

Advanced motion control and navigation of robots in extreme environments

Edited by

Allahyar Montazeri, Nargess Sadeghzadeh-Nokhodberiz,
Khoshnam Shojaei and Kaspar Althoefer

Published in

Frontiers in Robotics and AI



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 978-2-8325-5690-0
DOI 10.3389/978-2-8325-5690-0

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: frontiersin.org/about/contact

Advanced motion control and navigation of robots in extreme environments

Topic editors

Allahyar Montazeri — Lancaster University, United Kingdom

Nargess Sadeghzadeh-Nokhodberiz — Qom University of Technology, Iran

Khoshnam Shojaei — Digital Processing and Machine Vision Research Center, Najafabad Branch, Islamic Azad University; Department of Electrical Engineering, Najafabad Branch, Islamic Azad University, Iran

Kaspar Althoefer — Queen Mary University of London, United Kingdom

Citation

Montazeri, A., Sadeghzadeh-Nokhodberiz, N., Shojaei, K., Althoefer, K., eds. (2024).

Advanced motion control and navigation of robots in extreme environments.

Lausanne: Frontiers Media SA. doi: 10.3389/978-2-8325-5690-0

Table of contents

- 04 **Editorial: Advanced motion control and navigation of robots in extreme environments**
Allahyar Montazeri, Nargess Sadeghzadeh-Nokhodberiz, Khoshnam Shojaei and Kaspar Althoefer
- 07 **Path planning with the derivative of heuristic angle based on the GBFS algorithm**
Daehee Lim and Jungwook Jo
- 21 **Treatise on Analytic Nonlinear Optimal Guidance and Control Amplification of Strictly Analytic (Non-Numerical) Methods**
Timothy Sands
- 37 **A neural flexible PID controller for task-space control of robotic manipulators**
Nguyen Tran Minh Nguyet and Dang Xuan Ba
- 47 **A method to benchmark the balance resilience of robots**
Simone Monteleone, Francesca Negrello, Giorgio Grioli, Manuel G. Catalano, Antonio Bicchi and Manolo Garabini
- 67 **Vision-based particle filtering for quad-copter attitude estimation using multirate delayed measurements**
Nargess Sadeghzadeh-Nokhodberiz, Mohammad Iranshahi and Allahyar Montazeri
- 78 **Towards reuse and recycling of lithium-ion batteries: tele-robotics for disassembly of electric vehicle batteries**
Jamie Hathaway, Abdelaziz Shaarawy, Cansu Akdeniz, Ali Aflakian, Rustam Stolkin and Alireza Rastegarpanah
- 93 **Simultaneous localization and mapping in a multi-robot system in a dynamic environment with unknown initial correspondence**
Hadiseh Malakouti-Khah, Nargess Sadeghzadeh-Nokhodberiz and Allahyar Montazeri
- 110 **A survey on autonomous environmental monitoring approaches: towards unifying active sensing and reinforcement learning**
David Mansfield and Allahyar Montazeri
- 130 **Distributed safe formation tracking control of multiquadcopter systems using barrier Lyapunov function**
Nargess Sadeghzadeh-Nokhodberiz, Mohammad Reza Sadeghi, Rohollah Barzamini and Allahyar Montazeri



OPEN ACCESS

EDITED AND REVIEWED BY
Kostas J. Kyriakopoulos,
National Technical University of
Athens, Greece

*CORRESPONDENCE
Allahyar Montazeri,
✉ a.montazeri@lancaster.ac.uk

RECEIVED 12 October 2024
ACCEPTED 24 October 2024
PUBLISHED 15 November 2024

CITATION
Montazeri A, Sadeghzadeh-Nokhodberiz N,
Shojaei K and Althoefer K (2024) Editorial:
Advanced motion control and navigation of
robots in extreme environments.
Front. Robot. AI 11:1510013.
doi: 10.3389/frobt.2024.1510013

COPYRIGHT
© 2024 Montazeri,
Sadeghzadeh-Nokhodberiz, Shojaei and
Althoefer. 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.

Editorial: Advanced motion control and navigation of robots in extreme environments

Allahyar Montazeri^{1*}, Nargess Sadeghzadeh-Nokhodberiz²,
Khoshnam Shojaei³ and Kaspar Althoefer⁴

¹School of Engineering, Lancaster University, Lancaster, United Kingdom, ²Department of Electrical Engineering, Qom University of Technology, Qom, Iran, ³Department of Electrical Engineering, Islamic Azad University Najafabad, Najafabad, Iran, ⁴School of Engineering and Materials Science, Queen Mary University of London, London, United Kingdom

KEYWORDS

uncertainties, motion control, extreme environmnet, unstructured environmnet, robust and adaptive control

Editorial on the Research Topic

[Advanced motion control and navigation of robots in extreme environments](#)

The use of robotics and artificial intelligence in environments which are dangerous, demanding, and dull is one of the key research areas developed recently to address the industrial need on reducing the human burden in hazardous workplace and create a safer environment for humans by taking them out of extreme environments. This may span various unstructured and cluttered environments that are difficult or dangerous for humans to access, including monitoring and sampling of deep sea, glaciers, and frozen ocean, volcanology, underground mining, space exploration, and nuclear production and decommissioning. Robotics research into challenging and cluttered environment covers Research Topic such as advanced mobility, navigation, and mapping as well as advanced motion control, manipulation, and teleoperation. Environmental characterization includes research into machine vision, remote sensing, and deep reinforcement learning.

This research need is covered in this Research Topic through nine contributions co-authored by researchers from three continents (Asia, Europe and North America). This confirms the importance of the research and development in this area across various countries and an ongoing demand for more investment.

The content of this Research Topic has been organized as follows. The first paper [Mansfield and Montazeri](#) is a review paper discusses the application of reinforcement learning (RL) in active environmental monitoring (EM) systems. The need for reliable and intelligent monitoring solutions to address environmental pollution and climate change is highlighted, with a focus on the use of RL to train agents for adaptive and robust sensing in dynamic and extreme environments. The paper proposes a framework that formulates active sensing as an RL problem, unifying various EM tasks such as coverage, patrolling, source seeking, and exploration. Despite the potential of RL for EM applications, practical implementation and

research in multi-agent systems are lacking, with most work remaining in the simulation phase.

The next five papers address the navigation problems in unstructured and extreme environments. [Sadeghzadeh-Nokhodbderiz et al.](#) study the problem of attitude estimation of a quad-copter system when the quad is equipped with camera and gyroscope sensors in which cameras, usually suffer from a slow sampling rate and processing time delay compared to inertial sensors, such as gyroscopes. Toward this, a sampling importance resampling (SIR) particle filter (PF) is extended using a discretized attitude kinematics in Euler angles and the processing images captured by the camera using the ORB feature extraction method and the homography method in Python-OpenCV. Experimental results are provided for a DJI Tello type quadcopter to demonstrate the performance of the proposed method.

[Malakouti-Khah et al.](#) solves the problem of simultaneously localization and mapping (SLAM) for a multi-robot system in a dynamic environment. The use of several robots in large, complex, and dynamic environments can significantly improve performance on the localization and mapping task, which has attracted many researchers to this problem more recently. Toward this, a modified Fast-SLAM method is proposed by implementing SLAM in a decentralized manner by considering moving landmarks in the environment. Due to the unknown initial correspondence of the robots, a geographical approach is embedded in the proposed algorithm to align and merge their maps. Data association is also embedded in the algorithm; this is performed using the measurement predictions in the SLAM process of each robot.

The study conducted by [Lim and Jo](#) introduces WA^*DH+ , an improved version of WA^*DH for path planning and navigation of robots in the extreme environments. WA^*DH struggles to find suboptimal nodes due to its filtering method, so the study inflated the suboptimality of the initial solution. WA^*DH+ uses the GBFS algorithm with an infinitely bounded suboptimal solution, resulting in faster solution returns than WA^*DH .

The work in [Sadeghzadeh-Nokhodbderiz et al.](#), however, addresses the inter-agent collision avoidance problem for a group of quadcopters cooperate each other for a totally distributed collision-free formation tracking control using Barrier Lyapunov function (BLF). The problem is formulated in a backstepping setting where both tracking and inter-agent collision avoidance are obtained through a predefined accuracy due to the use of BLFs. Virtual control inputs are considered for the translational (x and y -axes) subsystems that are then used to generate the desired values for the roll and pitch angles for the attitude control subsystem to solve the underactuated nature of the system leading to a hierarchical controller structure for each quadcopter. Finally, the attitude controller is designed for each quadcopter locally by taking into account a predetermined error limit by another BLF. Simulation results demonstrate the performance of the proposed approach.

Nevertheless, the fifth paper on navigation published by [Sands](#) has incorporated optimality criteria in problem formulation. Optimization techniques are useful for autonomous navigation but face challenges like noisy multi-sensor technologies and computational burdens. This study aims to highlight the efficacy and limitations of common methods and proposes more, applying them to full, nonlinear, coupled equations of motion. Five different types of optimum guidance and control algorithms are presented

and compared to a classical benchmark. Real-time optimization with singular switching and nonlinear transport theorem decoupling is introduced, showing superior performance in tracking errors, fuel usage, and computational burden.

The investigation by [Hathaway et al.](#) addresses the need of teleoperation in challenging environments. The use of telerobotics for semi-autonomous robotic disassembly of electric vehicle batteries is studied in this work. It compares a traditional haptic-cobot framework with identical cobots, revealing a time reduction of 22%–57%. However, this improvement is mainly due to expanded workspace and 1:1 positional mapping, and a 10%–30% reduction in first attempt success rate. The study also highlights the importance of realism in directional information for unbolting and grasping tasks.

The last paper is dealing with designing advanced motion controllers for robotics applications in front of external disturbances and uncertainties. [Nguyet and Ba](#) introduces the task-space position-tracking control of robotic manipulators using an adaptive robust Jacobian-based controller. The controller's structure is based on the conventional Proportional-Integral-Derivative (PID) paradigm. To compensate for both internal and external disturbances in the robot dynamics, an additional neural control signal is then synthesized under a non-linear learning law. Then, a novel gain learning feature is included to automatically change the PID gains for different operating situations, providing the high robustness of such a controller. Lyapunov constraints ensure the closed-loop system's stability. Results from extensive simulations are used to rigorously verify the suggested controller's effectiveness.

Author contributions

AM: Project administration, Supervision, Writing–original draft, Writing–review and editing. NS-N: Writing–original draft, Writing–review and editing. KS: Writing–original draft, Writing–review and editing. KA: Writing–original draft, Writing–review and editing.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

Acknowledgments

The authors would acknowledge using grammarly free version for proofreading and summarising some texts in this editorial.

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 author(s) declared that they were an editorial board member of *Frontiers*, at the time of submission. This had no impact on the peer review process and the final decision.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated

organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.



OPEN ACCESS

EDITED BY
Allahyar Montazeri,
Lancaster University, United Kingdom

REVIEWED BY
Khoshnam Shojaei,
Islamic Azad University of
Najafabad, Iran
Meysam Yadegar,
Qom University of Technology, Iran
Nargess Sadeghzadeh-Nokhodberiz,
Qom University of Technology, Iran

*CORRESPONDENCE
Daehee Lim,
dhlim@vessel21.com

SPECIALTY SECTION
This article was submitted to Robotic
Control Systems,
a section of the journal
Frontiers in Robotics and AI

RECEIVED 01 June 2022
ACCEPTED 11 July 2022
PUBLISHED 19 August 2022

CITATION
Lim D and Jo J (2022), Path planning
with the derivative of heuristic angle
based on the GBFS algorithm.
Front. Robot. AI 9:958930.
doi: 10.3389/frobt.2022.958930

COPYRIGHT
© 2022 Lim and Jo. 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.

Path planning with the derivative of heuristic angle based on the GBFS algorithm

Daehee Lim* and Jungwook Jo

R&D 2 Team, Vessel Aerospace Co., Ltd., Suwon-si, South Korea

Robots used in extreme environments need a high reactivity on their scene. For fast response, they need the ability to find the optimal path in a short time. In order to achieve this goal, this study introduces WA*DH+, an improved version of WA*DH (weighted A* with the derivative of heuristic angle). In some path planning scenes, WA*DH cannot find suboptimal nodes with the small inflation factor called the critical value due to its filtering method. It is hard to develop a new filtering method, so this study inflated the suboptimality of the initial solution instead. Critical values vary in every path planning scene, so increasing the inflation factor for the initial solution will not be the solution to our problem. Thus, WA*DH + uses the GBFS algorithm with the infinitely bounded suboptimal solution for its initial solution. Simulation results demonstrate that WA*DH + can return a better solution faster than WA*DH by finding suboptimal nodes in the given environment.

KEYWORDS

greedy best first search, heuristic angle, heuristic search, node-based algorithm, path planning, trajectory planning

Introduction

Path planning is an essential part of self-moving machines such as self-driving cars, unmanned aerial vehicles, or other robotics systems (De Momi, Elena et al., 2016; Boulares and Barnawi, 2021; Mac, Thi Thoa et al., 2016; Fu, Bing et al., 2018; Duan, Chao, et al., 2020; Liu, Zhiyuan et al., 2020; Hou, Mengxue et al., 2021). Path planning aims to find the path that has the lowest path cost in the shortest time because self-moving machines need fast reactions on their scene. Here, the path cost means the distance from the start to the target. Many path planning algorithms were developed to achieve this goal. Bioinspired algorithms such as genetic algorithm (GA), particle swarm optimization (PSO) (Das et al., 2016; Song et al., 2016), sampling-based algorithms such as rapidly exploring random tree (RRT), Voronoi, and artificial potential field (APF) algorithms are the examples (Yang, Liang et al., 2014; Yang, Liang et al., 2016). However, these algorithms sometimes show poor performances due to some limitations, such as the local minimum situation.

In order to avoid these threats, the A* algorithm (Hart et al., 1968) motivated by Dijkstra's algorithm and other node-based algorithms were developed. The main characteristic of A* and all variations of A* is the heuristic. The heuristic means the

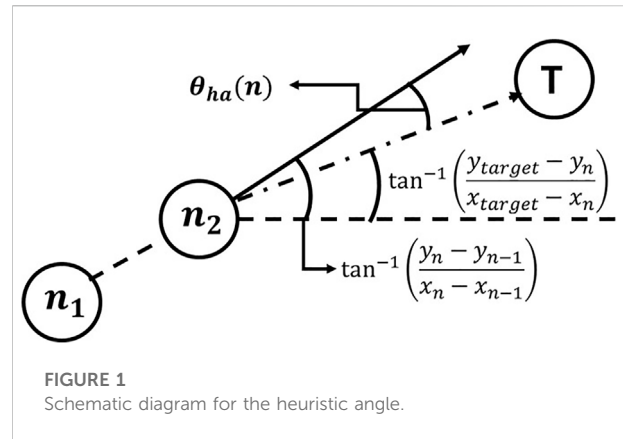
estimated distance from the current node to the target. The heuristic is the most important factor in A* and all variations of A* because the heuristic can change the performance of algorithms (Jing and Yang, 2018). The concept of the heuristic is used not only for node-based algorithms but also for other algorithms, such as the ACO algorithm (Dai, Xiaolin et al., 2019).

The heuristic is a powerful method for finding the optimal path. However, algorithms using the heuristic have a time-consuming problem. This problem made A* hard to use in real-time systems, so many researchers developed various methods to get a result of heuristic-using algorithms in a short time. With these trials, weighted A*(WA*), the bounded suboptimal search algorithm, was developed (Pohl, 1970). WA* uses the heuristic multiplied by the inflation factor ($\epsilon > 1$). The concept of inflating the heuristic solved a time-consuming problem. However, the inflated heuristic cannot guarantee the optimality of the result anymore.

Many researchers focused on this side effect, and as a result, many variations of WA* were developed. For example, dynamically weighted A*(DWA*) uses the $d(\text{root})$ as a depth bound (Pohl, 1973) and A_e^* uses the desired suboptimality bound to build the focal list from a node selected to expand (Pearl and Kim, 1982; Thayer and Ruml, 2008; Thayer and Ruml, 2009). Also, Aine Sandip et al. (2016) suggested two versions of multi-heuristic A*(MHA*), which uses multiple heuristic functions to find the path; independent MHA*(IMHA*), which uses independent g and h values for each search, and shared MHA*(SMHA*), which uses different g but a single g value for all the searches. Ying et al. (2018) suggested the evolutionary heuristic A*(EHA*), which uses GA to automatically design, calibrate, and optimize multi-weighted heuristic functions to maximize the performance of the algorithm (Yiu et al., 2018).

Anytime algorithms were also used in various path planning environments. Anytime algorithms have a flexible time cost and can return a sub-optimal solution in a short time and progressively optimize it till the time limit expires. The naïve anytime A*(ATA*) returns its result by iteratively reducing the inflation factor; however, it repeats previous works (Zhou and Hansen, 2002). To address this inefficient work, anytime repairing A*(ARA*) reuses the previous work to optimize the path efficiently (Likhachev et al., 2003; Li et al., 2012). The concept of reusing the previous work was adapted to D*. Thus, anytime dynamic A* was developed (Likhachev, Maxim et al., 2005). However, they need an understanding of complex mathematical logic, which can make users reluctant to use these algorithms.

Recently, weighted A* with the derivative of the heuristic angle (WA*DH), motivated by the concept of anytime algorithms, was suggested (Lim et al., 2020). WA*DH returns its path by getting an initial solution with a certain inflation factor and partially replans the path with the same inflation factor used in the initial solution. Because WA*DH improves the initial



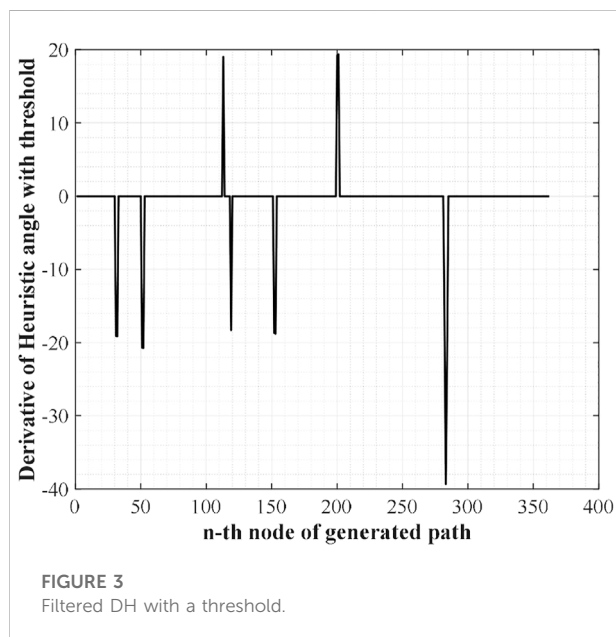
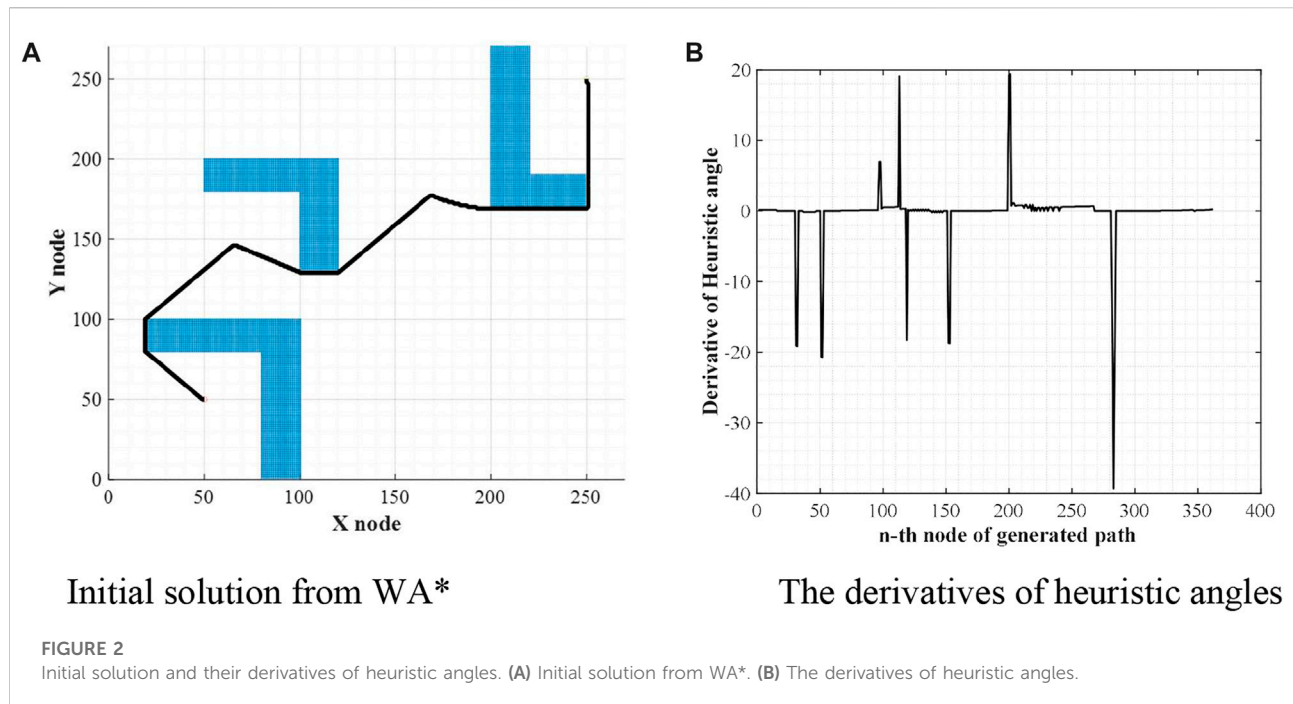
solution only with the direction of the path, it does not require complicated mathematical logic. Thus, WA*DH has the advantage that users can easily understand how WA*DH can improve its initial solution. However, we found that the performance of WA*DH at a certain range of inflation factors is worse than that of the larger inflation factor. We supposed that this is because of the quality of the initial solution of WA*DH and this problem makes WA*DH hard to use in the self-moving vehicles used in extreme environments.

