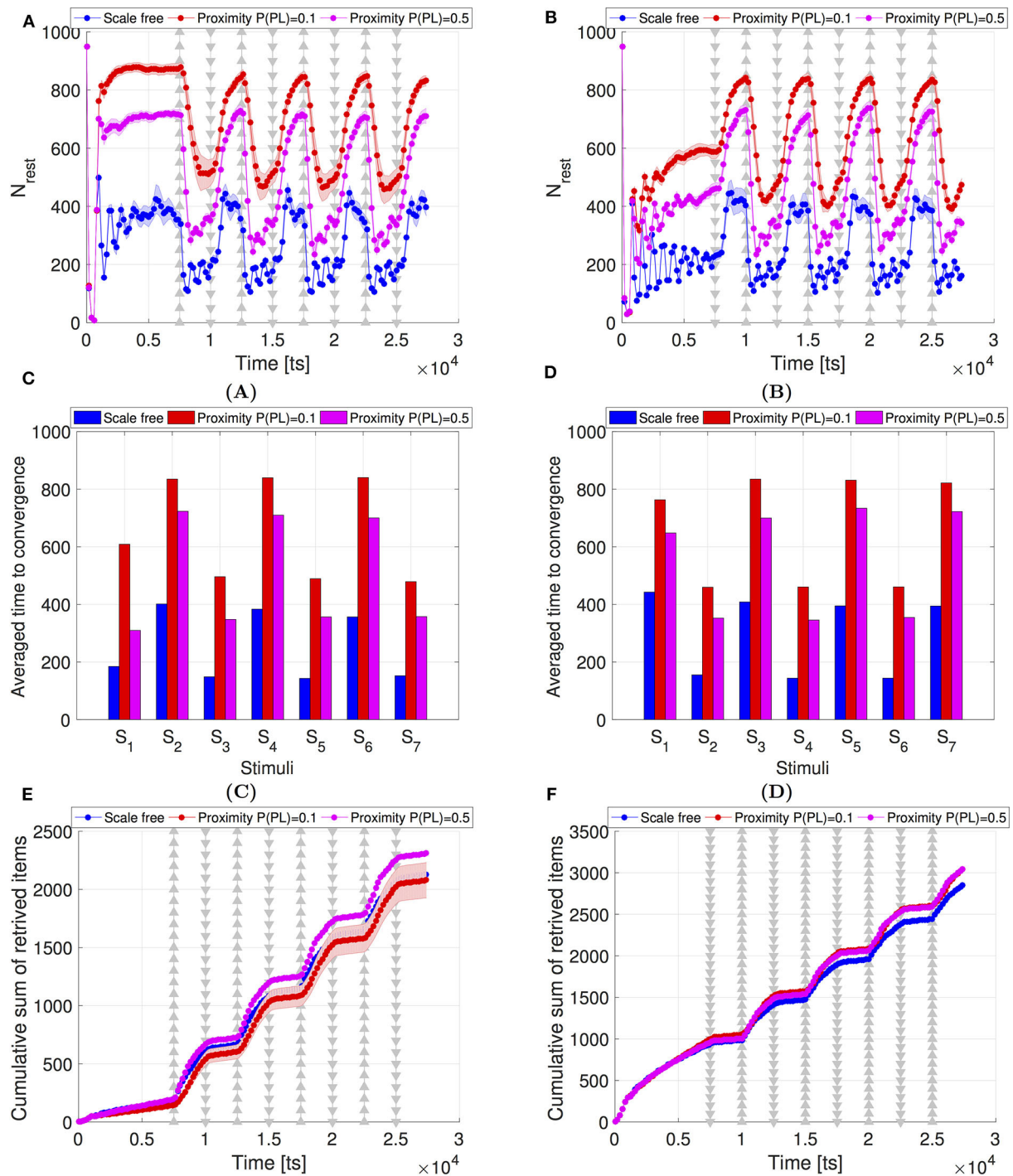
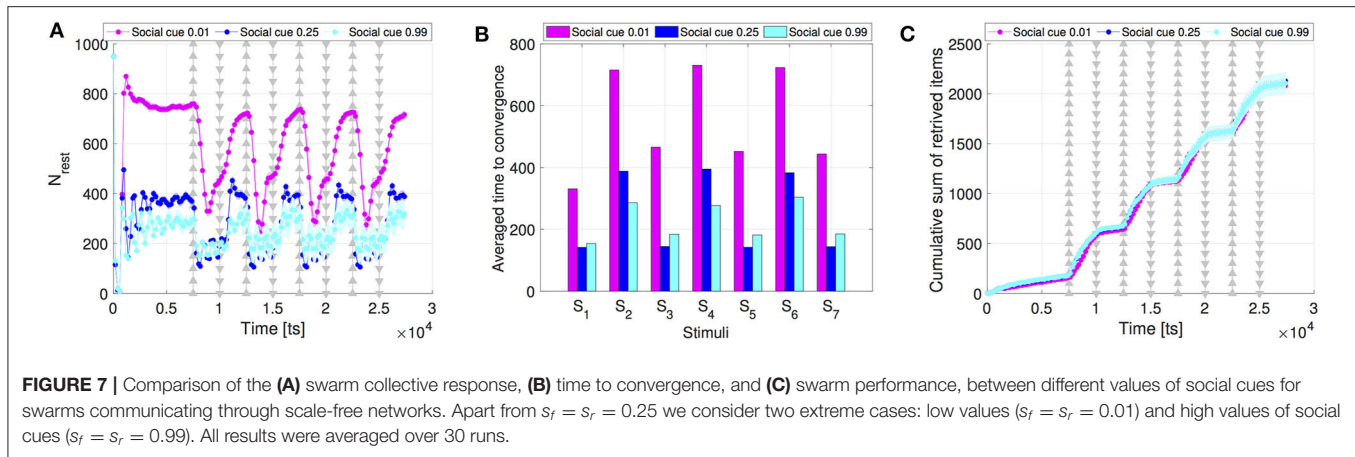