In order to address this problem, this study introduces WA*DH+. WA*DH+ is motivated by WA*DH, so the overall procedures of WA*DH+ are the same as those of WA*DH. The difference between WA*DH and WA*DH+ is that WA*DH+ uses the GBFS algorithm to get the initial solution, whereas WA*DH uses the WA* to get the initial solution. We confirmed from the simulations that the suggested method not only reduces the elapsed times but also removes the probability of the degradation of the performance of the algorithm. From the simulation results, we believe that WA*DH+ will make self-moving vehicles used in extreme environments more responsive.

Weighted A* with the derivative of the heuristic angle

As stated in the introduction, WA*DH uses the derivative of the heuristic angle (hereinafter referred to as DH) to refine the initial solution. The heuristic angle can be defined as (1), and its schematic diagram is stated in Figure 1. In Eq. 1, $\{T, n_1, n_2\} \in \mathcal{P}$ denotes the target node, a current node, and the parent node of a current node, respectively. \mathcal{P} denotes the set of nodes that consist of the path. T denotes the target, and subscripts $\{\text{target}, n, n-1\}$ are subscripts of $\{T, n_1, n_2\}$, respectively:

$$\theta_{ha} = \left| \tan^{-1} \left(\frac{y_{\text{target}} - y_n}{x_{\text{target}} - x_n} \right) - \tan^{-1} \left(\frac{y_n - y_{n-1}}{x_n - x_{n-1}} \right) \right| \quad (1)$$



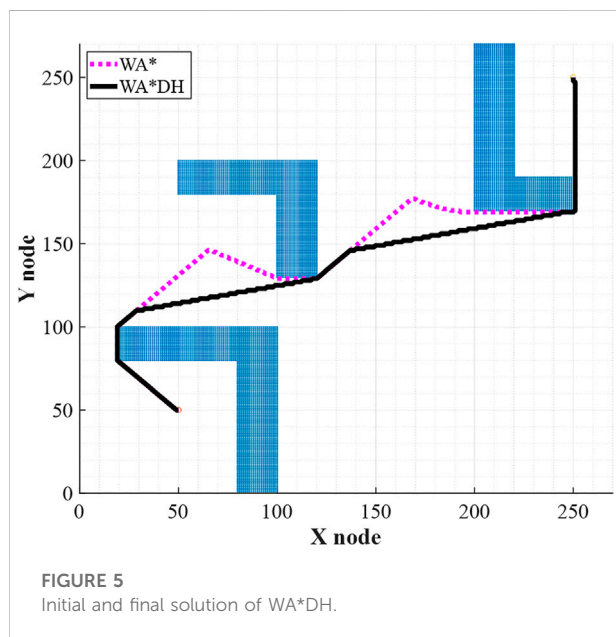
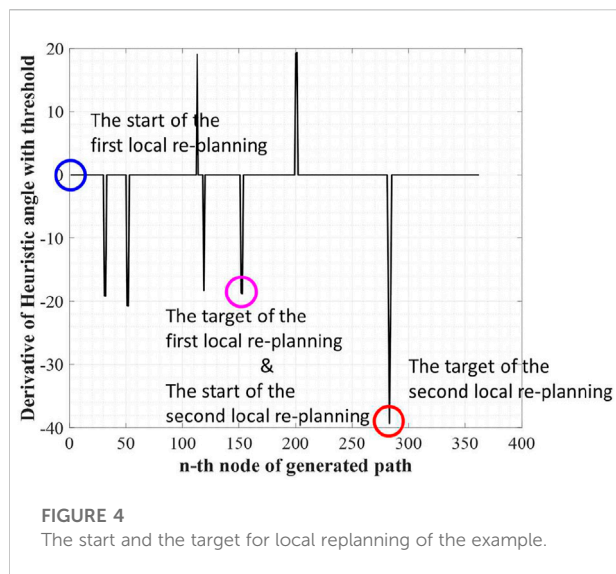
The first step of WA*DH is getting an initial solution from WA*. Once the procedures of WA* are executed, coordinates of nodes that consist of the initial solution will be listed. With this list, WA*DH calculates heuristic angles and their derivatives. The shape of the initial solution and the derivatives of heuristic angles are stated in Figures 2A,B.

From Figure 2B, there are some noise-shaped patterns near 0. These elements can be considered suboptimal nodes by the definition of suboptimal nodes, so they must be eliminated. To do so, WA*DH filters them with a threshold defined as (2). With these methods, Figure 2B will be changed to Figure 3. In Eq. 2, $\theta'_{ha}(n)$ denotes the derivative of the heuristic angle of n_{th} node of the initial solution and m denotes the total number of nodes that consist of the initial solution:

$$Threshold = \sum_{n=1}^m (\theta'_{ha}(n)) \div m \quad (2)$$

The next step of WA*DH is searching suboptimal nodes. The suboptimal node contains two nodes: occurrence and escape. The occurrence node is defined as a node with a positive DH, and the escape node is defined as a node with negative DH. Also, the escape node must satisfy one more condition; there must be at least one occurrence node between the start and a node with negative DH. By the definition of suboptimal nodes, we can find two occurrence nodes near the 110th and 200th nodes and three escape nodes near the 120th, 150th, and 280th nodes from Figure 3.

The purpose of searching suboptimal nodes is to make the node-set. The node-set contains the start and the target nodes for local replanning. Local replanning needs to be executed for the number of occurrence nodes. In the example environment, there are two occurrence nodes, so local replanning needs to be executed twice. The start and the



target for local replanning can be determined as follows and Figure 4 shows the start and the target for local replanning of the example environment:

- 1) The start of the first local replanning is the start of the initial solution, and the target of the first local replanning is the first escape node. However, if there are two or more escape nodes between two occurrence nodes (or target of the initial solution), the last escape node will be chosen for the target of the local replanning.
- 2) The start and the target of the second and the subsequent local replanning are targets of the previous local replanning

and n_{th} escape node. If there are two or more escape nodes between two occurrence nodes (or target of the initial solution), the last escape node will be chosen for the target of the local replanning.

Next, WA*DH executes WA* multiple times based on the node-set. The number of executions is the same as the number of occurrence nodes, and the inflation factor in each execution is equal to the initial solution. After that, WA*DH replaces the initial solution with the locally replanned paths. The procedure of the replacement contains not only replacing nodes but also updating $g(n)$ and $h(n)$. As a result, WA*DH returns its result. Figure 5 states the initial solution and final solution of WA*DH in a dotted line and a full line, respectively.

WA*DH+: Locally replans paths based on the infinitely bounded suboptimal solution

Critical values on WA*DH

Theoretically, the path cost of WA*DH decreases with the decreasing inflation factor. However, we found that the path cost of WA*DH with a certain inflation factor is higher than that with a larger inflation factor. This is stated in Figure 6 and Table 1. In Figures 6B,D,F, θ'_{haf} denotes the filtered DH with a threshold; a dotted line and a full line in Figures 6A,C,E denote the initial and final solution of WA*DH, respectively. Table 1 states the elapsed times and path costs of each simulation.

From the initial solution in Figures 6A,C,E, it is intuitively clear that there are two detouring sections, so we can expect that there will be two escape nodes. When $\epsilon = 2$, WA*DH detected two escape nodes correctly, so the quality of the path cost is equal to A^* . However, when $\epsilon = 1.8$, WA*DH detected only one escape node near the 300_{th} node and the path cost of WA*DH is higher than the path cost of WA*DH when $\epsilon = 2$. We first supposed this is because WA*DH cannot detect all suboptimal nodes. However, as stated in Figure 6D, WA*DH with $\epsilon = 1.5$ found only one escape node at the same location in Figure 6D, but its path cost is equal to A^* . From these results, we concluded that the fault detection of suboptimal nodes is not the cause of the poor performance. Also, we defined inflation factors that make performance degradation critical value.

Get an initial solution from the greedy GFS algorithm

It is hard to avoid the threat of critical values because they are unpredictable. We thought that using a large inflation factor would avoid the threat of the critical value. However, it is hard to decide on a large inflation

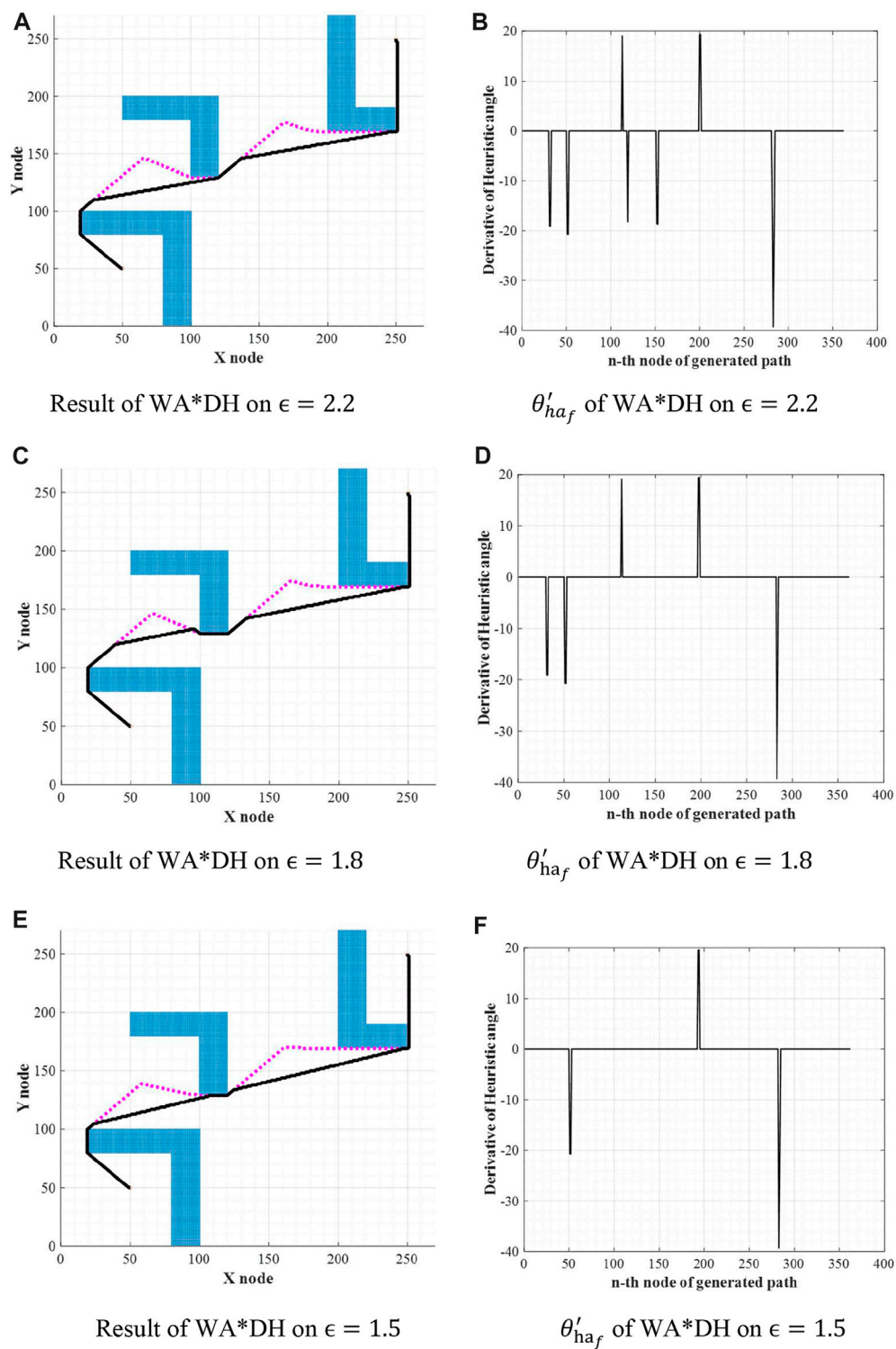


FIGURE 6

Results of WA*DH on the different inflation factors. (A) Result of WA*DH on $\epsilon = 2.2$. (B) θ'_{ha_f} of WA*DH on $\epsilon = 2.2$. (C) Result of WA*DH on $\epsilon = 1.8$. (D) θ'_{ha_f} of WA*DH on $\epsilon = 1.8$. (E) Result of WA*DH on $\epsilon = 1.5$. (F) θ'_{ha_f} of WA*DH on $\epsilon = 1.5$.

TABLE 1 Elapsed times and costs Figure 4.

Inflation factor	Time (s)	Cost (node)
2.2	2.04	404.84
1.8	2.49	408.15
1.5	4.08	404.84
1 (= ∞)	15.40	404.84

factor because the criteria for big and small are different for each person. From this, we hypothesized that an extremely high inflation factor, such as an infinite inflation factor, will be enough to call a large inflation factor. Therefore, we suggest the greedy best-first search (GBFS) algorithm as the algorithm for the initial solution. GBFS is an algorithm that searches nodes with only heuristic, so we thought that using the GBFS algorithm is equal to using WA* with the infinite inflation factor. The cost function of GBFS is stated in Eq. 3, where $f(n)$ denotes the cost function of a node of the GBFS algorithm and $h(n)$ denotes the heuristic of a node:

$$f(n) = h(n) \quad (3)$$

Theorem. If the inflation factor is extremely high (or infinite), then the effect of the g term will be disappeared. Here, g is the cost of the path from the start node to the n_{th} node.

Proof.

Let the cost function of an algorithm be

$$f(\epsilon) = g + \epsilon \times h \quad (4)$$

We can change Eq. 4 to

$$\frac{f(\epsilon)}{\epsilon} = \frac{g}{\epsilon} + h \quad (5)$$

Taking the limit on both sides of Eq. 5,

$$\lim_{\epsilon \rightarrow \infty} \left(\frac{f(\epsilon)}{\epsilon} \right) = \lim_{\epsilon \rightarrow \infty} \left(\frac{g}{\epsilon} \right) + \lim_{\epsilon \rightarrow \infty} (h) \quad (6)$$

$$\frac{f(\epsilon)}{\epsilon} \approx h \quad (7)$$

$$\therefore f(\epsilon) \approx \epsilon \times h \quad (8)$$

The role of the inflation factor on the cost function, such as Eq. 4, is deciding the influence of the heuristics compared to the cost of the path, $g(n)$. However, there is no need for the inflation factor because Eq. 8 does not contain $g(n)$ anymore. so we can remove ϵ from Eq. 8. With these procedures, we can derive Eq. 3 as a result.

Using the GBFS algorithm gives us some advantages, as stated in Figure 7. Figure 7 states escape nodes found from the result of WA* with $\epsilon = 1.8$, $\epsilon = 2.2$, and the GBFS algorithm,

and circles in Figures 7A,C,E state the locations of occurrence nodes.

WA*DH in Figure 7B detected two occurrence nodes, so we can expect there would be two escape nodes. However, WA*DH in Figure 7 detected only one escape node. Also, in Figure 7D, WA*DH detected two occurrence nodes and two escape nodes. Moreover, WA*DH in Figure 7F found three occurrence nodes and escape nodes. In fact, considering the placement of obstacles in the simulation environment of Figure 7, there must be three occurrence nodes and escape nodes. However, Figures 7B,D could not detect all suboptimal nodes due to the relatively high optimality of WA*. From these results, we can prove that the high inflation factor can detect suboptimal nodes clearly.

The path planning with the GBFS algorithm is also very useful in terms of elapsed time. The elapsed time of WA* gets shorter as the inflation factor increases. This means that an infinite inflation factor can theoretically get a result of WA* in the fastest time. Thus, we can expect that the GBFS algorithm can reduce the elapsed time of our algorithm, WA*DH+. This will be simulated in Section 4.

Procedures of WA*DH+

This section introduces our algorithm, WA*DH+, and how WA*DH+ gets its result. Procedures of WA*DH+ are similar to those of WA*DH: getting an initial solution, finding escape nodes, locally replanning the paths, and connecting them. The details of procedures of WA*DH+ are stated below.

First of all, WA*DH+ gets an initial solution from the GBFS algorithm. Unlike WA*DH, WA*DH+ does not need to decide the inflation factor for the initial solution. After getting an initial solution, then WA*DH+ calculates θ'_{haf} with the moving median filter and the threshold.

The next step of WA*DH+ is finding the suboptimal nodes to decide the start and the targets for local replanning. Next, WA*DH+ executes the local replanning. In this procedure, WA*DH+ needs to decide on an inflation factor.

The final step of WA*DH+ is path-connecting. In this procedure, WA*DH+ needs to update only heuristics, whereas WA*DH needs to update the actual costs, $g(n)$, and heuristics. Also, WA*DH+ needs an additional procedure to calculate the path cost, whereas WA*DH can get its path cost from $g(n)$ of the target. After calculating the path cost, WA*DH+ can finally get its result. The pseudocode and the flowchart of WA*DH+ are stated in Figures 8, 9. In the flowchart, when $i = 1$, the $(i-1)_{th}$ node is equal to the original start node.

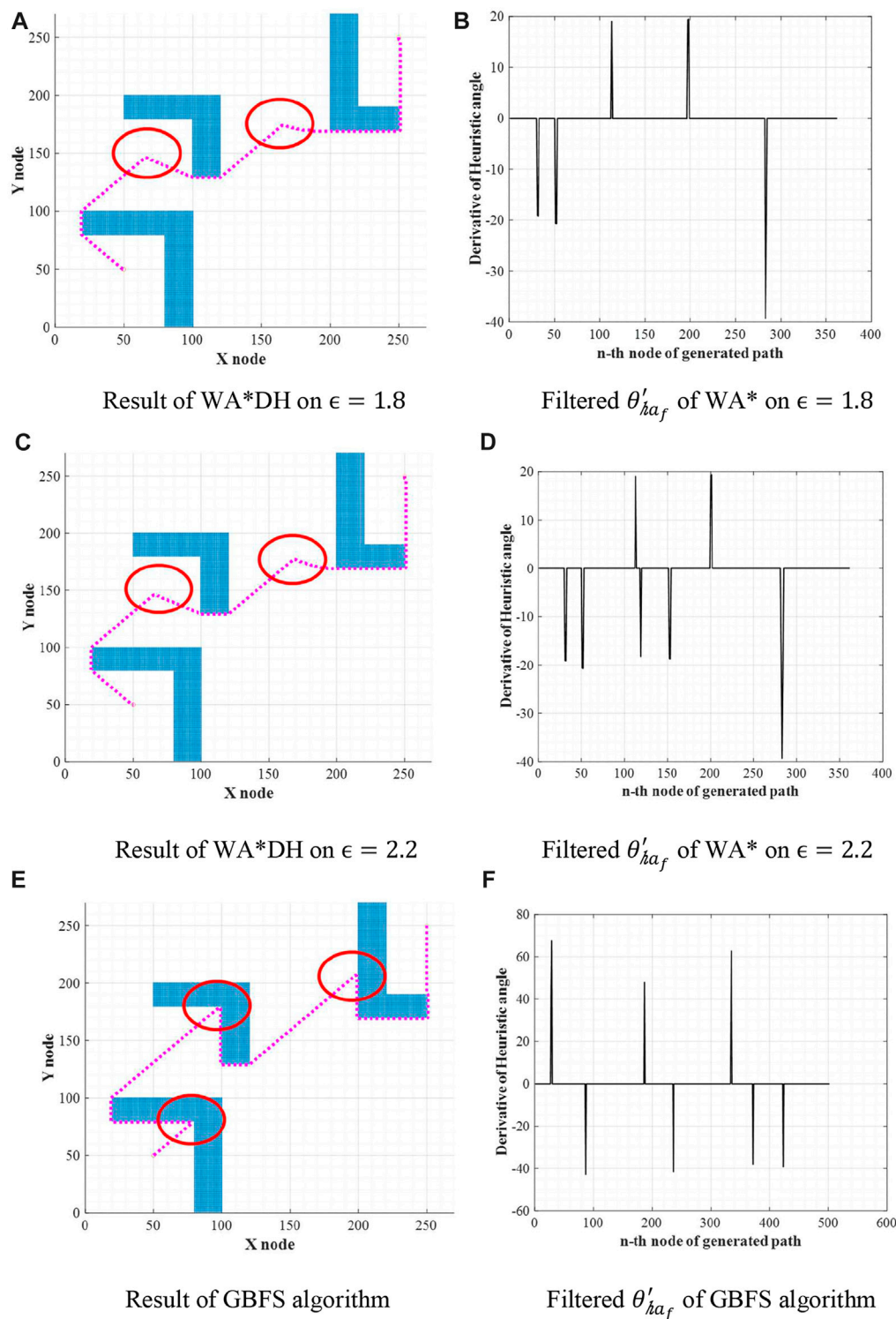


FIGURE 7

Escape nodes of WA* on different inflation factors. (A) Result of WA*DH on $\epsilon = 1.8$. (B) Filtered θ'_{ha_f} of WA* on $\epsilon = 1.8$. (C) Result of WA*DH on $\epsilon = 2.2$. (D) Filtered θ'_{ha_f} of WA* on $\epsilon = 2.2$. (E) Result of the GBFS algorithm. (F) Filtered θ'_{ha_f} of the GBFS algorithm.

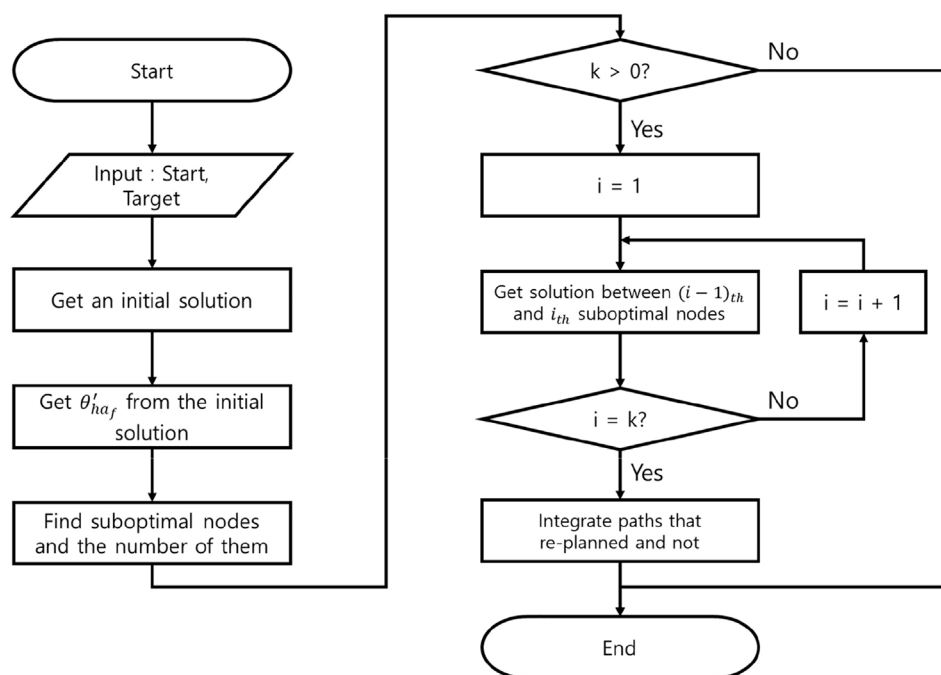


FIGURE 8
Flowchart of WA*DH+.

Simulation results

Simulation environments

In order to compare the performances with WA*DH, obstacles were placed the same as in the simulation environments of WA*DH, and the sizes of all simulation environments are also the same as those of the simulation environments of WA*DH (270 x 27 nodes) as stated in Figure 10. All simulations were conducted in MATLAB with Windows 10, i7-9700 CPU with 32 GB RAM, and there are no acceleration methods or parallel processes such as Graphics Processing Unit (GPU) and parallel processing with CPU cores.

Performances of WA*DH+

To validate the performance of WA*DH+, we simulated WA*DH+, WA*DH, and WA* in terms of the path cost and elapsed time in each environment. Simulations were conducted by reducing the inflation factor from 3 to 1 by 0.2 to compare performances near critical values. Results of simulations are stated in Figures 11–14, and quantitative comparisons are stated in Tables 2–5.

From Table 2, the path cost of WA*DH with a relatively large inflation factor (> 2.2) is the same path cost of A*s. However,

when ϵ is in the range from 2.2 to 1.6, the path cost of WA*DH is about 0.81% larger than that with larger inflation factors. In contrast, the path cost of WA*DH + does not change with varying inflation factors, and its quality keeps the same as the path cost of A*.

Table 3 shows that path costs of WA*DH and WA*DH+ with relatively high inflation factors (> 2.0) are the same as the path cost of A*. However, when the inflation factor is lower than 2, the path cost of WA*DH is the same as WA* until the inflation factor decreases to 1, whereas the path cost of WA*DH + does not change with varying inflation factors.

In the case of simulation case 3, the path cost of WA*DH+ is about 5.04% lower than that of WA*DH. Thus, we can see that WA*DH + not only removes the risk of a critical value but also returns a lower path cost at a relatively high inflation factor. Moreover, we also confirmed that the path cost of WA*DH + keeps the same quality as the path cost of A* regardless of the inflation factor.

Unlike other results of simulation cases, when $\epsilon = 3$ in case of simulation case 4, the path cost of WA*DH+ is about 0.5% higher than that of WA*DH, and this difference rises to about 1.47% when $\epsilon = 1.6$. However, it is hard to say that the quality of WA*DH+ is bad because the purpose of using WA*DH+ is to avoid the threat of the critical value.

Besides being free from the threat of the critical value, we also found that WA*DH + has an advantage in terms of the

Algorithm 1. WA*DH+(Weighted A* with the Derivative of Heuristic angle +)

Procedure Main()

```

1: GBFS()
2: get  $\theta'_{ha}$  from all nodes on the path
3: node_set = detect_nodes( $\theta'_{ha}$ )
4: replanned_path = local_replanning(node_set)
5: Result = connect_paths(replanned_path, original_path)
6: return Result

```

Procedure GBFS()

```

7:  $h(n_{start}) = c(start, target)$ ;
8: OPEN = OPEN  $\cup$  ( $n_{start}$ )
9: CLOSED =  $\emptyset$ ;
10: while ( $n_n \neq n_{target}$ )
11:   remove  $n$  with the smallest  $f(n)$  from OPEN
12:   CLOSED = CLOSED  $\cup$  { $n$ };
13:   for each successor  $n'$  of  $n$ 
14:     if  $n'$  was not visited before then
15:       insert  $n'$  into OPEN with  $f(n')$ 

```

Procedure WA*()

```

16:  $g(n_{start}) = 0$ ;
17: OPEN = CLOSED =  $\emptyset$ ;
18: while ( $n_n \neq n_{target}$ )
19:   remove  $n$  with the smallest  $f(n)$  from OPEN
20:   CLOSED = CLOSED  $\cup$  { $n$ };
21:   for each successor  $n'$  of  $n$ 
22:     if  $n'$  was not visited before, then
23:        $g(n') = \infty$ ;
24:       if  $g(n') > g(n) + c(n, n')$ 
25:          $g(n') = g(n) + c(n, n')$ 
26:       if  $n' \notin$  CLOSED
27:         insert  $n'$  into OPEN with  $f(n')$ 

```

Procedure detect_nodes()

```

28: node_set =  $\emptyset$ ;  $N = \emptyset$ ;
29:  $i = 0$ ; flag = 0; size_n = 0;
30: while  $i < size(\theta'_{ha}())$ 
31:    $i++$ ;
32:   if  $\theta'_{ha}(i) > 0$ 
33:     flag = 1;
34:   while flag = 1 &&  $i < size(\theta'_{ha}())$ 
35:      $i++$ ;
36:     if  $\theta'_{ha}(i) < 0$ 
37:       size_n++;
38:        $N(size\_n) = i$ ;
39:       while  $\theta'_{ha}(i) < 0$  &&  $i < size(\theta'_{ha}())$ 
40:          $i++$ ;
41:         if  $\theta'_{ha}(i) < 0$ 
42:            $N(size\_n) = i$ ;
43:           flag = 0;
44:   for  $n = 1 : size(N)$ 
45:     if  $n = 1$ 
46:       node_set( $n$ ) = [start_node original_path( $N(n)$ )];
47:     else
48:       node_set( $n$ ) = [original_path( $N(n-1)$ ) original_path( $N(n)$ )];
49:   return node_set;

```

Procedure local_replanning()

```

50: replanned_path =  $\emptyset$ ;
51: for  $j = 1 : size(col(node\_set))$ 
52:   if  $j = 1$ 
53:     replanned_path =  $WA_e^*(node\_set(j))$ ;
54:   else
55:     local_path =  $WA_e^*(node\_set(j))$ ;
56:     replanned_path = [replanned_path; local_path];
57:   return replanned_path

```

Procedure connect_paths()

```

58: nr_path = original_path(node_set(end):end);
59: for all  $h$  of nodes of replanned_path
60:    $f(n) = c(n, goal)$ ;
61:   update  $f$  of replanned_path with changed  $h$ 
62:   final_path = [replanned_path; nr_path];
63:   return final_path

```

FIGURE 9
Pseudocode of WA*DH+.

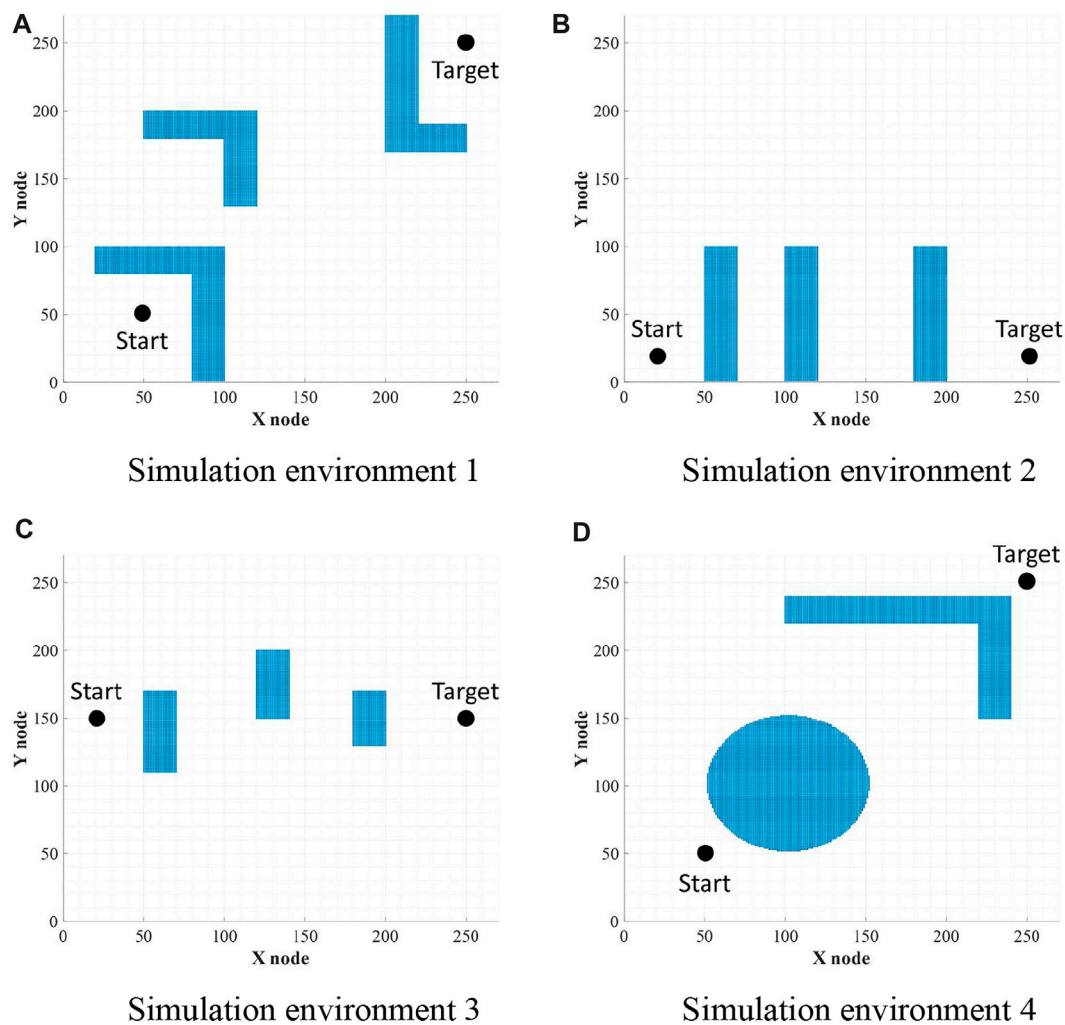


FIGURE 10

Environments for simulations. (A) Simulation environment 1. (B) Simulation environment 2. (C) Simulation environment 3. (D) Simulation environment 4.

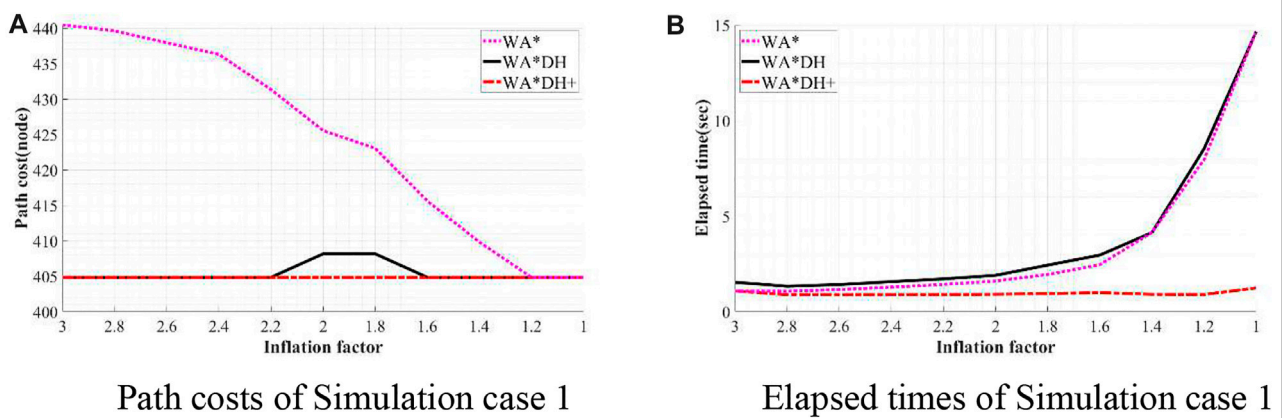


FIGURE 11

Result of simulation case 1. (A) Path costs of simulation case 1. (B) Elapsed times of simulation case 1.

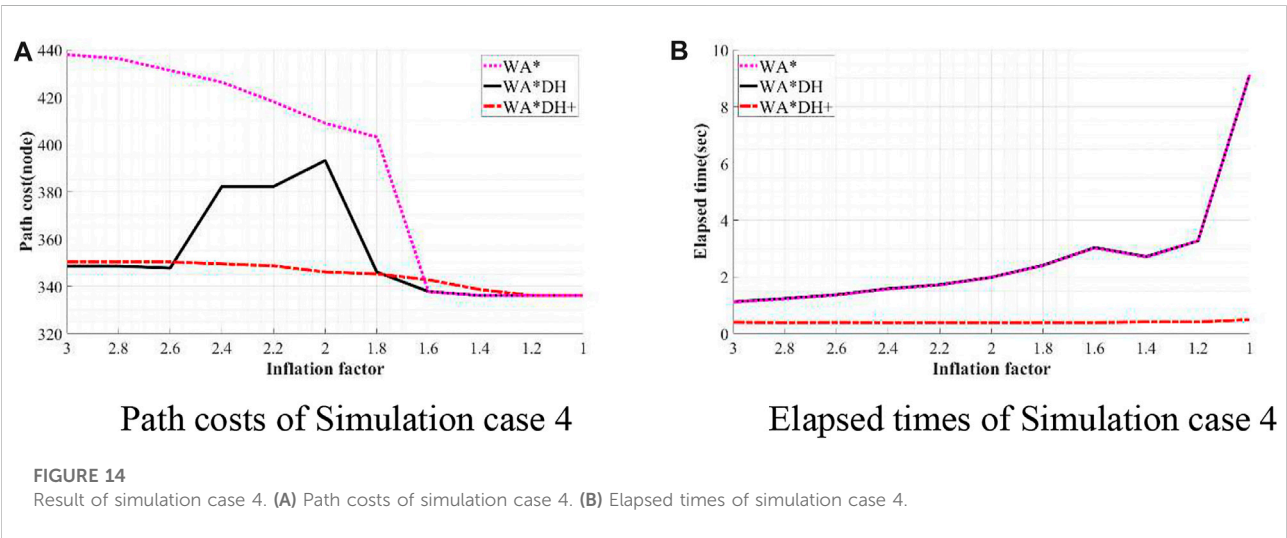
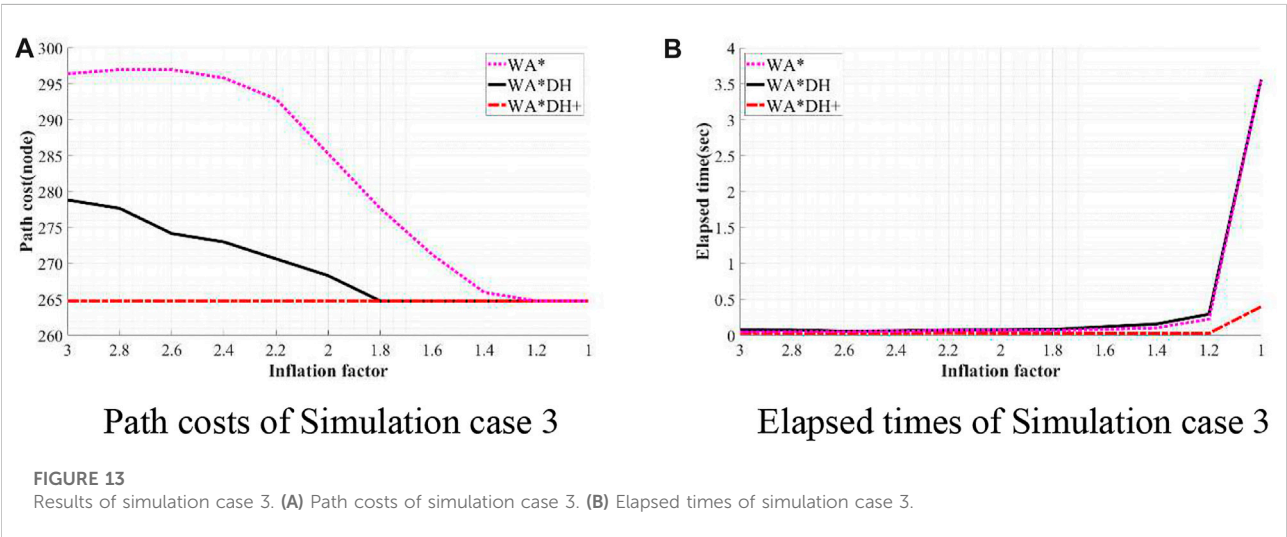
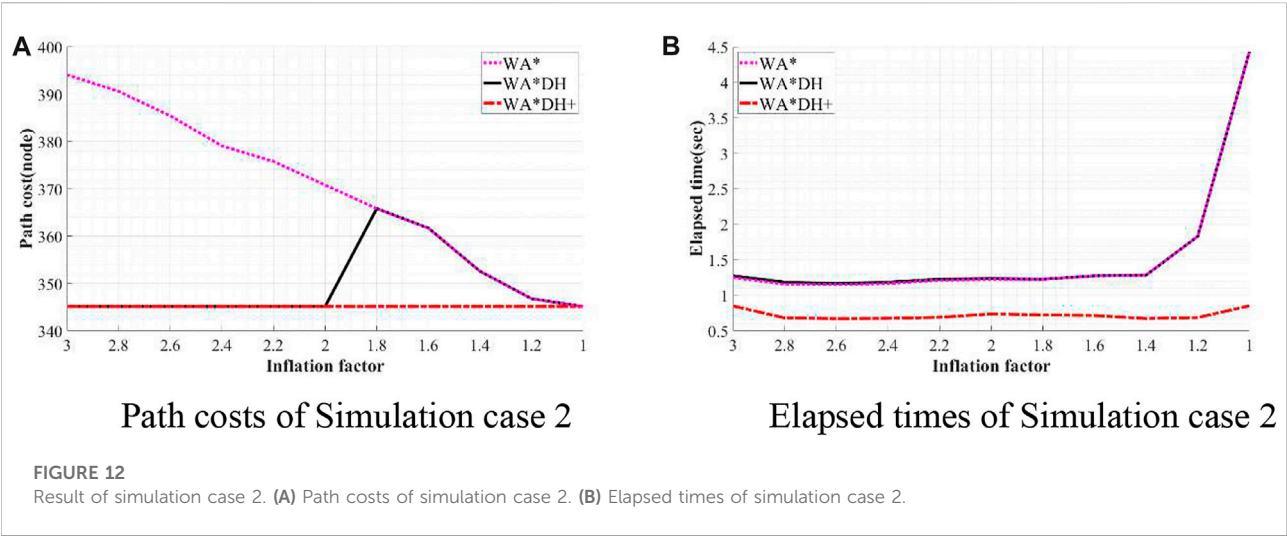


TABLE 2 Quantitative comparison of the performances of algorithms on simulation 1.

€	WA*DH+		WA*DH		WA*	
	Cost (node)	Time (s)	Cost (node)	Time (s)	Cost (node)	Time (s)
3	404.84	1.08	404.84	1.54	440.46	1.08
2.8	404.84	0.88	404.84	1.33	439.63	1.07
2.6	404.84	0.88	404.84	1.41	437.97	1.15
2.4	404.84	0.88	404.84	1.57	436.32	1.28
2.2	404.84	0.88	404.84	1.71	431.35	1.42
2.0	404.84	0.91	408.15	1.90	425.55	1.60
1.8	404.84	0.94	408.15	2.43	423.06	1.94
1.6	404.84	0.99	404.84	2.96	415.61	2.46
1.4	404.84	0.90	404.84	4.13	409.81	4.14
1.2	404.84	0.89	404.84	8.52	404.84	7.95
1.0	404.84	1.24	404.84	14.65	404.84	14.65

TABLE 3 Quantitative comparison of the performances of algorithms on simulation 2.

€	WA*DH+		WA*DH		WA*	
	Cost (node)	Time (s)	Cost (node)	Time (s)	Cost (node)	Time (s)
3	345.14	0.85	345.14	1.27	394.13	1.25
2.8	345.14	0.68	345.14	1.18	390.62	1.15
2.6	345.14	0.67	345.14	1.16	385.44	1.15
2.4	345.14	0.68	345.14	1.18	379.1	1.17
2.2	345.14	0.69	345.14	1.22	375.79	1.21
2.0	345.14	0.74	345.14	1.23	370.82	1.22
1.8	345.14	0.72	365.85	1.23	365.85	1.23
1.6	345.14	0.72	361.71	1.27	361.71	1.27
1.4	345.14	0.67	352.59	1.29	352.59	1.29
1.2	345.14	0.69	346.79	1.83	346.79	1.83
1.0	345.14	0.85	345.14	4.43	345.14	4.43

TABLE 4 Quantitative comparison of the performances of algorithms on simulation 3.

€	WA*DH+		WA*DH		WA*	
	Cost (node)	Time (s)	Cost (node)	Time (s)	Cost (node)	Time (s)
3	264.79	0.03	278.85	0.08	296.43	0.05
2.8	264.79	0.03	277.68	0.08	297.01	0.05
2.6	264.79	0.03	274.17	0.06	297.01	0.05
2.4	264.79	0.03	273.00	0.06	295.84	0.06
2.2	264.79	0.04	270.65	0.08	292.91	0.07
2.0	264.79	0.03	268.31	0.08	285.3	0.07
1.8	264.79	0.03	264.79	0.08	277.68	0.07
1.6	264.79	0.03	264.79	0.12	271.24	0.08
1.4	264.79	0.03	264.79	0.16	265.97	0.10
1.2	264.79	0.03	264.79	0.29	264.79	0.23
1.0	264.79	0.40	264.79	3.56	264.79	3.56

TABLE 5 Quantitative comparison of the performances of algorithms on simulation 4.