FIGURE 6 | Swarm performance comparison between the scale-free networks (blue) and the proximity networks in presence of packet loss, with packet loss probability $p_{pl} = 0.1$ (red) and $p_{pl} = 0.5$ (magenta). The number of items is repeatedly increased to $N_{items} = 300$ (indicated by Δ) or reduced to $N_{items} = 30$ (indicated by ∇). These repeating changes occur in intervals of $\Delta t_{crit} = 2,500$ ts, starting at $t_{crit} = 7,500$ ts. (Left) Scenario with initially $N_{items} = 300$; (A,B) Swarm collective response in terms of N_{rest} . (C,D) Swarm convergence time. $S_{1...7}$ correspond to the seven stimuli between $t_{crit} = 7,500$ ts and $t = 25,000$ ts. (E,F) Cumulative sum of the retrieved items. In (A,B,E,F), the shaded areas indicate the confidence interval of 95%. All results represent averages over 30 runs.



findings⁵. Moreover, here we consider constant values of p_{pl} that are the same for every robot in the swarm and that do not change based on the location of the robot or the number of communication links. In contrast, in more realistic settings not only the packet loss but also p_{pl} itself may have fluctuating values depending on the situation and both could be profoundly difficult to control.

Finally, we compare the intensity of the collective response resulting from different social cues. As mentioned above, social cues are the main driver of the dynamics to build up a faster response over the interaction network. Our results show that higher social cues lead to a higher activation of the resting robots, see **Figure 7** that shows the activation of the resting robots when setting $s_f = 0.99, s_r = 0.99$ in comparison to the setting $s_f = 0.01, s_r = 0.01$ (results are averaged over 30 runs). High social cues activate considerably more resting robots (i.e., reduces number of resting robots) than low cue values (**Figure 7A**). However, the convergence time with high cue values is comparable to the previously considered default case of $s_f = s_r = 0.25$ (see **Figure 7B**). The number of collected items overlaps for all three cue values (see **Figure 7C**).

4. CONCLUSION

The goal of this study is to investigate the role of network topology in influencing the propagation of information in a foraging scenario with changing the availability of food items. Therefore, we have addressed scenarios with dynamic environments, a realistic aspect of most real-world applications. We considered two types of changes: a single abrupt change (referred to a single stimulus) and periodic changes (multiple stimuli). We aimed to examine how scale-free networks, in particular, may accelerate the spreading of information and hence enable a quicker collective response than proximity networks to the global changes.