ϵ	WA*DH+		WA*DH		WA*	
	Cost (node)	Time (s)	Cost (node)	Time (s)	Cost (node)	Time (s)
3	350.33	0.40	348.58	1.12	437.97	1.11
2.8	350.33	0.39	348.58	1.23	436.32	1.23
2.6	350.33	0.40	347.75	1.37	431.34	1.36
2.4	349.50	0.38	382.23	1.59	426.37	1.57
2.2	348.68	0.39	382.23	1.73	418.09	1.71
2.0	346.09	0.39	393.24	1.99	409.00	1.98
1.8	345.26	0.39	346.09	2.41	403.18	2.40
1.6	342.78	0.39	337.81	3.03	337.81	3.02
1.4	338.63	0.41	336.15	2.72	336.15	2.69
1.2	336.15	0.41	336.15	3.27	336.15	3.27
1.0	336.15	0.49	336.15	9.13	336.15	9.13

elapsed time. In all simulation cases, elapsed times of WA*DH increase exponentially near $\epsilon = 1$. However, the elapsed time of WA*DH + does not significantly increase with decreasing inflation factor, and elapsed times of WA*DH + are much lower than WA*DH. In fact, considering that using the GBFS algorithm has the same meaning as using the infinite inflation factor at WA*, it is a natural result because the higher the inflation factor is, the faster the result can be returned.

Discussion

This study aims to evade the threat of the critical values on WA*DH. We found that the critical value comes from the fault detection of suboptimal nodes from the initial solution with relatively high optimality. We hypothesized that high suboptimality could find suboptimal nodes clearly, so we suggested our algorithm, WA*DH +, which uses the GBFS algorithm for the initial solution. From simulations, it can be proven that WA*DH + can avoid the threat of the critical value successfully by detecting suboptimal nodes more clearly than WA*DH. In terms of the elapsed time, we also confirmed that the elapsed time does not change significantly with varying inflation factors; however, the elapsed time of WA*DH increases exponentially when the inflation factor is near 1.

Although WA*DH + shows powerful performances in terms of the path cost and the elapsed time, WA*DH + still cannot guarantee admissibility because WA*DH + refines the initial solution based on the GBFS algorithm that has the infinitely bounded suboptimality. It will remain a limitation of algorithms using the concept of WA*DH. Also, WA*DH+ cannot refine the initial solution if there are circular obstacles in the path planning scene. However, we expect this will be removed in future works using new filtering methods of DH or additional procedures to the result of WA*DH+.

Data availability statement

The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding author.

Author contributions

DL proposed the idea and simulated the algorithm and wrote the paper except the simulation part. JJ analyzed the result of the computer simulation and wrote the computer simulation part of the paper.

Funding

This research was a part of project “No. 20190497,” funded by the Korea Coast Guard, Korea.

Conflict of interest

DL and JJ were employed by Vessel aerospace Co.Ltd.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Aine, S., Swaminathan, S., Narayanan, V., Hwang, V., and Likhachev, M. (2016). Multi-heuristic a. *Int. J. Robotics Res.* 35 (1-3), 224–243. doi:10.1177/0278364915594029
- Boulares, M., and Barnawi, A. (2021). A novel UAV path planning algorithm to search for floating objects on the ocean surface based on object's trajectory prediction by regression. *Robotics Aut. Syst.* 135, 103673. doi:10.1016/j.robot.2020.103673
- Dai, X., Long, S., Zhang, Z., and Gong, D. (2019). Mobile robot path planning based on ant colony algorithm with A* heuristic method. *Front. Neurobot.* 13, 15. doi:10.3389/fnbot.2019.00015
- De Momi, E., Kranendonk, L., Valenti, M., Enayati, N., and Ferrigno, G. (2016). A neural network-based approach for trajectory planning in robot-human handover tasks. *Front. Robot. AI* 3, 34. doi:10.3389/frobt.2016.00034
- Duan, C., Hu, Q., Zhang, Y., and Wu, H. (2020). Constrained single-axis path planning of underactuated spacecraft. *Aerosp. Sci. Technol.* 107, 106345. doi:10.1016/j.ast.2020.106345
- Fu, B., Chen, L., Zhou, Y., Zheng, D., Wei, Z., Dai, J., et al. (2018). An improved A* algorithm for the industrial robot path planning with high success rate and short length. *Robotics Aut. Syst.* 106, 26–37. doi:10.1016/j.robot.2018.04.007
- Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cyber.* 4 (2), 100–107. doi:10.1109/tssc.1968.300136
- Hou, M., Cho, S., Zhou, H., Edwards, C. R., and Zhang, F. (2021). Bounded cost path planning for underwater vehicles assisted by a time-invariant partitioned flow field model. *Front. Robot. AI* 8, 575267. doi:10.3389/frobt.2021.575267
- Jing, X., and Yang, X. (2018). Application and improvement of heuristic function in A* algorithm 2018 37th Chinese Control Conference (CCC), 25–27 July 2018, Wuhan, China.
- Li, B., Gong, J., Jiang, Y., Nasry, H., and Xiong, G. (2012). ARA*+: Improved path planning algorithm based on ARA. *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 04–07 December 2012, Macau, China.
- Likhachev, M. (2005). Anytime dynamic A*: An anytime, Replanning Algorithm. *ICAPS*. 5, 262–271.
- Likhachev, M., Gordon, G. J., and Thrun, S. (2003). ARA*: Anytime A* with provable bounds on sub-optimality. *Adv. neural Inf. Process. Syst.* 16, 767–774.
- Lim, D., Park, J., Han, D., Jang, H., Park, W., Lee, D., et al. (2020). UAV path planning with derivative of the heuristic angle. *Int. J. Aeronaut. Space Sci.* 22, 140–150. doi:10.1007/s42405-020-00323-1
- Liu, Z., Yue, M., Guo, L., and Zhang, Y. (2020). Trajectory planning and robust tracking control for a class of active articulated tractor-trailer vehicle with on-axle structure. *Eur. J. Control* 54, 87–98. doi:10.1016/j.ejcon.2019.12.003
- Mac, T. T., Copot, C., Tran, D. T., and De Keyser, R. (2016). Heuristic approaches in robot path planning: A survey. *Robotics Aut. Syst.* 86, 13–28. doi:10.1016/j.robot.2016.08.001
- Pearl, J., and Kim, J. H. (1982). Studies in semi-admissible heuristics. *IEEE Trans. Pattern Anal. Mach. Intell.* 4, 392–399. doi:10.1109/tpami.1982.4767270
- Pohl, I. (1970). Heuristic search viewed as path finding in a graph. *Artif. Intell.* 1 (3–4), 193–204. doi:10.1016/0004-3702(70)90007-x
- Pohl, I. (1973). The avoidance of (relative) catastrophe, heuristic competence, genuine dynamic weighting and computational issues in heuristic problem solving. In *Proceedings of the 3rd international joint conference on Artificial intelligence*, Stanford USA, August 20–23, 1973, 12–17.
- Thayer, J., and Ruml, W. (2009). Using distance estimates in heuristic search. In *Proceedings of the International Conference on Automated Planning and Scheduling*, Thessaloniki, September 19–23, 2009 (Vol. 19, 1).
- Thayer, J. T., and Ruml, W. (2008). Faster than weighted A*: An optimistic approach to bounded suboptimal search. In *ICAPS*, New Delhi, India, June, 2008, 355–362.
- Yang, L., Qi, J., Song, D., Xiao, J., Han, J., and Xia, Y. (2016). Survey of robot 3D path planning algorithms. *J. Control Sci. Eng.*, 1–22. doi:10.1155/2016/7426913
- Yang, L., Qi, J., Xiao, J., and Yong, X. (2014). A literature review of UAV 3D path planning. *Proceeding of the 11th World Congress on Intelligent Control and Automation*, 29 June 2014 – 04 July 2014, Shenyang
- Yiu, Y. F., Du, J., and Mahapatra, R. (2018). Evolutionary heuristic a* search: Heuristic function optimization via genetic algorithm. *IEEE First International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)* 26–28 September 2018, Laguna Hills, CA, USA.
- Zhou, R., and Hansen, E. A. (2002). Multiple sequence alignment using A*. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, Alberta, Canada, July 28–August 1, 2002.



Treatise on Analytic Nonlinear Optimal Guidance and Control Amplification of Strictly Analytic (Non-Numerical) Methods

Timothy Sands *

Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY, United States

OPEN ACCESS

Edited by:

Allahyar Montazeri,
Lancaster University, United Kingdom

Reviewed by:

Khoshnam Shojaei,
Islamic Azad University of
Najafabad, Iran
Nargess Sadeghzadeh-Nokhodberiz,
Qom University of Technology, Iran
Imil Imran,
Lancaster University, United Kingdom

*Correspondence:

Timothy Sands
tas297@cornell.edu

Specialty section:

This article was submitted to
Robotic Control Systems,
a section of the journal
Frontiers in Robotics and AI

Received: 26 February 2022

Accepted: 16 May 2022

Published: 04 October 2022

Citation:

Sands T (2022) Treatise on Analytic
Nonlinear Optimal Guidance and
Control Amplification of Strictly Analytic
(Non-Numerical) Methods.
Front. Robot. AI 9:884669.
doi: 10.3389/frobt.2022.884669

Optimal control is seen by researchers from a different perspective than that from which the industry practitioners see it. Either type of user can easily become confounded when deciding which manner of optimal control should be used for guidance and control of mechanics. Such optimization methods are useful for autonomous navigation, guidance, and control, but their performance is hampered by noisy multi-sensor technologies and poorly modeled system equations, and real-time on-board utilization is generally computationally burdensome. Some methods proposed here use noisy sensor data to learn the optimal guidance and control solutions in real-time (online), where non-iterative instantiations are preferred to reduce computational burdens. This study aimed to highlight the efficacy and limitations of several common methods for optimizing guidance and control while proposing a few more, where all methods are applied to the full, nonlinear, coupled equations of motion including cross-products of motion from the transport theorem. While the reviewed literature introduces quantitative studies that include parametric uncertainty in nonlinear terms, this article proposes accommodating such uncertainty with time-varying solutions to Hamiltonian systems of equations solved in real-time. Five disparate types of optimum guidance and control algorithms are presented and compared to a classical benchmark. Comparative analysis is based on tracking errors (both states and rates), fuel usage, and computational burden. Real-time optimization with singular switching plus nonlinear transport theorem decoupling is newly introduced and proves superior by matching open-loop solutions to the constrained optimization problem (in terms of state and rate errors and fuel usage), while robustness is validated in the utilization of mixed, noisy state and rate sensors and uniformly varying mass and mass moments of inertia. Compared to benchmark, state-of-the-art methods state tracking errors are reduced one-hundred ten percent. Rate tracking errors are reduced one-hundred thirteen percent. Control utilization (fuel) is reduced eighty-four percent, while computational burden is reduced ten percent, simultaneously, where the proposed methods have no control gains and no linearization.

Keywords: optimization problems, control problems, nonlinear problems, mathematical modeling, transport theorem

1 INTRODUCTION

Considering intermittent coverage and communication delays with typical stellar satellites like those illustrated in **Figure 1**, autonomous guidance and control necessitates real-time on-board computation with demanding accuracy and robustness requirements, despite potentially coarsely known system characteristics, varying environmental conditions, and mission-related constraints. Many solutions have been developed, including optimal analytic methods for simple cases (Chai et al., 2019), while optimal methods for guiding and controlling realistic nonlinear systems ubiquitously necessitate either computational solutions or linearization to achieve analytical solutions. *This manuscript proposes new techniques for utilizing optimization techniques applied to the full, nonlinear, coupled equations of mechanical motion, and the techniques are analytic as opposed to numeric.* Rao proposed numerical trajectory optimization applied to orbital transfer problems (Rao et al., 2002) and also produced a survey of numerical methods for optimal control (Rao, 2009). Numerical methods are very quickly resorted to as researchers grapple with six nonlinear, coupled equations of mechanical motion (both translation and rotation). A generalized treatment method (again numerical) for optimization problems was proposed by Ross and Karpenko (2012) for such orbital transfer problems, spacecraft rendezvous and docking (Gao et al., 2009; Pontani and Conway, 2013; Bonnans and Festa, 2017), and planetary entry and hypersonic space planes (Windhorst et al., 1997; Arora, 2002; Chen et al., 2005; Ivanov et al., 2007; Zhang and Chen, 2011). Arguably, following the publication of Ross (2015), numerical optimization in general form realized the current dominance of numerical methods: for example, Tian et al. (2015) and Sagliano et al. (2017) for real-time (numerical) trajectory optimization and Chai et al. (2018a) and Chai et al. (2018b) for aero-assisted optimal tracking guidance.

Lacking ubiquitous analytic methods to treat the nonlinear, coupled systems of equations, linearization followed by least-squares optimization leads to the so-called Riccati equations (NASA, 2021a) to produce optimal control gains (Kwakernaak

and Sivan, 1972) with presumed error feedback (Kelly, 2005) in both continuous and discrete form (Flugge-Lotz, 1953). Optimization is sometimes sought after first implementing adaptive (Sands et al., 2009) methods to use feedback achieving predictability (Sands, 2019). Duprez et al. (2017) sought to tackle the nonlinear transport theorem terms by proposing control being a Lipschitz vector field on a fixed control set angular velocity, ω . This manuscript seeks to extend the notion of tacking nonlinear transport theorem to include time-varying angular velocities. Arguchintsev and Poplevko (2021) proposed an optimal control for linear hyperbolic systems of ordinary differential equations by estimating the residuals in terms of the value that characterizes the smallness of the measure of the domain of the needle variation of control. Emphasis was placed on problem formulation by Srochko et al. (2021), but the focus was parameterizing the cost functional rather than the nonlinear constraint function as done in this work.

Championed by Lorenz, physics-based methods (Sands and Ghadawala, 2011) were proposed to instantiate “self-sensing machines methods”, where the sensing functions are fully integrated on a drive to detect key operating characteristics including rotor position, torque, speed, temperature, and motor/load diagnostics” (Malecek, 2021). The physics-based methods codified optimal feedforward forms, which were later augmented with optimal feedback (Smeresky et al., 2020), instantiating the relatively new method referred to as deterministic artificial intelligence (D.A.I.). D.A.I. necessitates analytic forms of desired state trajectories (Baker et al., 2018) for the feedforward control and state observers (Heidlauf and Cooper, 2017; Willems, 1971) for the feedback control. In 2021, the utilization of Pontryagin’s approach (Pontryagin et al., 1962; Boltyanskii, 1971) to impose necessary conditions of optimality as a first step led to boundary-value problems that produce optimal controls, but *also* optimal trajectories (Sands, 2021) as alternatives to the sinusoidal trajectories recommended by Baker et al. (2018). These optimal trajectories are utilized in this manuscript as prescriptions for the coupled motion cross-products resulting from the inclusion of the transport theorem of motion.

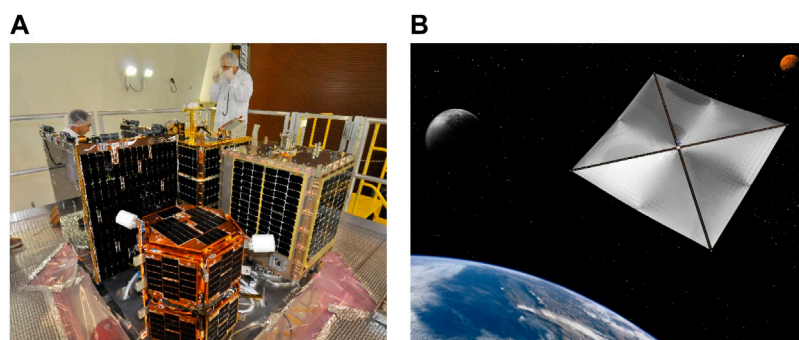


FIGURE 1 | (A) NASA's FASTSAT microsatellite readied to share ride to space (NASA, 2021a). **(B)** NASA ejects nanosatellite from microsatellite in space. Image used consistent with NASA policy, “NASA content (images, videos, and audios) are generally not copyrighted and may be used for educational or informational purposes without needing explicit permissions” (NASA, 2021b).

Thus, the reader may consider the use of classical control methods (proportional plus velocity will be evaluated here) and seek to optimize control gains or use the Ricatti equation to seek linear-quadratic optimal classical control gains. Alternatively, time-optimal control may be considered a feedback form or control-minimizing control in an open-loop feedforward topology. Furthermore, real-time optimal controls could be derived that utilize feedback of state and rate in a matrix-inverse to enforce optimality in a closed-loop. Several different options for matrix inversion are available, generating subsets of the broader category of real-time optimal control. A key limitation of all the methods described so far is the inability to deal with nonlinear, coupled equations generated by the transport theorem in both translational and rotational mechanics.

Parametric uncertainty is another challenging aspect of nonlinear systems. Hu, et al. (2015) investigated nonlinear regression including parameter uncertainty estimates using the Monte Carlo and bootstrap methods to estimate nonlinear parameter uncertainties with a Microsoft Excel spreadsheet. Similarly, Monte Carlo statistical analysis is utilized here in MATLAB/SIMULINK and presented in section 3 Results. A modified James–Stein State Estimator (JSSE), named Modified James–Stein State Estimator (JSSE-M) was proposed by Meda-Campana (2018) as an alternative to filtering the states of nonlinear systems within a control scheme. Ferreres and Fromion (2010) studied the existence of limit-cycles in a closed-loop, which simultaneously contains nonlinearities and parametric uncertainties, addressed using three methods: 1) using a necessary condition of oscillation embodies in a graphical method, 2) checking the absence of limit-cycles despite parametric uncertainties using a sufficient condition of non-oscillation, and 3) using the necessary condition of oscillation to synthesize a controller which modifies the characteristics (magnitude and frequency) of the limit-cycle. A generically similar approach is used here, where necessary conditions of optimality are used to yield a nonlinear controller that can accommodate uncertainties in nonlinear systems. Arguably, a much more common approach to stability robustness of uncertain nonlinear multivariable systems under input-output feedback linearization is to allow plant uncertainty to be propagated through the control design, yielding an uncertainty description of the closed-loop in polytopic form, as presented by Botto et al. (2001).

Recently, Taghieh and Shafiei (2021a) proposed an observer-based robust model predictive control scheme to control a class of switched nonlinear systems in the presence of time delay and parametric uncertainties under arbitrary switching in addition to proposing a static output feedback controller (Taghieh and Shafiei, 2021b) for a class of switched nonlinear systems subject to time-varying delay and uncertainties under asynchronous switching. Zhang et al. (2022) addressed nonlinear systems with mismatched uncertainties under input/output quantization proposing adaptive output feedback control. Fractional parametric uncertainties and distributed delays in nonlinear systems together with time delay, parametric uncertainties and actuator faults were just addressed by Sweetha et al. using a non-fragile fault-tolerant controller,

which makes the system asymptotically stable with the specified mixed H_∞ and passive performance index (Sweetha et al., 2022). Wei et al. (2022) sought to control uncertain nonlinear processes using neural networks incorporating into the control loop an adaptive neural network embedded contraction-based controller (to ensure convergence to time-varying references) and an online parameter identification module coupled with reference generation (to ensure modeled parameters converge those of the physical system). Wang et al. (2009) investigated quasi-Hamiltonian systems with parametric uncertainty using the stochastic averaging method and stochastic dynamical programming principle. A particular strength of the work lies in two examples given to illustrate the proposed control procedure and its robustness. Mahmoodabadi and Andalib Sahnehsaraei (2021) introduced a new online optimal control based on the input–output feedback linearization and a multi-crossover genetic algorithm for under-actuated nonlinear systems having parametric uncertainties. Optimal control problems with bounded uncertainties on parameters were addressed using interval arithmetics by Etienne et al. where an interval method based on Pontryagin’s Minimum Principle (as proposed here) is proposed in Bertin et al. (2021) to enclose the solutions of an optimal control problem with embedded bounded uncertainties. This method is used to compute an enclosure of all optimal trajectories of the problem and open-loop and closed-loop enclosures meant to validate an optimal guidance algorithm on a concrete system with inaccurate knowledge of the parameters.

Next-generation methods are required that apply mathematically optimal results yet retain the simplicity of analytics solutions obfuscating numerical (or otherwise more complicated) methods and providing further advancements in autonomous navigation. The current movement toward the utilization of very small vehicles is accompanied by very limited computational resources while maintaining autonomy, robustness, and accuracy. Newly proposed methods and algorithms for autonomous guidance and control are presented in direct, critical comparison to the recent research trends of both academia and industry, presuming utilization of noisy sensors, for example, star trackers, rate gyroscopes, inertial measurement units, and global navigation systems, amongst other sensors in multi-sensor-based architectures for vehicle navigation. Intelligence methods permitting systems to learn real-time optimal solutions (analytically) are preferred.

Proposed novelties:

1. A brief methodological recitation of five disparate incarnations of optimal control and their direct comparison to classical feedback control as a benchmark (the P + V proportional plus velocity controller): 1) control-minimizing open-loop optimal, 2) linear-quadratic optimal regulator, 3) time-optimal, 4) real-time optimal, and 5) real-time optimal with singular switching. Methods 3, 4, and 5 involve no feedback control gains tuning.
2. Direct comparison of the efficacy of each of the five methods listed in item #1 controlling *linear* double-integrator plants,

where comparison is made using state accuracy, rate accuracy, control (fuel) usage, and computational runtime (as a manifestation of computational burden).

3. Direct comparison of the efficacy of each of the five methods listed in item #1 controlling double-integrator plants, *including nonlinear transport theorem cross-products of motion* induced by measurement in rotating reference frames, where comparison is made using state accuracy, rate accuracy, control (fuel) usage, and computational runtime (as a manifestation of computational burden). In item #3, the *linear control designs are used on the nonlinear plants* to evaluate the error resulting from using linear control designs in the real-world on nonlinear systems.
4. Direct comparison of the efficacy of each of the five methods listed in item #1 controlling double-integrator plants, *including nonlinear transport theorem cross-products of motion* induced by measurement in rotating reference frames, where comparison is made using state accuracy, rate accuracy, control (fuel) usage, and computational runtime (as a manifestation of computational burden). *Unlike item #3 above, nonlinear decoupling control stemming from the solution to the minimum-control optimization problem is introduced* to each control methodology by utilizing the optimal rate trajectories that result from the original open-loop optimization problem that minimizes control effort. This nonlinear control utilizing the constrained optimization problem results (linear control and nonlinear combinations of the optimal trajectories) may be considered the largest contribution to the article.
5. Items #3 and #4 are both repeated to evaluate the deleterious effects on each method of *noisy sensors and random uniformly varying system mass and mass moments of inertia*.

Motivated to develop simple methods that flow from the solution of constrained optimization problems yet do not necessitate numerical solutions leads to arguably, the most interesting proposal: Utilization of analytic solutions to the constrained optimization problem in either a feedforward or feedback sense applied to full nonlinear, coupled guidance and control problems, specifically including the transport theorem coupling cross-products for rotation and translation, respectively. This method is mathematically developed in section 2, resulting in proposals for both feedforward and feedback methods.

Section 2 includes brief derivations of each respective approach as briefly as practicable, while section 3 provides the results of implementing each disparate methodology. Tables of variable definitions and nomenclature have been placed throughout the manuscript: **Tables 1-6**, while **Table 5** articulates necessary methods for repeating the presented work. **Table 7** summarizes Monte Carlos analysis of parameter variations from **Figure 2**, while **Table 8** summarizes percent performance improvements for each of the six evaluated techniques.

2 MATERIALS AND METHODS

Motion (both translational and rotational) is governed by so-called double-integrator dynamics where the integral of the

applied forces vector (inversely scaled by the mass or mass moments, respectively) is the velocity vector and the integral of the velocity vector (translational or rotational) is the displacement vector. Each vector is relative to an inertial (non-rotating) reference frame, while the expression of the vectors in the coordinates of the basis vectors of rotating reference frames necessitates the inclusion of the transport theorem, which articulates the induced motion of the rotating reference frame in cross-products that make the results nonlinear and coupled. The three degrees of rotational motion are coupled to each other by the transport theorem, and the three degrees of translational motion are coupled to each other as well. Furthermore, the three degrees of translation are coupled nonlinearly to the three degrees of rotation, particularly through the angular velocity vector. Especially since this nonlinear coupling is a foremost challenge that is often deemed insurmountable by analytic methods, the foremost subsections of this part of the manuscript begin so. The Materials and Methods section of the manuscript is described with sufficient details to allow readers to replicate and build on the published results.

2.1 Double-Integrator Based Plant Equations

Eq. 1 illustrates the fundamental relationships of both translational and rotational motion may be expressed as so-called double-integrators, meant to mean the twice integration of the applied force or torques produces the respective translational or rotational displacement.

$$\bar{F} = \bar{m}\bar{a} = \bar{m}\ddot{\bar{x}} = \bar{m}\frac{d^2\bar{x}}{dt^2} \leftrightarrow \bar{T} = \bar{J}\bar{\alpha} = \bar{J}\ddot{\bar{\omega}} = \bar{J}\ddot{\bar{\theta}} = \bar{J}\frac{d^2\bar{\theta}}{dt^2}, \quad (1)$$

Eq. 1 comprises two sets of three equations each for translation and rotation combining for six equations of mechanical motion. For simplicity of expression, states referred to generically as x can represent rotations (θ) with regards to the basic, shared motion described by the double-integrator. The transport theorem described next will generate differences in the governing equations for translation and rotation.

2.2 Transport Theorem Cross-Product Coupled Motion Expressed in Rotating Reference Frames

Attach three mutually perpendicular unit vectors to each frame: the non-rotating inertial frame and body-fixed frame. The meaning of differentiation of vectors when specification is made of differentiation with respect to a specific frame. Both rotational and translational motion relative to the non-rotating reference frame may be represented by double-integrators in accordance with **Eq. 2**.

$$\bar{m}\bar{a}|_{\text{relative}} = \bar{m}\ddot{\bar{x}}|_{\text{relative}} = \bar{m}\frac{d^2\bar{x}}{dt^2}\bigg|_{\text{relative}}, \quad (2)$$

Theorem 1. Transport Theorem. *The derivative of any vector expressed in the coordinates of a rotating reference frame equals*

TABLE 1 | Double-integrator plant ten-run mini-Monte Carlo analysis (faults occurred after first simulation run) executed in MATLAB®/SIMULINK® R2021b (9.11.0.1769968) whose machine precision $\epsilon_{ps} = 2.2204 \times 10^{-16}$.

Method ^a	State error	Rate error	Cost	Runtime
$[T]^{-1}$	Fault	Fault	Fault	Fault
$1 \backslash [T]$	0.000052	-0.0048598	4.0281	1.6221
$\text{inv}[T]$	Fault	Fault	Fault	Fault
$\text{pinv}[T]$	0.001169	-0.003941	4.0281	1.5589
LU Inverse $[T]$	Fault	Fault	Fault	Fault

^aReal-time optimal control (with singular switching).

the sum of the derivative relative to a non-rotating reference frame plus the cross-product of the angular velocity and the vector.

Proof of Theorem 1. The proof of this well-known theorem is provided by Kinematics Handout - MIT OpenCourseWare, 2021. The tedious process may be summarized as follows: 1) express the position vector with respect to the non-rotating inertial reference frame; 2) differentiate to find the expression for velocity remembering to differentiate both the component measurements and the unit vectors; 3) simplify and substitute the defined unit vectors, define the angular velocity in the direction perpendicular to the two-dimensional space of rotation; and 4) substitute the newly defined angular velocity to arrive at the transport theorem as expressed in Eq. 3. □

$$\bar{J} \frac{d\bar{\omega}}{dt} = \bar{J} \frac{d\bar{\omega}}{dt} \Big|_{\text{relative}} + \bar{\omega} \times J\bar{\omega}, \quad (3)$$

The inclusion of theorem 1, despite being very well-known, is purposely performed to emphasize the most novel proposals presented. In particular, Eqs. 2 and 3 are ubiquitously approximated first by Eq. 1 and also often by linearization of Taylor's Series of each equation, respectively.

2.2.1 Euler's Moment Equations of Rotation Expressed in a Rotating Reference Frame

Externally applied torques, \bar{T} change angular momentum $\bar{J} \frac{d\bar{\omega}}{dt}$ permitting the substitution into Eq. 3, resulting in Euler's nonlinear moment equation in Eq. 4. A system is called linear if it has two mathematical properties: homogeneity and additivity in accordance with the principle of superposition. If two or more solutions to an equation or set of equations can be added together so that their sum is also a solution, linearity may be asserted. In other words, two or more states of the system must be added together to create an additional state. Adding two single-channel angles cannot also be a solution without accounting for the presence of the other two channels' states.

An easy way to understand the nonlinear nature of each motion channel induced by the linear cross-product transformation is to recall the mathematical definition: $a \times b = ab|\sin\theta|$, while $f(x + y) = f(x) + f(y)$ is a simple counterexample showing that this function f is not linear: $\sin(\theta_1 + \theta_2) \neq \sin(\theta_1) + \sin(\theta_2)$. An often-confused notion is the fact that the cross-product is a linear transformation, but

nonetheless, each motion channel is evidently nonlinear in the states (evidenced by the presence of multiplicative state pairs and cross-coupled states in each channel. From the perspective of linear algebra, the matrix representation of the cross-product is skew-symmetric and has determinant zero, so it will not always have a solution.

$$\begin{aligned} \bar{T} &= \bar{J} \dot{\bar{\omega}} + \bar{\omega} \times \bar{J} \bar{\omega} \\ \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} &= \begin{bmatrix} J_{xx}\dot{\omega}_x + J_{xy}\dot{\omega}_y + J_{xz}\dot{\omega}_z - J_{xy}\omega_x\omega_z - J_{yz}\omega_y\omega_z - J_{yz}\omega_x^2 + J_{xz}\omega_x\omega_y + J_{zz}\omega_z\omega_y + J_{yz}\omega_y^2 \\ J_{yx}\dot{\omega}_x + J_{yy}\dot{\omega}_y + J_{yz}\dot{\omega}_z - J_{yz}\omega_x\omega_y - J_{zz}\omega_x\omega_z - J_{xz}\omega_x^2 + J_{xx}\omega_x\omega_z + J_{xy}\omega_z\omega_y + J_{xz}\omega_z^2 \\ J_{zx}\dot{\omega}_x + J_{zy}\dot{\omega}_y + J_{zz}\dot{\omega}_z - J_{zx}\omega_x\omega_y - J_{xz}\omega_y\omega_z - J_{xy}\omega_y^2 + J_{yy}\omega_x\omega_y + J_{yz}\omega_z\omega_x + J_{xy}\omega_x^2 \end{bmatrix}, \quad (4) \end{aligned}$$

It should be noted the dominant double-integrator dynamics are embodied in $J\dot{\omega}$ in Eq. 4, while the additional accelerations due to the transport theorem are embodied in the coupling cross-product term $\bar{\omega} \times \bar{J}\bar{\omega}$. Control designs based on the double-integrator dynamics alone are hypothesized to have less efficacy than proposed techniques that utilize optimization and the transport theorem terms.

2.2.2 Newton's Equations of Translation Expressed in a Rotating Reference Frame

Performing similar expression of translational motion in non-rotating inertial frames as just performed in section 2.2.1 for rotational motion leads to Eq. 5 for Newton's equations of translational motion.

$$\bar{F} = \underbrace{\bar{m}\bar{a}}_{\text{Relative}} + \underbrace{\bar{m} \frac{d\bar{\omega}}{dt}}_{\text{Euler}} \times \bar{r} + \underbrace{2\bar{m}\bar{\omega} \times \bar{v}}_{\text{Coriolis}} + \underbrace{\bar{m}\bar{\omega} \times \bar{\omega} \times \bar{r}}_{\text{Centrifugal}}, \quad (5)$$

Notice the dominant double-integrator dynamics relative to the rotating reference frame in Eq. 5 are embodied in $m\bar{a}$, while the additional accelerations due to the transport theorem are embodied in the coupling cross-product terms: Euler ($\bar{m} \frac{d\bar{\omega}}{dt} \times \bar{r}$), Coriolis ($2\bar{m}\bar{\omega} \times \bar{v}$), and centrifugal ($\bar{m}\bar{\omega} \times \bar{\omega} \times \bar{r}$). Control designs based on the double-integrator dynamics alone are hypothesized to have less efficacy than proposed techniques that utilize optimization and the transport theorem terms.

2.2.3 Impacts on Control Design

Neglecting the cross-products of acceleration resulting from the transport theorem reduces both Eqs. 4 and 5 to the double-integrators of Eqs. 1 and 2. The goal of this research is to develop controls (for applied forces \bar{F} and applied torques \bar{T}) that account for the nonlinear, coupling cross-products produced by the application of the transport theorem. Typically, nonlinearities like those presented in Eqs. 4 and 5 caused by transport theorem are simplified by assumption, neglected altogether, or linearized to permit linear control design. Subsequently, the linear controllers are applied to the nonlinear systems and augmented as necessary to improve performance. Instead, the optimal trajectories that result from the solution of constrained optimization problems (for translation and rotation, respectively) are combined to form new nonlinear controls. The exact form of the nonlinear equation is used to form the new nonlinear control components where the motion states are taken from the solution to the constrained optimization problem.

TABLE 2 | Proximal variable definitions.

Variable	Definition	Variable	Definition
\bar{x}_d	Desired state trajectory	$\bar{\xi}$	Critical damping ratio
\bar{K}_p	Proportional gain	$\bar{\omega}_n$	Natural frequency
\bar{K}_v	Velocity gain	t_s	Settling time

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined master table of definitions is included in the **Supplementary Appendix**.

2.3 Classical Position Plus Velocity (P + V) Feedback Control

Proportional plus velocity control (Chai et al., 2019) utilizes proportional control by forming a state error scaled by a proportional gain adding a negative gained value of velocity (translational or rotational), as elaborated in Eq. 5. The velocity channel is not a differentiated version of the position or angle channel, as is the case with classical cascaded control topologies of PD, PI, and PID types (proportional plus derivative, proportional plus integral, and proportional plus integral plus derivative, respectively).

$$\begin{aligned} \bar{J}\ddot{\bar{x}} + \bar{K}_v\dot{\bar{x}} + \bar{K}_p\bar{x} &= \bar{K}_p\bar{x}_d \leftrightarrow \bar{u} = \bar{K}_p(\bar{x}_d - \bar{x}) \\ -\bar{K}_v\dot{\bar{x}} &= \frac{\bar{K}_p}{\bar{J}s^2 + \bar{K}_v s + \bar{K}_p} \rightarrow \text{C.E.: } s^2 + \bar{K}_v s + \bar{K}_p \Big|_{I=1} \\ &= s^2 + 2\xi\bar{\omega}_n s + \bar{\omega}_n^2. \end{aligned} \quad (6)$$

Gains were tuned for performance specification by equating the ubiquitous closed-loop system Eq. 5 to the performance specified, where C.E. annotates the characteristic equation. The desired rise time established the system natural frequency per $t_r = \frac{1.8}{\bar{\omega}_n}$, where $\bar{\omega}_n \approx \bar{\omega}_b$ is the desired control bandwidth; therefore, $\bar{\omega}_n = \frac{1.8}{t_r} \rightarrow \bar{K}_p = \bar{\omega}_n^2$. Settling time: oscillation stabilize within 2–5% percent of steady state $t_s = \frac{4.6}{\xi\bar{\omega}_n} \rightarrow \bar{\xi} = \frac{4.6}{t_s\bar{\omega}_n} \rightarrow \bar{K}_v = 2\xi\bar{\omega}_n$.

Elimination of differentiation in the derivative channel often bestows relative advantage in tracking desired velocity trajectories. Another approach is the optimize gain selection, and this alternative approach is called the linear quadratic regulator.

2.4 Linear-Quadratic Optimal Regulator of Proportional Derivative Type (Murray, 2010)

Eqs. 3 and 4, representing the full, nonlinear, coupled equations of motion in six degrees, may be linearized and be expressed in the form displayed in Eq. 6. This linearization is the basis for the word “linear” in the LQR title. The word “quadratic” refers to selecting gains K that minimize a quadratic cost function displayed in Eq. 7. The LQR solution (Kwakernaak and Sivan, 1972; NASA, 2021a) only bestows optimal solutions for control gains of the form Eq. 8 that minimizes the quadratic cost simultaneously satisfying the (linearized) dynamic constraints displayed in Eq. 5.

TABLE 3 | Proximal variable definitions.

Variable	Definition	Variable	Definition
\bar{A}	State transition matrix	J	Cost function
\bar{B}	Control matrix	t_f	Final time
\bar{K}	Gain matrix	∞	Infinity
\bar{Q}	State weighting matrix	sgn	Signum function
\bar{R}	Control weighting matrix	$p(t)$	Parameters (co-states)
\bar{P}	Covariance matrix	b_i	Control coefficients

$$\dot{\bar{x}} = \bar{A}\bar{x} + \bar{B}\bar{u}, \quad (7)$$

$$J = \int_0^\infty (\bar{x}^T \bar{Q} \bar{x} + \bar{u}^T \bar{R} \bar{u}) dt, \quad (8)$$

$$\bar{u} = -\bar{K}\bar{x}. \quad (9)$$

The control designer may select the state weighting matrix \bar{Q} and the control weighting matrix \bar{R} to penalize the state errors and the control effort, respectively. In section 3, equally weighted identity matrices were chosen for both \bar{Q} and \bar{R} . This choice facilitates a multi-faceted comparison in section 3 that does not solely focus on tracking errors or costs. The gains \bar{K} are found using Eq. 9, where the matrix \bar{P} is first found by solving the algebraic relation in Eq. 10, often referred to as a Riccati equation which is most often solved iteratively by a computer (the MATLAB®/SIMULINK® *lqr* command).

$$\bar{K} = \bar{R}^{-1} (\bar{B}^T \bar{P}) \quad (10)$$

$$\bar{A}^T \bar{P} + \bar{P} \bar{A} - \bar{P} \bar{B} \bar{R}^{-1} \bar{B}^T \bar{P} + \bar{Q} = \bar{0} \quad (11)$$

2.5 Time-Optimal Control (Murray, 2010)

Minimizing a non-quadratic cost function comprised of only the final time (as displayed in Eqs. 11 and 12) constrained with the linearized dynamics of Eq. 6 with costate parameters $p(t)$ used in the Hamiltonian problem formulation leads to time-optimal control (Flugge-Lotz, 1953; Pontryagin et al., 1962; Boltyanskii, 1971; Sands et al., 2009; Sands and Ghadawala, 2011; Duprez et al., 2017; Heidlauf and Cooper, 2017; Baker et al., 2018; Sands, 2019; Smeresky et al., 2020; Arguchintsev and Poplevko, 2021; Malecek, 2021; Srochko et al., 2021).

$$J = \int_0^\infty t_f dt, \quad (12)$$

$$\bar{u} = \text{sgn}(\langle p(t), b_i \rangle) = \begin{cases} 1 & \text{if } \langle p(t), b_i \rangle > 0 \\ -1 & \text{if } \langle p(t), b_i \rangle < 0 \\ 0 & \text{if } \langle p(t), b_i \rangle = 0 \end{cases} \quad (13)$$

Simulation subsystems depicted in the appendix execute a *bang-bang control* where maximal application of control is normalized to unity such that desired unity state and unity

TABLE 4 | Proximal variable definitions.

Variable	Definition	Variable	Definition
J	Cost function	\bar{u}^*	Optimal control
dt	Differential time	$\dot{\bar{v}}^*$	Optimal (angular) acceleration
t	Time	\bar{v}^*	Optimal (angular) velocity
\bar{u}	Control	\bar{x}^*	Optimal (angular) position
$\bar{x}(t)$	Current position	$\bar{a}, \bar{b}, \bar{c}, \bar{d}$	Integration constants
$\bar{v}(t)$	Current velocity	$\hat{a}, \hat{b}, \hat{c}, \hat{d}$	Integration constant estimates

time is achieved to aid comparison to the other optimization approaches. One key feature of *bang-bang* control is the neglecting of the rate end condition leading to a so-called *bang-off-bang* control, which is not treated here.

2.6 Open Loop Minimum-Control Optimization (Pontryagin et al., 1962; Ross, 2015)

Minimizing only the control effort alone (not the state errors) (Sands, 2019) in accordance with Eq. 13 constrained by the double-integrator dynamics of Eq. 2 for specified initial and final conditions permits the solution of a two-point boundary value problem producing optimal control, acceleration, rate, and state profiles displayed in Eq. 14, respectively. Normalization for unity masses or mass moments is included, thus, control and acceleration equations are identical, where non-normalized control may be expressed by scaling the control equation by the masses or mass moments, respectively.

By specifying quiescent initial conditions and using variable scaling and balancing to normalize the final position coordinate to unity, the constants in Eq. 14 may be solved, resulting in Eq. 15, where $a = -12$, $b = 6$, and $c = d = 0$. It should be noted that states are not penalized in the cost function, instead only solution forms that satisfy the boundary values are produced by the two-point boundary value problem from the initial point $(x(0), v(0)) = (0, 0)$ to the final point $(x(1), v(1)) = (1, 0)$, thus, there is no need to solve an algebraic Riccati equation to produce the optimal control, where an additional benefit of this optimization approach includes the production of optimal state trajectories that will prove useful to decouple the nonlinear coupling effects of the transport theorem described in section 2.2. Scaling and balancing must be performed to normalize the initial and final conditions to zero and unity, and the operations are explained in section 2.11. The solution to the constrained optimization problem listed in Eqs. 2 and 13 was solved analytically and presented recently by Sands (2021) for virtual sensing, and that solution is presented here in Eqs. 14 and 15. The mathematical development is intentional since 1) the development is well-articulated by Sands (2021) and 2) increased focus on the utilization of these results toward nonlinear equations of motion (presented in Section 2.9).

$$J = \frac{1}{2} \int_0^\infty (\bar{u}^T \bar{u}) dt, \quad (14)$$

$$\begin{aligned} \bar{u}^* &= \bar{a}t + \bar{b}, \dot{\bar{v}}^* = \bar{a}t + \bar{b}, \bar{v}^* = \frac{1}{2} \bar{a}t^2 + \bar{b}t + \bar{c}, \bar{x}^* \\ &= \frac{1}{6} \bar{a}t^3 + \frac{1}{2} \bar{b}t^2 + \bar{c}t + \bar{d}, \end{aligned} \quad (15)$$

$$\bar{u}^* = -12t + \bar{c}, \dot{\bar{v}}^* = -12t + \bar{c}, \bar{v}^* = -6t^2 + \bar{c}t, \bar{x}^* = -2t^3 + 3t^2. \quad (16)$$

The open-loop optimal solution embodied in Eqs. 14 and 15 may be updated in real-time using state feedback resulting in real-time optimal control presented in Section 2.7. These optimal states in Eq. 15 are used to form nonlinear controls in Section 2.9.

2.7 Real-Time Optimal Control

A corollary is to the open-loop minimum-control optimization in section 2.6 augments the approach with feedback while maintaining the remaining portions of the problem approach. The solution for the constants in between Eqs. 14 and 15 may be accomplished in real-time using feedback but *asserting the current position and velocities (translational and rotational) are the initialization points of a new two-point boundary value problem.*

Eq. (16) may be written in a matrix-vector form as Eq. 17, permitting real-time solution for the integration constants in the vector by inverting the matrix and pre-multiplying both sides of the equation as depicted in (17). Notice the form of the control derived in Eq. 17 is the same as Eq. 14 in section 2.6, where the constants in the optimal solutions are solved in real-time.

$$\begin{aligned} v^* &= \frac{1}{2} at^2 + bt + c, x^* = \frac{1}{6} at^3 + \frac{1}{2} bt^2 + vt(t_f) = \frac{1}{2} a + b \\ &= 0, x^*(t_f) = \frac{1}{6} a + \frac{1}{2} b = 1, \end{aligned} \quad (17)$$

$$\begin{aligned} \underbrace{\begin{bmatrix} \frac{t_0^2}{6} & \frac{t_0^2}{2} & t_0 & 1 \\ \frac{t_0^2}{2} & t_0 & 1 & 0 \\ \frac{1}{6} & \frac{1}{2} & 1 & 1 \\ \frac{1}{2} & 1 & 1 & 0 \end{bmatrix}}_{[T]} \underbrace{\begin{Bmatrix} a \\ b \\ c \\ d \end{Bmatrix}}_{\{p\}} &= \underbrace{\begin{Bmatrix} \bar{x}_0 \\ \bar{v}_0 \\ 1 \\ 0 \end{Bmatrix}}_{\{b\}} \rightarrow \underbrace{\begin{Bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \\ \hat{d} \end{Bmatrix}}_{\{p\}} \\ &= \begin{bmatrix} \frac{t_0^2}{6} & \frac{t_0^2}{2} & t_0 & 1 \\ \frac{t_0^2}{2} & t_0 & 1 & 0 \\ \frac{1}{6} & \frac{1}{2} & 1 & 1 \\ \frac{1}{2} & 1 & 1 & 0 \end{bmatrix}^{-1} \begin{Bmatrix} \bar{x}(t) \\ \bar{v}(t) \\ 1 \\ 0 \end{Bmatrix} \text{ and } \bar{u}^* \equiv \hat{a}t + \hat{b}. \end{aligned} \quad (18)$$

TABLE 5 | Proximal variable definitions.