We have implemented scale-free networks across the robots resting in the nest, as the nest is usually the part of the environment in which communication takes place. We applied the well-known preferential attachment technique to construct the scale-free topology. Following preferential attachment, the probability of connecting to a robot is proportional to its current connectivity degree. Therefore, a number of robots emerge to have a relatively high degree of connectivity, those are referred to as the hub robots. When the density of food items changes at the foraging environment, and this change is reflected in the robots' experience, scale-free networks enable a faster spreading of this information in the nest. This led to a faster collective response compared to the scenarios in which interactions between the resting robots were implemented using proximity networks.

Our results suggest that the use of scale-free networks can improve the collective response of the swarm to changes in their dynamic environment, by improving the spread of shared information and reducing the spatial correlation in the robots' decisions. These two desired features in collective systems are achieved due to the introduced possibility to communicate over long distances, as well as due to the dynamic rewiring of the interaction network at every time step as a consequence of robot motion. These insights were obtained by comparing the swarm behavior in scenarios with and without systematic packet loss, in addition to comparing the swarm performance between scenarios with scale-free networks and with alternative random networks. Furthermore, our findings showcase the effect of social cues on the intensity of the collective response in presence of scale-free networks. Our results show that higher social cues lead to a higher activation of the resting robots, due to the increased influence of their neighbors' experience.

Although scale-free networks have shown to equip the swarm with a quicker reaction to changes in dynamic environments—studied for the collective foraging task—this came at the cost of the coherence of the collective response. Scale-free topologies led to more fluctuations of the swarm decision (whether to rest or to forage). These fluctuations can be explained in terms of the high influence of particular individuals (i.e., the hubs) on the opinions of a large population of the resting

⁵To this end, the interested reader is encouraged to use our publicly available resources provided on <https://osf.io/48b9h/>.

robots. Two particularly promising research directions for future work include the design of self-organized algorithms to implement scale-free topologies in robots swarms. Additionally, the design of efficient individual decision mechanisms that helps the collective response to demonstrate a higher stability. Finally, generalizing this study to other collective tasks such as site selection, flocking, and others may also lead to new interesting insights.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are publicly available. This data can be found here: <https://osf.io/48b9h/>.