Variable	Definition	Variable	Definition
\bar{F}	Externally applied forces	\bar{T}	Externally applied torques
\bar{m}	Mass	\bar{J}	Mass moment of inertia
$\bar{a} = \ddot{\bar{x}}$	Translational acceleration	$\bar{\alpha} = \ddot{\bar{\theta}}$	Rotational acceleration
$\frac{d^2 \bar{x}}{dt^2} = \dot{\bar{v}}$	Translational acceleration	$\frac{d^2 \bar{\theta}}{dt^2} = \dot{\bar{\omega}}$	Rotational acceleration
\bar{r}	Radius vector relative to rotating frame	$\bar{\omega} = \dot{\bar{\theta}}$	Rotational velocity
\bar{v}	Velocity vector relative to rotating frame	$\bar{\theta}$	Displacement angle

Such tables are distributed throughout the manuscript to increase the ease of reading, while a combined master table of definitions is included in the **Supplementary Appendix**.

TABLE 6 | Proximal variable definitions.

Variable	Definition	Variable	Definition
$[T]^{-1}$	MATLAB inverse	m	Mass
$1 \setminus [T]$	MATLAB inverse	J	Mass moments
$\text{inv}[T]$	MATLAB inverse	ω	Angular velocity
$\text{pinv}[T]$	MATLAB pseudo-inverse	r	Position vector
$\text{lu}([T])$	MATLAB LU-inverse	v	Translational velocity
$U_{\text{transport}}$	Feedback decoupler	ω^*	Optimal angular velocity
$U_{\text{transport}}^*$	Feedforward (decoupler)	r^*	Optimal position vector
u	Control, the sum of Eq. 18 or Eq. 19 with Eqs 5, 8, 12, 14, or Eq. 15	v^*	Optimal Translational velocity

One key feature of the open-loop solution method using a two-point boundary-value problem is the enforcement of end conditions producing optimal trajectories for state (\bar{x}^*), rate (\bar{v}^*), acceleration (\bar{a}^*), and jerk (\bar{j}^*), in addition to the formulation of an optimal control, u^* . These signals yield the opportunity to formulate decoupling control components to mitigate the transport theorem (illustrated in section 2.9).

2.8 Real-Time Optimal Control With Singular Switching

Highlighting the matrix inverse in **Eq. 17**, the possibility of issues inverting a poorly conditioned or rank-deficient matrix may be addressed by monitoring matrix conditioning or determinant and switching away from the feedback solution when encountering rank-deficient instances in favor of the optimal solution in **Eq. 15**.

2.8.1 Matrix Inverse Formulas

Five disparate methods to invert the $[T]$ matrix were investigated, as listed in **Eq. 18**. Matrix inversion methods already coded in MATLAB/SIMULINK: $[T]^{-1}$, $1 \setminus [T]$, $\text{inv}[T]$, $\text{pinv}[T]$, LU Inverse $[T]$. Each method has specific strengths and weaknesses expressed in state error, rate error, control effort (quadratic cost), and runtime, as displayed in **Table 1**. In several instances, the simulation would fault as a result of encountering matrix singularity.

2.9 Nonlinear Transport Theorem Decoupling (Recall Transport Theorem in Section 2.2)

As mentioned in **Section 2.6**, the desire is to use the results of **Sections 2.7, 2.8** applied to nonlinear dynamics coupled by transport theorem. **Section 2.2** describes the nonlinear coupling effects of measuring motion in coordinates of rotating reference frames extracted for highlighting in **Eq. 18** for translation and rotation, respectively. These effects were neglected when optimizing the double-integrator-based systems of equations or simplified by linearization in other instantiations. Taking advantage of the results in **Sections 2.6, 2.7**, nonlinear decoupling control components may be formulated using the optimal trajectories as displayed in **Eq. 18**, where each component (translation and rotation, respectively) should be added to augment the control in **Eq. 17**.

$$\left. \begin{aligned}
 &\text{Rotation: } u_{\text{nonlinear}} = \bar{\omega} \times \bar{J} \bar{\omega} \\
 &\text{Translation: } u_{\text{nonlinear}} = \bar{m} \frac{d\bar{\omega}}{dt} \times \bar{r} + 2\bar{m}\bar{\omega} \times \bar{v} + \bar{m}\bar{\omega} \times \bar{\omega} \times \bar{r}
 \end{aligned} \right\} \text{Feedback}$$

$$\left. \begin{aligned}
 &\text{Rotation: } u_{\text{nonlinear}} = \bar{\omega}^* \times \bar{J} \bar{\omega}^* \\
 &\text{Translation: } u_{\text{nonlinear}} = \bar{m} \frac{d\bar{\omega}^*}{dt} \times \bar{r}^* + 2\bar{m}\bar{\omega}^* \times \bar{v}^* + \bar{m}\bar{\omega}^* \times \bar{\omega}^* \times \bar{r}^*
 \end{aligned} \right\} \text{Feed forward}$$
(19)

It is proposed to take the nonlinear control components in **Eq. 18** formulated using the optimal trajectories from **Eq. 18** and

augment them with Eq. 17's optimal control solutions to comprise the *total control* for rotation and translation, respectively, displayed in Eq. 19.

$$\begin{aligned}
 &\left. \begin{aligned} \text{Rotation: } u_{total}^* &= \hat{a}t + \hat{b} + \bar{\omega} \times \bar{J}\bar{\omega} \\ \text{Translation: } u_{total}^* &= \hat{a}t + \hat{b} + \bar{m} \frac{d\bar{\omega}}{dt} \times \bar{r} + 2\bar{m}\bar{\omega} \times \bar{v} + \bar{m}\bar{\omega} \times \bar{\omega} \times \bar{r} \end{aligned} \right\} \text{Feedback} \\
 &\left. \begin{aligned} \text{Rotation: } u_{total}^* &= \hat{a}t + \hat{b} + \bar{\omega}^* \times \bar{J}\bar{\omega}^* \\ \text{Translation: } u_{total}^* &= \hat{a}t + \hat{b} + \bar{m} \frac{d\bar{\omega}^*}{dt} \times \bar{r}^* + 2\bar{m}\bar{\omega}^* \times \bar{v}^* + \bar{m}\bar{\omega}^* \times \bar{\omega}^* \times \bar{r}^* \end{aligned} \right\} \text{Feedforward}
 \end{aligned} \quad (20)$$

The distinction between feedforward and feedback is determined by the chosen manner of decoupling the nonlinear transport theorem. One option is feedback decoupling, where feedback states are combined (for example, for rotation) as $\bar{\omega} \times \bar{J}\bar{\omega}$ and added to the optimal control to form a new nonlinear control. On the other hand, feedforward decoupling (used in section 3) combines the optimal equations of state from Eqs. 15 and 16, respectively, combined as $\bar{\omega}^* \times \bar{J}\bar{\omega}^*$ and augments the optimal controls with this new nonlinear control. The efficacy of this latter suggestion is evaluated in section 3.

Section 3 validates the proposed developments culminating in the combination of real-time optimal control (Eq. 17) together with transport theorem decoupling in Eq. 19, specifically for rotation using feedforward: $u_{total}^* = \hat{a}t + \hat{b} + \bar{\omega}^* \times \bar{J}\bar{\omega}^*$. After evaluating initial efficacy, effectiveness against realistically varying systems with noise is investigated.

2.10 Noisy Mixed Sensors and Parameter Variations

With the inclusion of feedback, noise must be accounted for in the feedback signals (state and rate only here) in the form of random numbers with a standard deviation of 0.01 (1% of the final state value when states are scaled and balanced to unity). In addition, mass and mass moments are assumed to be unknown or not precisely known; therefore, mass and moments were allowed to vary randomly (uniformly) ten percent heavier and lighter. The resulting scatterplots are presented in section 3.

2.11 Scaling and Balancing

Poorly conditioned problems are those requiring simultaneous mathematical operations on very large and small numbers. A common mitigation strategy is to scale and balance the variables transforming equations to nominally remain of the same order. Scaling problems by common, well-known values permits single developments to be broadly applied to a wide range of state spaces not initially intended. Normalizing time per Eq. 20 restricts simulation time to vary between zero and unity. Scaling mass and mass moments of inertia matrices by their nominal values per Eqs. 21 and 22, respectively, keep their values roughly on the order of unity. Generic displacements (translation or rotation) are normalized in accordance with Eq. 23, where $\{r\}$ could indicate either general translational or rotational displacement.

$$t \equiv \frac{\bar{t}}{t_f}, \quad (21)$$

$$\bar{m} \equiv \frac{[m]}{[m]_{nominal}}, \quad (22)$$

$$\bar{J} \equiv \frac{J}{J_{nominal}}, \quad (23)$$

$$\bar{r} \equiv \frac{\{r\}}{\{r\}_f} \text{ for translation and } \bar{\theta} \equiv \frac{\{\theta\}}{\{\theta\}_f} \text{ for rotation.} \quad (24)$$

3 RESULTS

Following the brief introduction to each control technique presented earlier in section 2, section 3 displays the results of individual simulations in addition to the Monte Carlo investigation of ten-thousand simulations. Section 3.1 begins with commonly simplifying assumptions of control design using dominant, double-integrator dynamics with no transport theorem, where the control $\bar{u}^* \equiv \hat{a}t + \hat{b}$ is applied to the same idealized system equations. The use of idealized results provides interesting measures of performance under ideal circumstances subject to mathematical optimization. Another interesting artifact is the immediately obvious differences in the responses to disparate control techniques.

Next, in section 3.2, the performance of controllers designed using simplified double-integrators $\bar{u}^* \equiv \hat{a}t + \hat{b}$ was applied to more realistic plant equations with transport theorem. Then, in section 3.3, nonlinear control designs $u_{total}^* = \hat{a}t + \hat{b} + \bar{\omega}^* \times \bar{J}\bar{\omega}^*$ are introduced, and comparisons are made applied to nonlinear plants, including transport theorem. Lastly, in section 3.4, random uniformly varying inertia was studied with random noise added to sensor data for both state and rate, and the comparisons were repeated in ten-thousand simulations. These final simulations all utilized nonlinear control designs based on various optimization methods, and the results were applied to nonlinear, coupled system equations, including the transport theorem, where controls were tailored specifically for the transport theorem in the recommended application of optimization (real-time optimal control with singular switching and transport theorem decoupling). All simulations were executed in MATLAB/SIMULINK® R2021b (9.11.0.1769968), whose machine precision was $eps = 2.2204 \times 10^{-16}$.

A new presentation style is offered to increase the ease of reading and contemplation of the results. Quantitative figures of merit are presented in tables inserted as sub-figures immediately proximal to corresponding data plots presenting qualitative results.

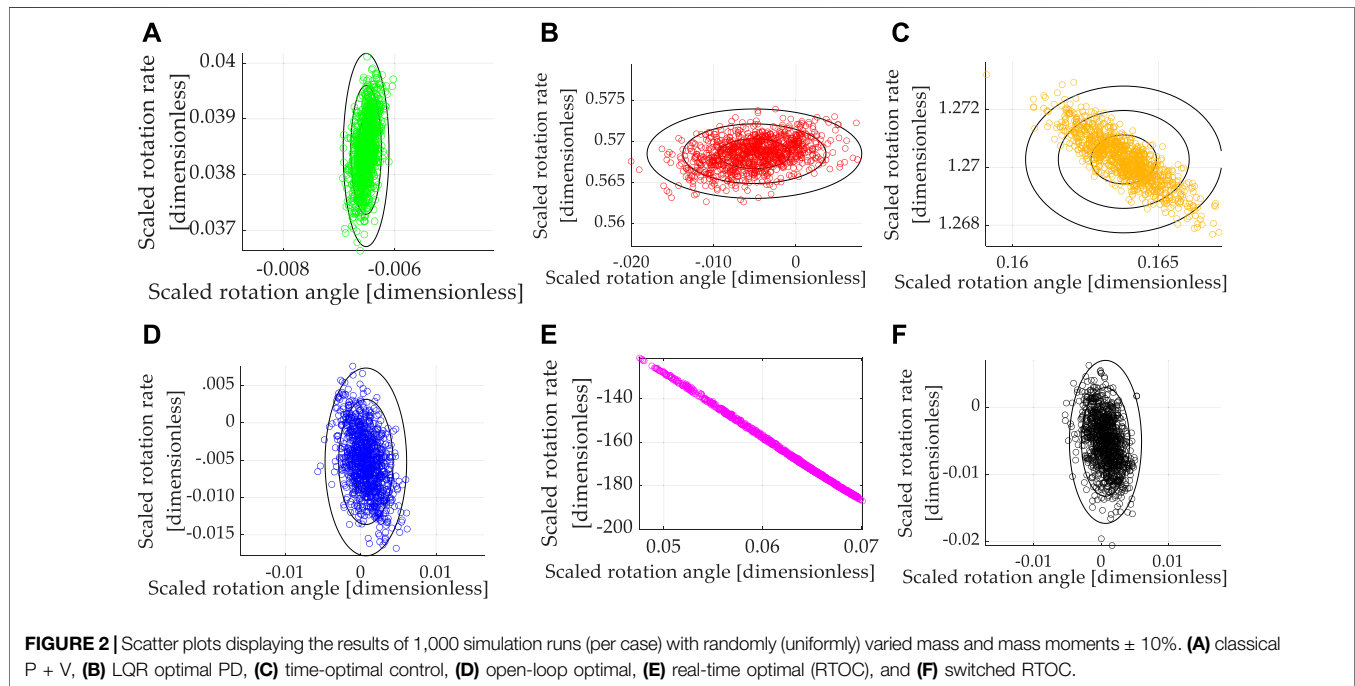
3.1 Ideal, Linear Double Integrator System Equations

Double-integrator equations expressing relationships between displacement, displacement rate, and acceleration are canonical

TABLE 7 | One-thousand-run (respectively) Monte Carlo analysis.

Method	State error	Rate error	Cost	Runtime
Classical $p + V$	-0.0065157	0.038442	25.3181	1.2342
LQR Optimal PD	-0.0050504	0.56851	75.5278	1.23
Time-optimal control	0.16381	1.2703	1.3643	1.3055
Open loop optimal ^a	0.00069197	-0.0052251	4.0281	1.2335
Real-time optimal (RTOC) ^a	0.062551	-165.1258	40,959.5421	1.2818
Switched RTOC ^a	0.00066117	-0.0051746	4.0281	1.1068

^aReal-time optimal control (with and without switching) and open-loop optimal control are visually indistinguishable from one another in the graphic depiction.



relationships used to describe the movement of mass. The relationships are linear, allowing easy control design using classical methods, which predominantly use linear systems methods to design controllers for any equations (including linearizing any nonlinear equations). Each control design technique introduced in section 2 was sequentially used to control the linear, double-integrator system equations, and the results are presented in **Figure 3**, where **Table 9** contains quantitative results corresponding to the qualitative results presented in the multi-plots.

The baseline approach (classical proportional plus velocity, or so-called “ $P + V$ ” control) tuned to performance specification exhibits better accuracy and lower costs than linear-quadratic optimal regulators of the proportional, derivative (PD) type, but the $P + V$ controller has the highest computational burden as indicated by computational runtime. The embedded differentiation of the noisy feedback signal in the rate channel would logically explain the relatively lower performance of the LQR tracking. Time-optimal (bang-bang) control achieved machine precision state tracking accuracy with the largest

rate tracking error of the controllers investigated. Such performance is validated by the instinct that time-optimal control is mathematically designed to achieve the desired state in the shortest time but is not structured to simultaneously achieve rate tracking in minimal time. The cost was the lowest of the controllers investigated, indicating the benefits of not requiring simultaneous rate tracking. Computational runtime was the second largest.

Open-loop optimization calculates the minimum control effort that simultaneously meets state and rate endpoint conditions, and accordingly, both state and rate endpoints are achieved to machine precision, while the computational burden is modest compared to low and high cases. Real-time optimization solves the open-loop optimal control problem in real-time using ideal sensor feedback of state and rate but involves a matrix inverse. Rate and state errors (particularly) are quite small, but machine precision tracking is not achieved. Part of the cause of tracking errors is the inversion of the rank-deficient matrix as the final time is approached. Seeking to ameliorate the issue, switched real-time optimal control is presented where the matrix condition

TABLE 8 | Percentage change in performance in one-thousand-run (respectively) Monte Carlo analysis: double-integrator plant (with transport theorem) with control design based off double-integrator with transport theorem and noisy, mixed sensors (state and rate).

Method	State error	Rate error	Cost	Runtime
Classical $p + V$	—	—	—	—
LQR Optimal PD	−22%	1,379%	198%	0%
Time-optimal control	2,614%	3,204%	−95%	6%
Open loop optimal ^a	−111%	−114%	−84%	0%
Real-time optimal (RTOC) ^a	1,060%	429,645%	161,680%	4%
Switched RTOC ^a	−111%	−114%	−84%	−10%

^aReal-time optimal control $\hat{u}_{total}^* = \hat{a}t + \hat{b} + \hat{\omega}^* \times \vec{J}\hat{\omega}^*$ (with and without switching) and open-loop optimal control are visually indistinguishable from one another in the graphic depiction.

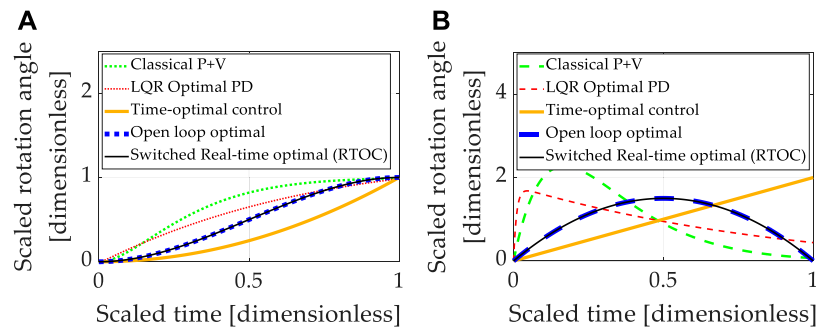


FIGURE 3 | Double-integrator plant (no transport theorem) with control design based on double-integrator. **(A)** Motion states (translation or rotational) normalized to propagate from zero and unity in one normalized second. **(B)** Motion rates (translation or rotational) intended to propagate from zero initial velocity to zero velocity at the endpoint in one normalized second.

TABLE 9 | Double-integrator plant (no transport theorem) with control design based off double-integrator (no transport theorem): quantitative comparative data corresponding to the qualitative display in **Figures 3A,B**.

Method	State error	Rate error	Cost	Runtime
Classical $p + V$	0.010115	0.066169	28.1671	3.1012
LQR Optimal PD	0.015015	0.43861	76.3418	2.4597
Time-optimal control	eps	2	2	2.9038
Open loop optimal ^a	eps	Eps	6	2.6086
Real-time optimal (RTOC) ^a	-9.1882×10^{-6}	0.019289	6.7656	2.7497
Switched RTOC ^a	eps	Eps	6	2.7281

^aReal-time optimal control $\bar{u}^* \equiv \hat{a}t + \hat{b}$ (with and without switching) and open-loop optimal control are visually indistinguishable from one another in the graphic depiction.

is used to switch away from real-time optimal control to open-loop optimal control during timesteps when matrix inversion becomes poorly conditioned. Machine precision tracking is attained, open-loop optimally low costs are reached, and the computational burden is slightly elevated compared to the best case investigated.

Summarizing the results so far, real-time optimal control (designed only to minimize control effort) with singular switching to counter the deleterious effects of poor matrix conditioning achieves the best simultaneous state and rate error (machine precision) with costs matching the open-loop minimal and average computational burden. Unfortunately, these results are achievable only in idealized circumstances of double-integrators. Expressing motion in coordinates of rotating reference frames introduced nonlinear coupling described in the next section (3.2).

3.2 Nonlinear Plants With Cross-Product Coupled Transport Theorem With Linear Control Designs

Expressing motion in coordinates of rotating reference frames is referred to as the “transport theorem,” which introduces nonlinear coupling between the six channels of motion that would otherwise have been well-described by simple, linear double-integrators. Very often, linearized system equations or linear assumptions (the double-integrators) are used to design linear controllers. Accordingly, each instance investigated in section 3.1 was applied to nonlinear coupled system equations, including the transport theorem. Increased errors and reduced robustness is generally anticipated since the controllers are not designed to accommodate system nonlinearities specifically. Each control technique introduced in Section 2 was sequentially

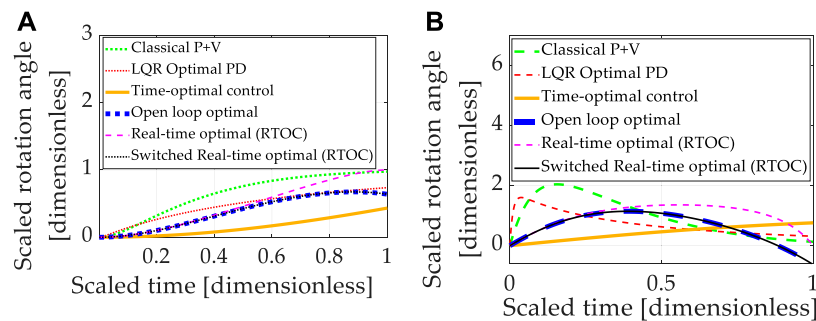


FIGURE 4 | Double-integrator plant (no transport theorem) with control design based on double-integrator. **(A)** Motion states (translation or rotational) normalized to propagate from zero and unity in one normalized second. **(B)** Motion rates (translation or rotational) intended to propagate from zero initial velocity to zero velocity at the endpoint in one normalized second.

simulated, and the results are presented in **Figure 4**, where **Table 10** contains quantitative results corresponding to the qualitative results presented in the multi-plots.

It should be noted that all approaches designed to control double-integrators illustrate degraded performance compared to the idealized case investigated in **Section 3.1**. All methods compared achieved similar costs. The baseline approach (classical proportional plus velocity, or P + V) tuned to performance specification exhibits second-best state accuracy and second-best rate accuracy, while real-time optimal control achieved the lowest state and rate errors but used nearly double the amount of control. No technique achieved machine precision tracking.

Summarizing the results so far, all control techniques are degraded from the idealized case. Real-time optimal control (designed only to minimize control effort) and classical control methods were the most robust, but all three methods utilized substantially more control effort.

3.3 Nonlinear Plants With Cross-Product Coupled Transport Theorem and Nonlinear Control Designs

Double-integrator relationships (implemented identically as done in sections 3.1 and 3.2) are next augmented with feedback decoupling of the transport theorem using state feedback in **Eq. 19**. Each control design technique introduced in section 2 was sequentially used to control the nonlinear, double-integrator system equations including transport theorem, and the results are presented in **Figure 5**, where

Table 11 contains quantitative results corresponding to the qualitative results presented in the multi-plots.

All methods are improved by addition of the nonlinear decoupling control designed for the transport theorem. Time-optimal control performs worst regarding state and rate errors, while cost figures are generally increased for methods that effectively track state and rate. Near machine-precision is achieved by open-loop optimal control and switched, real-time optimal control where both are designed to minimize control effort alone (with no state error representation in the minimized cost function). Computational burdens of all approaches are roughly comparable.

Having initially analyzed idealized systems (the double-integrators), nonlinear coupling was induced by the transport theorem with significantly degraded performance using the controllers designed for linear systems. Adding nonlinear control components designed specifically to decouple the transport theorem in feedback roughly restores nominal performances, but feedback remains ideal (without noise). Section 3.4 adds zero-mean Gaussian noise to both sensor types (state and rate)

3.4 Nonlinear Plants With Cross-Product Coupled Transport Theorem and Nonlinear Control Designs Utilizing Noisy, Mixed-Sensors

Double-integrator equations with nonlinearities induced by transport theorem were controlled by linear control designs

TABLE 10 | Double-integrator plant (with transport theorem) with control design based off double-integrator (without transport theorem): displays the quantitative comparative data corresponding to the qualitative display in **Figure 4**.

Method	State error	Rate error	Cost	Runtime
Classical $p + V$	0.024582	0.12803	26.6076	3.4291
LQR Optimal PD	0.26241	0.31372	75.8051	3.4593
Time-optimal control	0.56622	0.76159	0.5	3.4437
Open loop optimal ^a	0.3606	-0.63176	6	3.5315
Real-time optimal (RTOC) ^a	-1.9654×10^{-5}	0.041323	11.2002	3.4579
Switched RTOC ^a	0.3606	-0.63176	6	3.5097

^aReal-time optimal control $u_{total}^* = \hat{a} + \hat{b}$ (with and without switching) and open-loop optimal control are visually indistinguishable from one another in the graphic depiction. Notice the control design did not account for nonlinear transport theorem with $u_{nonlinear} = \bar{\omega}^* \times \bar{J}\bar{\omega}^*$, the resulting open-loop optimal costs were unchanged, but the state and rate errors increased substantially due to the nonlinear plant not being included in control design. **Figure 5** displays the results of including nonlinear control design $u_{nonlinear} = \bar{\omega}^* \times \bar{J}\bar{\omega}^*$.

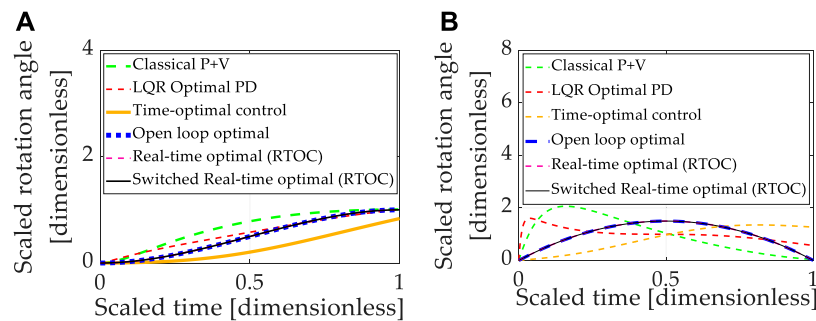


FIGURE 5 | Double-integrator plant (no transport theorem) with control design based on double-integrator. **(A)** Motion states (translation or rotational) normalized to propagate from zero and unity in one normalized second. **(B)** Motion rates (translation or rotational) intended to propagate from zero initial velocity to zero velocity at the endpoint in one normalized second.

TABLE 11 | Double-integrator plant (with transport theorem) with control design based off double-integrator with transport theorem: displays the quantitative comparative data corresponding to the qualitative display in **Figures 5A, B**.

Method	State error	Rate error	Cost	Runtime
Classical $p + V$	0.0078728	0.038016	27.0064	3.5535
LQR Optimal PD	-0.0063144	0.57121	75.7706	3.5738
Time-optimal control	0.16359	1.2712	2.7286	3.629
Open loop optimal ^a	3.0287×10^{-13}	-1.0092×10^{-12}	7.0286	3.4504
Real-time optimal (RTOC) ^a	-9.1882×10^{-6}	0.019288	7.7942	3.6725
Switched RTOC ^a	3.0287×10^{-13}	-1.0092×10^{-12}	7.0286	3.5765

^aUnlike the results of **Figure 4** where only linear, time-varying control designs $u_{total}^* = \hat{a}t + \hat{b}$ were used, in **Figure 5** nonlinear designs for real-time optimal control $u_{total}^* = \hat{a}t + \hat{b} + \hat{\omega}^* \times \hat{J}\hat{\omega}^*$ (with and without switching) were used. Notice the open-loop optimal control is visually indistinguishable from nonlinear, time-varying control designs in the graphic depiction.

augmented with nonlinear feedback decoupling designed specifically for transport theorem. Feedback was provided by simulated mixed state and rate sensors, and Gaussian noise was added. Each control design technique introduced in section 2 was sequentially used to control the linear, double-integrator system equations, and the results are presented in **Figure 6**, where **Table 12** contains quantitative results corresponding to the qualitative results presented in the multi-plots.

The baseline approach (classical proportional plus velocity, or P + V) tuned to performance specification exhibits better rate accuracy and lower costs than linear-quadratic optimal regulators of the proportional, derivative (PD) type, but the P + V controller has relatively inferior state errors compared to LQR. Time-optimal control performs poorly in the face of transport theorem and noisy sensors. Real-time optimal control is severely degraded by noise, particularly with respect to rate errors and control effort.

Open-loop optimization and real-time optimal control with singular switching simultaneously achieve the lowest state and rate endpoint errors, with the lowest costs (that may be claimed to meet endpoint conditions), while the computational burden is modest compared to low and high cases.

3.4.1 Monte Carlos Analysis (6,000 Simulation Runs)

Summarizing the results so far, real-time optimal control (designed only to minimize control effort) with singular switching to counter the deleterious effects of poor matrix conditioning achieves the best simultaneous state and rate

error with costs matching the open-loop minimal and average computational burden in cases where nonlinear feedback decoupling of transport theorem is incorporated and where feedback is provided by noisy state and rate sensors.

4 DISCUSSION

The results are multi-variate, but some general comments are evident regarding the proposed real-time optimal control with singular switching and transport theorem decoupling and its performance compared to a classical benchmark and four other instantiations of optimal control. In the most realistic situations revealed by Monte Carlo analysis with random variations of inertia and state and rate sensor noise, time-optimal bang-bang control achieved respectable rate accuracy with the lowest cost but highest runtime and modest rate tracking errors. Meanwhile, optimal (control minimizing constrained to meet endpoint conditions) open-loop control and its companion real-time optimal control with singular switching achieved the lowest state errors (three orders of magnitude better than time-optimal control) and control effort, while real-time optimal control with singular switching and transport theorem decoupling achieved the lowest rate tracking error. Real-time optimal control without singular switching displayed vulnerability in rate errors and high costs.

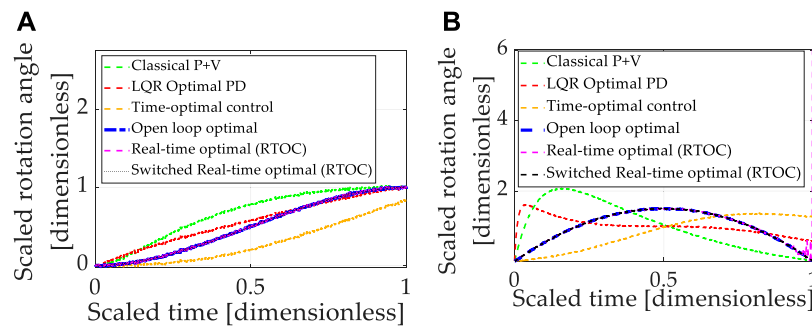


FIGURE 6 | Double-integrator plant (with transport theorem) with control design based on double-integrator (with transport theorem). **(A)** Motion states (translation or rotational) normalized to propagate from zero and unity in one normalized second. **(B)** Motion rates (translation or rotational) intended to propagate from zero initial velocity to zero velocity at the endpoint in one normalized second. Notice that the open-loop optimal (minimum) control effort is increased from 6.0 (controlling double-integrators without transport theorem) to just over 7.0 (controlling double-integrators with transport theorem), manifesting as a 17% increase to account for transport theorem over idealized cases of double-integrators alone.

TABLE 12 | Double-integrator plant (with transport theorem) with control design based off double-integrator with transport theorem and noisy, mixed sensors (state and rate): sub-displays the quantitative comparative data corresponding to the qualitative display in sub-Figure 6.

Method	State error	Rate error	Cost	Runtime
Classical $p + V$	-0.0066807	0.03828	27.0755	2.3988
LQR Optimal PD	-0.0047636	0.56965	76.8806	2.4667
Time-optimal control	0.16546	1.2693	2.7286	2.4837
Open loop optimal ^a	0.0018665	-0.0018665	7.0286	2.5816
Real-time optimal (RTOC) ^a	0.06463	-171.6553	41,436,948	2.469
Switched RTOC ^a	0.0018665	-0.0018665	7.0286	2.6125

^aReal-time optimal control $u_{total}^* = \hat{a}t + \hat{b} + \hat{\omega}^* \times \bar{J}\hat{\omega}^*$ (with and without switching) and open-loop optimal control are visually indistinguishable from one another in the graphic depiction.

Other general conclusions apply to all techniques: designing controls based on simplified plants and then applying them to realistic plants is particularly weak compared to the relatively modern approaches. Arguably, milli-degree accuracy with “low” costs is admirable performance, but the modern methods of control design, including optimality and nonlinear coupling effects (with feedback), achieved, in general, three orders of magnitude superior performance, with the admission that real-time optimal control performed particularly poorly.

Furthermore, well-known lessons from classical control are re-validated in this study. Linear-quadratic regulators are very robust and useful, but suffer from cascaded topologies, particularly in the differentiation of the state feedback to achieve rate feedback, thus the utilization of velocity control was established as the classical baseline (with a requisite demand to purchase and utilize rate sensors).

The proposed instantiation of real-time optimal control with singular switching and nonlinear transport theorem decoupling $u_{total}^* = \hat{a}t + \hat{b} + \hat{\omega}^* \times \bar{J}\hat{\omega}^*$ was the overall top-performing option with the lowest state errors, lowest rate errors, lowest computational burden, and second-lowest control effort (fuel usage).

Lastly, it should be noted that all the control techniques performed very well (naturally, since most of the techniques were formulated to satisfy optimization problems). The

indication of superior performance should not be judged as mandating the proposed technique, especially in instances where operators would be more comfortable with classical techniques and the order of milli-degree accuracy is sufficient.

4.1 Performance Improvement Percentages

The claim was just immediately earlier, validating that real-time optimal control with singular switching and transport theorem decoupling was the overall top-performing option, and this section describes the results validating the claim in generally understandable terms (percent performance improvement comparison). Open-loop (control minimizing optimal control constrained to meet end state and rate) performed very well, while real-time optimal control with singular switching matched the performance and was slightly better in terms of computational burden.

4.2 Future Research

The derivation of optimal trajectories (state, rate, acceleration, and jerk) should prove useful in the implementation of deterministic artificial intelligence (Smeresky et al., 2020), which requires some scheme of autonomous trajectory generation. The current state of the art utilizes sinusoidal trajectory generation schemes, and the optimal trajectories illustrated here should have improved efficacy when used to augment deterministic artificial intelligence.

5 CONCLUSION

Real-time optimal control is proposed to deal with nonlinear mechanics, including transport theorem coupling nonlinearities, where noisy (random) sensors are assumed, and random parameter variation is countered with time-varying solutions to Pontryagin's necessary conditions of optimality. Specifically, the Hamiltonian minimization condition and the adjoint equations produce the form of the control parameterized in terms of time and mass or mass moment of inertia, respectively. Singularity-based switching is proposed to address divergence of the adjoints approaching the final state. Ubiquitous figures of merit are used to compare the proposed methods to benchmark classical and modern optimal control methods: mean state and rate errors, quadratic costs embodying necessary fuel usage, and computational runtime as an avatar of the computational burden. Open-loop optimal control established an intermediate baseline over the benchmark classical control, while the proposed method yielded identical performance improvements in terms of state and rate accuracy and quadratic cost while experimentally illustrating an unexpected ten percent improvement in computational burden.

REFERENCES

- Arguchintsev, A., and Poplevko, V. (2021). An Optimal Control Problem by a Hybrid System of Hyperbolic and Ordinary Differential Equations. *Games* 12, 23. doi:10.3390/g12010023
- Arora, R. (2002). "Reentry Trajectory Optimization: Evolutionary Approach," in *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (Atlanta, Georgia: AIAA Press, 4–6. doi:10.2514/6.2002-5466
- Baker, K., Cooper, M., Heidlauf, P., and Sands, T. (2018). Autonomous Trajectory Generation for Deterministic Artificial Intelligence. *Electr. Electron. Eng.* 8 (3), 59. doi:10.5923/j.eee.20180803.01
- Bertin, E., Brendel, E., Hérissé, B., Sandretto, J., and Chapoutot, A. (2021). "Prospects on Solving an Optimal Control Problem with Bounded Uncertainties on Parameters Using Interval Arithmetics," in *Acta Cybernetica* (Austin, TX: University of Szeged, Institute of Informatics), 1–25. doi:10.14232/actacyb.285798
- Boltyanskii, V. (1971). *Mathematical Methods of Optimal Control*. Holt Rinehart & Winston. Austin, TX: (Translated from Russian).
- Bonnans, J. F., and Festa, A. (2017). Error Estimates for the Euler Discretization of an Optimal Control Problem with First-Order State Constraints. *SIAM J. Numer. Anal.* 55 (2), 445–471. doi:10.1137/140999621
- Botto, M. A., van den Boom, T., and Costa, J. S. d. (2001). "Stability Analysis to Parametric Uncertainty: Extension to the Multivariable Case," in *Nonlin. Con. Year* (London: Springer), 93–102. doi:10.1007/BFb0110210
- Chai, R., Savvaris, A., Tsourdos, A., Chai, S., and Xia, Y. (2019). A Review of Optimization Techniques in Spacecraft Flight Trajectory Design. *Prog. Aerosp. Sci.* 109 (2), 100543. doi:10.1016/j.paerosci.2019.05.003
- Chai, R., Savvaris, A., Tsourdos, A., Chai, S., and Xia, Y. (2018b). Optimal Fuel Consumption Finite-Thrust Orbital Hopping of Aeroassisted Spacecraft. *Aerosp. Sci. Technol.* 75, 172–182. doi:10.1016/j.ast.2017.12.026
- Chai, R., Savvaris, A., Tsourdos, A., Chai, S., and Xia, Y. (2018b). Optimal Tracking Guidance for Aeroassisted Spacecraft Reconnaissance Mission Based on Receding Horizon Control. *IEEE Trans. Aerosp. Electron. Syst.* 54 (4), 1575–1588. doi:10.1109/taes.2018.2798219
- Chen, G., Wan, Z.-M., Xu, M., and Chen, S.-L. (2005). "Genetic Algorithm Optimization of RLV Reentry Trajectory," in *Proceedings of the AIAA/CIRA 13th International Space Planes and Hypersonic Systems and Technologies Conference* (Italy: Capua). doi:10.2514/6.2005-3269

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

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

FUNDING

The APC was funded by the corresponding author.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2022.884669/full#supplementary-material>

- Duprez, M., Morancey, M., and Rossi, F. (2017). "Controllability and Optimal Control of the Transport Equation with a Localized Vector Field," in *2017 25th Mediterranean Conference on Control and Automation (MED)*. IEEE, 74–79. doi:10.1109/MED.2017.7984098
- Ferreres, G., and Fromion, V. (2010). Nonlinear Analysis in the Presence of Parametric Uncertainties. *Int. J. Control* 69, 695–716. doi:10.1080/002071798222631
- Flugge-Lotz, I. (1953). *Discontinuous Automatic Control*. Princeton, USA: Princeton University Press.
- Gao, H., Yang, X., and Shi, P. (2009). Multi-Objective Robust H_∞ Control of Spacecraft Rendezvous. *IEEE Trans. Contr. Syst. Technol.* 17 (4), 794–802. doi:10.1109/TCST.2008.2012166
- Heidlauf, P., and Cooper, M. (2017). "Nonlinear Lyapunov Control Improved by an Extended Least Squares Adaptive Feed Forward Controller and Enhanced Luenberger Observer," in *Proceedings of the International Conference and Exhibition on Mechanical & Aerospace Engineering* (NV, USA: Las Vegas), 2–4.
- Hu, W., Xie, J., Chau, H. W., and Si, B. C. (2015). Evaluation of Parameter Uncertainties in Nonlinear Regression Using Microsoft Excel Spreadsheet. *Environ. Syst. Res.* 4, 4. doi:10.1186/s40068-015-0031-4
- Ivanov, M., Vashchenkov, P., and Kashkovsky, A. (2007). "Numerical Investigation of the EXPERT Reentry Vehicle Aerothermodynamics along the Descent Trajectory," in *Proceedings of the 39th AIAA Thermophysics Conference* (Miami, Florida, 25–28. doi:10.2514/6.2007-4145
- Kelly, R. (2005). "Proportional Control Plus Velocity Feedback and PD Control," in *Control of Robot Manipulators in Joint Space. Advanced Textbooks in Control and Signal Processing* (London: Springer), 141–156. doi:10.1007/1-85233-999-3_8
- Kinematics Handout - MIT OpenCourseWare (2021). Newton's Laws, Vectors, and Reference Frames. Available online at: https://ocw.mit.edu/courses/mechanical-engineering/2-003sc-engineering-dynamics-fall-2011/newton2019s-laws-vectors-and-reference-frames/MIT2_003SCF11Kinematic.pdf (Accessed December 30, 2021).
- Kwakernaak, H., and Sivan, R. (1972). *Linear Optimal Control Systems*. First Edition. Hoboken, USA: Wiley-Interscience.
- Mahmoodabadi, M. J., and Andalib Sahnehsaraei, M. (2021). Parametric Uncertainty Handling of Under-actuated Nonlinear Systems Using an Online Optimal Input-Output Feedback Linearization Controller. *Syst. Sci. Control Eng.* 9 (1), 209–218. doi:10.1080/21642583.2021.1891993
- Malecek, A. (2021). Robert Lorenz, Pioneer in Controls Engineering, Passes Away. Available online at: <https://www.engr.wisc.edu/news/robert-lorenz-pioneer-controls-engineering-passes-away/> (Accessed February 23, 2021).