REFERENCES

- Albert, R., and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Rev. Modern Phys.* 74, 47–97. doi: 10.1103/RevModPhys.74.47
- Albert, R., Jeong, H., and Barabási, A.-L. (2000). Error and attack tolerance of complex networks. *Nature* 406:378. doi: 10.1038/35019019
- Alstott, J., Bullmore, E., and Plenz, D. (2014). powerlaw: A python package for analysis of heavy-tailed distributions. *PLoS ONE* 9:e85777. doi: 10.1371/journal.pone.0085777
- Bayındır, L. (2016). A review of swarm robotics tasks. *Neurocomputing* 172, 292–321. doi: 10.1016/j.neucom.2015.05.116
- Bonani, M., Longchamp, V., Magnenat, S., Rétornaz, P., Burnier, D., Roulet, G., et al. (2010). "The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research," in 2010 *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Taipei), 4187–4193. doi: 10.1109/IROS.2010.5649153
- Broido, A. D., and Clauset, A. (2019). Scale-free networks are rare. *Nat. Commun.* 10, 1–10. doi: 10.1038/s41467-019-08746-5
- Campo, A., and Dorigo, M. (2007). "Efficient multi-foraging in swarm robotics," in *European Conference on Artificial Life* (Berlin; Heidelberg: Springer), 696–705. doi: 10.1007/978-3-540-74913-4_70
- Cavagna, A., Cimarelli, A., Giardina, I., Parisi, G., Santagati, R., Stefanini, F., et al. (2010). Scale-free correlations in starling flocks. *Proc. Natl. Acad. Sci. U.S.A.* 107, 11865–11870. doi: 10.1073/pnas.1005766107
- Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Rev.* 51, 661–703. doi: 10.1137/070710111
- Cohen, R., and Havlin, S. (2003). Scale-free networks are ultrasmall. *Phys. Rev. Lett.* 90:058701. doi: 10.1103/PhysRevLett.90.058701
- Conradt, L. (2011). Models in animal collective decision-making: information uncertainty and conflicting preferences. *Interface Focus* 2, 226–240. doi: 10.1098/rsfs.2011.0090
- Goh, K.-I., Kahng, B., and Kim, D. (2001). Universal behavior of load distribution in scale-free networks. *Phys. Rev. Lett.* 87:278701. doi: 10.1103/PhysRevLett.87.278701
- Hoff, N., Wood, R., and Nagpal, R. (2013). "Distributed colony-level algorithm switching for robot swarm foraging," in *Distributed Autonomous Robotic Systems*, eds A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M. Ani Hsieh, et al. (Berlin; Heidelberg: Springer), 417–430. doi: 10.1007/978-3-642-32723-0_30
- Holme, P. (2019). Rare and everywhere: perspectives on scale-free networks. *Nat. Commun.* 10, 1–3. doi: 10.1038/s41467-019-09038-8
- Jiang, L., Jin, X., Xia, Y., Ouyang, B., Wu, D., and Chen, X. (2014). A scale-free topology construction model for wireless sensor networks. *Int. J. Distrib. Sensor Netw.* 10:764698. doi: 10.1155/2014/764698
- Khaluf, Y., Birattari, M., and Rammig, F. (2016). Analysis of long-term swarm performance based on short-term experiments. *Soft Comput.* 20, 37–48. doi: 10.1007/s00500-015-1958-0
- Khaluf, Y., and Dorigo, M. (2016). Modeling robot swarms using integrals of birth-death processes. *ACM Trans. Auton. Adapt. Syst.* 11:8. doi: 10.1145/2870637
- Khaluf, Y., Ferrante, E., Simoens, P., and Huepe, C. (2017a). Scale invariance in natural and artificial collective systems: a review. *J. R. Soc. Interface* 14:20170662. doi: 10.1098/rsif.2017.0662
- Khaluf, Y., Pinciroli, C., Valentini, G., and Hamann, H. (2017b). The impact of agent density on scalability in collective systems: noise-induced versus majority-based bistability. *Swarm Intell.* 11, 155–179. doi: 10.1007/s11721-017-0137-6
- Khaluf, Y., Rausch, I., and Simoens, P. (2018). "The impact of interaction models on the coherence of collective decision-making: a case study with simulated locusts," in *International Conference on Swarm Intelligence* (Rome: Springer), 252–263. doi: 10.1007/978-3-030-00533-7_20
- Khaluf, Y., Simoens, P., and Hamann, H. (2019a). The neglected pieces of designing collective decision-making processes. *Front. Robot. AI* 6:16. doi: 10.3389/frobt.2019.00016
- Khaluf, Y., Vanhee, S., and Simoens, P. (2019b). Local ant system for allocating robot swarms to time-constrained tasks. *J. Comput. Sci.* 31, 33–44. doi: 10.1016/j.jocs.2018.12.012
- Lerman, K., and Galstyan, A. (2002). Mathematical model of foraging in a group of robots: effect of interference. *Auton. Robots* 13, 127–141. doi: 10.1023/A:1019633424543
- Li, M., Lu, K., Zhu, H., Chen, M., Mao, S., and Prabhakaran, B. (2008). "Robot swarm communication networks: architectures, protocols, and applications," in 2008 *Third International Conference on Communications and Networking in China* (Hangzhou), 162–166.
- Li, X., and Chen, G. (2003). A local-world evolving network model. *Phys. A* 328, 274–286. doi: 10.1016/S0378-4371(03)00604-6
- Liu, W., Winfield, A. F. T., Sa, J., Chen, J., and Dou, L. (2007). Towards energy optimization: emergent task allocation in a swarm of foraging robots. *Adapt. Behav.* 15, 289–305. doi: 10.1177/1059712307082088
- Michelena, P., Jeanson, R., Deneubourg, J.-L., and Sibbald, A. M. (2009). Personality and collective decision-making in foraging herbivores. *Proc. R. Soc. B Biol. Sci.* 277, 1093–1099. doi: 10.1098/rspb.2009.1926
- Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., et al. (2012). Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell.* 6, 271–295. doi: 10.1007/s11721-012-0072-5
- Pini, G., Brutschy, A., Pinciroli, C., Dorigo, M., and Birattari, M. (2013). Autonomous task partitioning in robot foraging: an approach based on cost estimation. *Adapt. Behav.* 21, 118–136. doi: 10.1177/1059712313484771
- Pinter-Wollman, N., Bala, A., Merrell, A., Queirolo, J., Stumpe, M., Holmes, S., et al. (2013). Harvester ants use interactions to regulate forager activation and availability. *Anim. Behav.* 86, 197–207. doi: 10.1016/j.anbehav.2013.05.012
- Rausch, I., Khaluf, Y., and Simoens, P. (2020). *Adaptive Foraging in Dynamic Environments Using Scale-Free Interaction Networks*. Retrieved from: <https://osf.io/48b9h/>.
- Rausch, I., Khaluf, Y., and Simoens, P. (2019a). Scale-free features in collective robot foraging. *Appl. Sci.* 9:2667. doi: 10.3390/app9132667
- Rausch, I., Reina, A., Simoens, P., and Khaluf, Y. (2019b). Coherent collective behaviour emerging from decentralised balancing of social

AUTHOR CONTRIBUTIONS

All authors contributed conception and design of the study. PS provided funding and administered the project. YK and PS supervised the study. IR implemented the simulation setup, performed the simulations, and the formal data analysis. YK wrote the first draft of the manuscript. YK and IR continuously discussed and elaborated on the design parameters and obtained results, which led in some cases to further implementations and analysis. YK and IR wrote most sections of the manuscript and analyzed the experimental data. IR and YK critically reviewed and edited the manuscript. All authors approved the submitted version.

- feedback and noise. *Swarm Intell.* 13, 321–345. doi: 10.1007/s11721-019-00173-y
- Reina, A., Cope, A. J., Nikolaidis, E., Marshall, J. A., and Sabo, C. (2017). Ark: augmented reality for kilobots. *IEEE Robot. Autom. Lett.* 2, 1755–1761. doi: 10.1109/LRA.2017.2700059
- Reina, A., Valentini, G., Fernández-Oto, C., Dorigo, M., and Trianni, V. (2015). A design pattern for decentralised decision making. *PLoS ONE* 10:e140950. doi: 10.1371/journal.pone.0140950
- Richardson, T. O., Mullon, C., Marshall, J. A., Franks, N. R., and Schlegel, T. (2018). The influence of the few: a stable-oligarchy-controls information flow in house-hunting ants. *Proc. R. Soc. B Biol. Sci.* 285:20172726. doi: 10.1098/rspb.2017.2726
- Schafer, R. J., Holmes, S., and Gordon, D. M. (2006). Forager activation and food availability in harvester ants. *Anim. Behav.* 71, 815–822. doi: 10.1016/j.anbehav.2005.05.024
- Seeley, T. D., Visscher, P. K., Schlegel, T., Hogan, P. M., Franks, N. R., and Marshall, J. A. R. (2012). Stop signals provide cross inhibition in collective decision-making by honeybee swarms. *Science* 335, 108–111. doi: 10.1126/science.1210361
- Thivierge, J.-P. (2014). Scale-free and economical features of functional connectivity in neuronal networks. *Phys. Rev. E* 90:022721. doi: 10.1103/PhysRevE.90.022721
- Valentini, G. (2017). *Achieving Consensus in Robot Swarms*, Vol. 706 of *Studies in Computational Intelligence*. Cham: Springer. doi: 10.1007/978-3-319-53609-5
- Valentini, G., Ferrante, E., Hamann, H., and Dorigo, M. (2016). Collective decision with 100kilobots: speed versus accuracy in binary discrimination problems. *Auton. Agents Multi-Agent Syst.* 30, 553–580. doi: 10.1007/s10458-015-9323-3

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Rausch, Simoens and Khaluf. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Advantages of publishing in Frontiers



OPEN ACCESS

Articles are free to read
for greatest visibility
and readership



FAST PUBLICATION

Around 90 days
from submission
to decision



HIGH QUALITY PEER-REVIEW

Rigorous, collaborative,
and constructive
peer-review



TRANSPARENT PEER-REVIEW

Editors and reviewers
acknowledged by name
on published articles

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org

Contact us: info@frontiersin.org | +41 21 510 17 00



REPRODUCIBILITY OF RESEARCH

Support open data
and methods to enhance
research reproducibility



DIGITAL PUBLISHING

Articles designed
for optimal readership
across devices



FOLLOW US

@frontiersin



IMPACT METRICS

Advanced article metrics
track visibility across
digital media



EXTENSIVE PROMOTION

Marketing
and promotion
of impactful research



LOOP RESEARCH NETWORK

Our network
increases your
article's readership