- Meda-Campana, J. A. (2018). On the Estimation and Control of Nonlinear Systems with Parametric Uncertainties and Noisy Outputs. *IEEE Access* 6, 31968–31973. doi:10.1109/ACCESS.2018.2846483
- Murray, R. M. (2010). “CalTech Notes,” in *Chapter 2 Optimization-Based Control*. Available online at: https://www.cds.caltech.edu/~murray/books/AM08/pdf/obc-optimal_04Jan10.pdf (Accessed December 30, 2021).
- NASA (2021b). NASA Ejects Nanosatellite from Microsatellite in Space. Available online at: https://www.nasa.gov/mission_pages/smallsats/fastosat/10-162.html (Accessed February 23, 2021).
- NASA (2021c). NASA Image Use Policy. Available online at: <https://gpm.nasa.gov/image-use-policy> (Accessed December 23, 2021).
- NASA (2021a). NASA’s FASTSAT Microsatellite Readied to Share Ride to Space. Available online at: https://www.nasa.gov/mission_pages/smallsats/fastosat/10-152.html (Accessed February 23, 2021).
- Pontani, M., and Conway, B. A. (2013). Optimal Finite-Thrust Rendezvous Trajectories Found via Particle Swarm Algorithm. *J. Spacecr. Rockets* 50 (6), 1222–1234. doi:10.2514/1.A32402
- Pontryagin, L., Boltayanskii, V., Gamkrelidze, R., and Mishchenko, E. (1962). *The Mathematical Theory of Optimal Processes*. New York: Wiley-Interscience. (Translated from Russian).
- Rao, A. (2009). A Survey of Numerical Methods for Optimal Control. *Adv. Astronautical Sci.* 135 (1), 497–528.
- Rao, A. V., Tang, S., and Hallman, W. P. (2002). Numerical Optimization Study of Multiple-Pass Aeroassisted Orbital Transfer. *Optim. Control Appl. Meth.* 23 (4), 215–238. doi:10.1002/oca.711
- Ross, I. M., and Karpenko, M. (2012). A Review of Pseudospectral Optimal Control: From Theory to Flight. *Annu. Rev. Control* 36 (2), 182–197. doi:10.1016/j.arcontrol.2012.09.002
- Ross, M. (2015). *A Primer on Pontryagin’s Principle in Optimal Control*. CA: Collegiate Publishers.
- Sagliano, M., Mooij, E., and Theil, S. (2017). Onboard Trajectory Generation for Entry Vehicles via Adaptive Multivariate Pseudospectral Interpolation. *J. Guid. Control, Dyn.* 40 (2), 466–476. doi:10.2514/1.G001817
- Sands, T., Kim, J. J., and Agrawal, B. N. (2009). “Improved Hamiltonian Adaptive Control of Spacecraft,” in Proceedings of the IEEE Aerospace Conference (MT, USA: Big Sky). doi:10.1109/AERO.2009.4839565
- Sands, T. A. (2011). “Physics-Based Control Methods,” in *Advances in Spacecraft Systems and Orbit Determination*. Editor R. Ghadawala (London, U.K.: IntechOpen). doi:10.5772/36128
- Sands, T. (2019). Comparison and Interpretation Methods for Predictive Control of Mechanics. *Algorithms* 12 (11), 232. doi:10.3390/a12110232
- Sands, T. (2021). Virtual Sensing of Motion Using Pontryagin’s Treatment of Hamiltonian Systems. *Sensors* 21 (13), 4603. doi:10.3390/s21134603
- Smeresky, B., Rizzo, A., and Sands, T. (2020). Optimal Learning and Self-Awareness versus PDI. *Algorithms* 13 (1), 23. doi:10.3390/a13010023
- Srochko, V. A., Antonik, V. G., and Aksenyushkina, E. V. (2021). The Method of Parameterization in the Quadratic Optimal Control Problem. *J. Phys. Conf. Ser.* 1847, 012023. doi:10.1088/1742-6596/1847/1/012023
- Sweetha, S., Sakthivel, R., Almakhlis, D. J., and Priyanka, S. (2022). Non-Fragile Fault-Tolerant Control Design for Fractional-Order Nonlinear Systems with Distributed Delays and Fractional Parametric Uncertainties. *IEEE Access* 10, 19997–20007. doi:10.1109/ACCESS.2022.3150477
- Taghieh, A., and Shafiei, M. H. (2021a). Observer-based Robust Model Predictive Control of Switched Nonlinear Systems with Time Delay and Parametric Uncertainties. *J. Vib. Control* 27 (17), 1939–1955. doi:10.1177/1077546320950523
- Taghieh, A., and Shafiei, M. H. (2021b). Static Output Feedback Control of Switched Nonlinear Systems with Time-Varying Delay and Parametric Uncertainties under Asynchronous Switching. *Trans. Inst. Meas. Control* 43 (5), 1156–1167. doi:10.1177/0142331220969056
- Tian, B., Fan, W., Su, R., and Zong, Q. (2015). Real-time Trajectory and Attitude Coordination Control for Reusable Launch Vehicle in Reentry Phase. *IEEE Trans. Ind. Electron.* 62 (3), 1639–1650. doi:10.1109/TIE.2014.2341553
- Wang, Y., Ying, Z. G., and Zhu, W. Q. (2009). Robustness of Non-linear Stochastic Optimal Control for Quasi-Hamiltonian Systems with Parametric Uncertainty. *Int. J. Syst. Sci.* 40 (12), 1217–1227. doi:10.1080/00207170902989296
- Wei, L., McCloy, R., and Bao, J. (2022). Adaptive Contraction-Based Control of Uncertain Nonlinear Processes Using Neural Networks. *arXiv*:2201.12816v1 [eess.SY] Accepted for presentation at 13th IFAC Symposium on Dynamics and Control of Process Systems. doi:10.48550/arXiv.2201.12816
- Willems, J. (1971). Least Squares Stationary Optimal Control and the Algebraic Riccati Equation. *IEEE Trans. Autom. Contr.* 16 (6), 621–634. doi:10.1109/tac.1971.1099831
- Windhorst, R., Ardema, M., and Bowles, J. (1997). “Minimum Heating Reentry Trajectories for Advanced Hypersonic Launch Vehicles,” in Proceedings of the Guidance, Navigation, and Control Conference (New Orleans, LA, U.S.A.: AIAA Press, 11–13. doi:10.2514/6.1997-3535
- Zhang, K., and Chen, W. (2011). “Reentry Vehicle Constrained Trajectory Optimization,” in Proceedings of the 17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference (San Francisco, California: AIAA Press, 11–14. doi:10.2514/6.2011-2231
- Zhang, Z., Wen, C., Xing, L., and Song, Y. (2022). Adaptive Output Feedback Control of Nonlinear Systems with Mismatched Uncertainties under Input/Output Quantization. *IEEE Trans. Autom. Contr.* doi:10.1109/TAC.2022.3159543

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.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Sands. 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.



OPEN ACCESS

EDITED BY

Holger Voos,
University of Luxembourg, Luxembourg

REVIEWED BY

Sunan Huang,
National University of Singapore,
Singapore
Dechao Chen,
Sun Yat-sen University, China

*CORRESPONDENCE

Dang Xuan Ba,
✉ badx@hcmute.edu.vn

SPECIALTY SECTION

This article was submitted to
Robotic Control Systems,
a section of the journal
Frontiers in Robotics and AI

RECEIVED 22 June 2022

ACCEPTED 06 December 2022

PUBLISHED 04 January 2023

CITATION

Minh Nguyet NT and Ba DX (2023), A
neural flexible PID controller for task-
space control of robotic manipulators.
Front. Robot. AI 9:975850.
doi: 10.3389/frobt.2022.975850

COPYRIGHT

© 2023 Minh Nguyet and Ba. 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 neural flexible PID controller for task-space control of robotic manipulators

Nguyen Tran Minh Nguyet¹ and Dang Xuan Ba^{2*}

¹Faculty of Electrical and Electronics Engineering, HCMC University of Technology and Education (HCMUTE), Ho Chi Minh City, Vietnam, ²Department of Automatic Control and Smart Robotic Center, HCMC University of Technology and Education (HCMUTE), Ho Chi Minh City, Vietnam

This paper proposes an adaptive robust Jacobian-based controller for task-space position-tracking control of robotic manipulators. Structure of the controller is built up on a traditional Proportional-Integral-Derivative (PID) framework. An additional neural control signal is next synthesized under a non-linear learning law to compensate for internal and external disturbances in the robot dynamics. To provide the strong robustness of such the controller, a new gain learning feature is then integrated to automatically adjust the PID gains for various working conditions. Stability of the closed-loop system is guaranteed by Lyapunov constraints. Effectiveness of the proposed controller is carefully verified by intensive simulation results.

KEYWORDS

intelligent controller, robotic, manipulators, PID controller, neural network

1 Introduction

Today, the great development of science and technology has created a premise for scientific research to develop to a new level in which the field of robotics has being chosen to be the leading industry by many countries. To promote science and technology backgrounds, intelligent robots in the industrial application are starting to prosper strongly, attracting many research experts. To control robot moving safely to desired positions with obstacles, collision avoidance and path planning were matters of concern. In recent years, various strategies have been studied for collision avoidance control purpose. The basic idea behind the collision avoidance algorithms is to design a proper controller which can result in a conflict-free trajectory. Path selection methods are the one of several techniques to avoid obstacles. It uses off-line/on-line algorithms to produce a curve that connects the starting and target points with a predefined initial position, velocity and acceleration. For example, an online trajectory generation algorithm called Ruckig considered third-order constraints (for velocity, acceleration, and jerk), so the complete kinematic state could be specified for waypoint-based trajectories (Berscheid and Kroeger, 2021). The smooth trajectory based on method combining of fourth and fifth order polynomial functions was presented in (Boscario et al., 2012) in which, the outcome of the method was the optimal time distribution of the *via* points, with respect to predefined objective function. After that, the joint based controller might use the inverse kinematic to solve the desired joint angular. Early collision avoidance approaches

concentrated on the static obstacles handling by the sensor-based motion planning methods (Borenstein and Koren, 1991), using nearness diagram navigation to successfully navigate in troublesome scenarios (Minguez and Montano, 2004) and using trajectory planning algorithms to avoid obstacles (Shiller, 2015). In reality, many techniques have been proposed to cope with moving obstacles. For instance, a reactive avoidance method incorporating with a non-linear differential geometric guidance was presented in (Mujumdar and Padhi, 2011) and a collision avoidance algorithm based on the potential fields was proposed in (Huang et al., 2019). It can be seen that in normal applications of robotic manipulators, the controllers were designed in the joint space in which it requires exact inverse kinematic computation as well. Nonetheless, complex internal dynamics and external disturbances coming from divergent working conditions are main obstacles hindering development of excellent controllers.

To realize control objectives of the robots in real-life missions, simple proportional-integral-derivative (PID) controllers are priority options (Bledt et al., 2018), (Wensing et al., 2017) due to simple design. If the proper control gains were found, the high control outcomes could be obtained (Park et al., 2015), (Ba and Bae, 2020). A lot of research have been then studied to improve the performance of the PID controllers using intelligent approaches such as evolutionary optimization and fuzzy logic (Astrom and Hagglund, 1995). The methods exhibited promising control results thanks to using both online and offline sections (Tan et al., 2004). The off-line control one could flexibly select the proper PID parameters based on the system overshoot, settling time and steady-state error, while the on-line one would adopt the operating control errors to adjust fuzzy logic parameters to re-optimize the system, improving the system quality significantly. However, the tuning methodology of fuzzy logic controllers is mostly based on experiences of operators (Juang and Chang, 2011). Another series of the intelligent control category was based on the biological properties of animals in which a genetic algorithm was combined with a bacterial foraging method to simulate natural optimization processes such as hybridization, reproduction, mutation, natural selection, etc., (Cucientes et al., 2007). This evolution could deliver the most optimal solution. That the solving process requires a large number of samples and takes a long-running time limits its application. Recently, tuning PID control parameters using neural networks has become an effective approach with many contributions (Kim and Cho, 2006), (Neath et al., 2014). The conventional PID one itself is a robust controller (Thanh and Ahn, 2006). The learning ability integrated to the controllers makes it flexible to the working environment (Ye, 2008). Lack of an intensive consideration of learning rules in steady-state time could make the system unstable in a long time used (Ba et al., 2019), (Ye, 2008), (Rocco, 1996).

To further improve the control performance, internal and external dynamics of robots need to be compensated during working processes. To this end, classical methods could be employed based on accurate mathematical models of the robots (Craig, 2018), (Zhu, 2010). Good control results were exhibited using such the conventional approaches, but it is not easy to extend the control outcome to complicated robot structures. Intelligent modeling methods could be adopted to increase applicability of the controllers to various robots in different working environments (Karayiannidis et al., 2016), (Gao et al., 2022). Excellent control performances were accomplished with the intelligent control approaches. However, convergence of the learning process is still not explicitly proven (He et al., 2020), (Wang et al., 2020). To support this kind of theoretical drawback, linear leakage functions were integrated the estimation phases of the network operation. However, this term could be slowdown the overall learning performance. Hence, advanced learning behaviors for the network need to be extensionally studied.

In this paper, an intelligent direct PID controller is proposed for position-tracking control in task space of robotic manipulators. Without using inverse kinematics, the operator just needs to input the desired position value, the controller will calculate and give the desired control position to the robot by itself (Craig, 2005; Ba and Bae, 2021; Ba et al., 2021). This process will be of great help since, in practice, there are quite few robots with quite complex hardware structures that make the inverse kinematics calculation difficult. The more degrees of freedom a robot has, the more difficult the calculation process, requiring more time and effort. The proposed controller is built based on a conventional PID framework. A non-linear neural network is then employed to eliminate internal/external disturbances during the working process. To increase the adaptive robustness of the controller, a new gain learning rule is integrated to flexible tune the PID gain for different working conditions.

Outline of the paper is structured as follows. Section 2 discusses system modeling and problem statements. Section 3 presents design of the proposed controller. Section 4 analyzes verification results. The paper is then concluded in Section 5.

2 System modelling and problem statements

Behaviors of a general robotic manipulator can be presented in the following form (Craig, 2018), (He et al., 2020):

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) + \tau_f + \tau_d = \tau, \quad (1)$$

where q , \dot{q} , \ddot{q} are respectively vectors of joint position, velocity, and acceleration, $M(q)$ is the mass matrix, $C(q, \dot{q})$ is the centrifugal-Coriolis moment, $G(q)$ is the gravitational

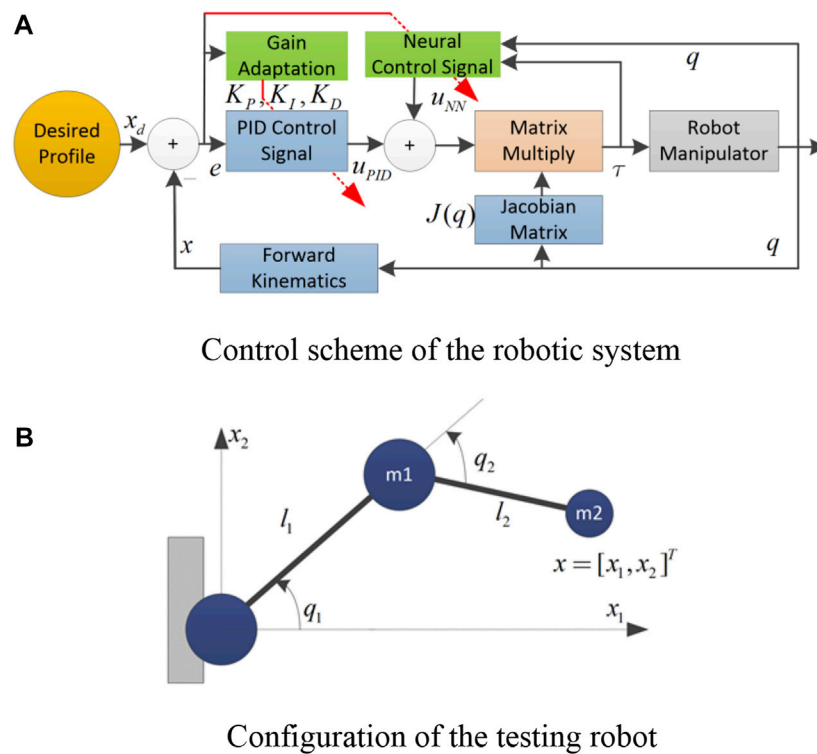


FIGURE 1

The testing robot. (A) Control scheme of the robotic system. (B) Configuration of the testing robot.

moment, τ_f is the frictional moment, τ_d stands for external disturbances, and τ is the actuator moment or control signals.

Remark 1: the control objective of this paper is to find out a proper control signal (τ) to control position of the end-effector of the robot following a desired profile. To accomplish this task, we can use inverse kinematics (IK) to compute desired joint positions from the end-effector reference signals. However, it is not trivial to find solutions of complicated robots. To avoid this shortcoming, we can apply direct control algorithms without caring of the IK problem. Hence, one needs consider dynamic model (1) in the task space as follows (Craig, 2018):

$$\ddot{x} = J(q)\bar{M}^{-1}(q)\tau + d, \quad (2)$$

where x is the end-effector position of the robot, $J(q)$ is the Jacobian matrix, and $\bar{M}(q)$ is the nominal value of the mass matrix $M(q)$, and d is the lumped disturbance as presented as follows:

$$d = J(q)\left(\ddot{M}^{-1}(q)\tau + J^{-1}(q)\dot{J}(q)\dot{q}\right) - J(q)\bar{M}^{-1}(q)(C(q, \dot{q}) + G(q) + \tau_f + \tau_d), \quad (3)$$

where $\bar{M}(q) = \ddot{M}(q) - \bar{M}(q)$ is the deviation mass matrix.

Remark 2: It is very difficult to determine accurate parameters of model (1), (2) or (3). Furthermore, the parameters sometimes vary during the working processes. To treat this drawback, the proposed controller is required to be model-free, robust and flexible.

3 Neural flexible PID controller

In this section, the proposed controller is designed with new features to realize the control mission stated. Theoretical effectiveness of the closed-loop system is then analyzed using Lyapunov constraints.

3.1 A flexible PID control framework

The controller is developed based on a conventional PID (Tan et al., 2004) structure as in Eq. 4.

$$\tau = -\bar{M}J^+\left(K_p e + K_d \dot{e} + K_i \int e dt\right) \quad (4)$$

where $e = x - x_d$ is the control objective, x_d is the desired trajectory, J^+ is pseudo-inverse of the Jacobian J and K_p, K_d, K_i are control gains.

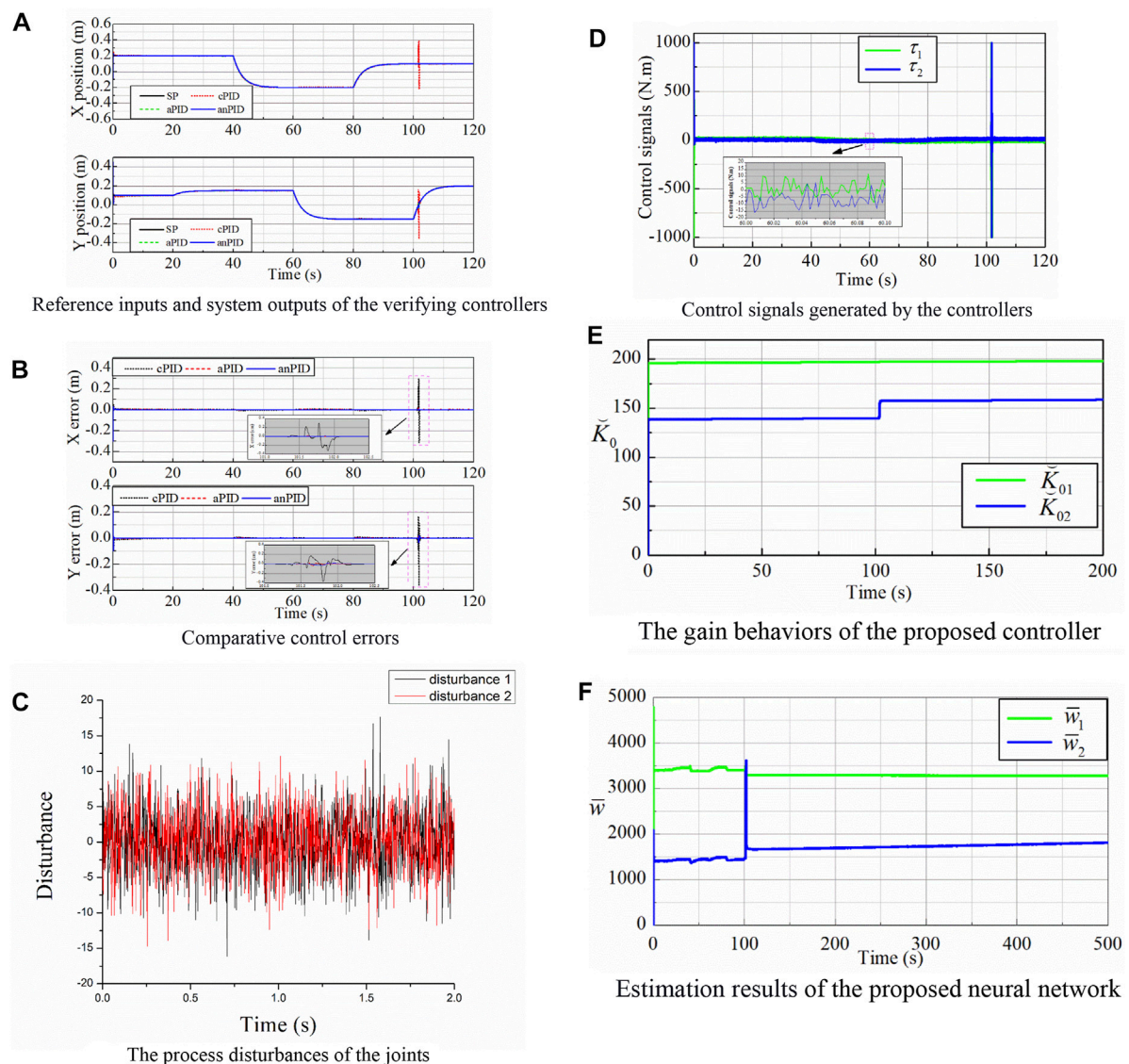


FIGURE 2

Simulation data of the controllers in the first simulation. (A) Reference inputs and system outputs of the verifying controllers. (B) Comparative control errors. (C) The process disturbances of the joints. (D) Control signals generated by the controllers. (E) The gain behaviors of the proposed controller. (F) Estimation results of the proposed neural network.

We assume that the desired trajectory x_d is inside of the workspace of the robot and the end-effector x of the robot can reach to the desired position selected. Advanced path-planning and obstacle-avoidance algorithms (Mujumdar and Padhi, 2011; Shiller, 2015; Huang et al., 2019) could be employed to generate appropriate desired profiles for the robot.

In real-time control, one can tune the control gains (K_p, K_d, K_i) for acceptable control performances. However, the fixed gains might not ensure good control errors for various working conditions (Thanh and Ahn, 2006), (Rocco,

1996). To cope with this problem, we propose an automatic tuning law for PID gains, as follows:

$$\begin{cases} K_p = K_2 K_2 + K_1 + K_2 \check{k}_0 \\ K_D = 2K_2 + \check{k}_0 \\ K_i = K_2 K_1 + \check{k}_0 K_1 \\ \dot{\check{k}}_0 = -\alpha_0 \text{diag}(|e|) \text{diag}(1 + |e|)^{-1} \check{k}_0 + \beta_0 \left(\dot{e} + K_2 e + K_1 \int e dt \right)^2 \end{cases} \quad (5)$$

where K_1, K_2 are positive core gains, α_0, β_0 are learning rates and $\check{k}_0 = \text{diag}(\check{k}_0)$ is the activation gain.

Remark 3: As seen in Eq. 5, the PID gains are structured from static and dynamic gains which respectively yield robustness and adaptation of the closed-loop system. The control gains are varied in non-linear manners to drive the control error to go into the desired region regardless of unknown environments. For faster control results, the disturbance term d needs to be effectively compensated by a proper control signal.

3.2 Additional neural network control signal

First of all, the disturbance d is modeled using the following Radial Basis Function (RBF) network:

$$d = W\xi(q) + \delta, \quad (6)$$

where W is the optimal weight vector, $\xi(q, \dot{q})$ is the regression vector, and δ is the modeling error.

Based on the neural network model (6), the control signal (4) is modified by adding an additional intelligent control term, as follows:

$$\tau = -\bar{M}J^+ \left(\underbrace{K_p e + K_d \dot{e} + K_i \int edt}_{u_{PID}} + \underbrace{\hat{W}\xi(q, \dot{q})}_{u_{NN}} \right) \quad (7)$$

where u_{PID} and u_{NN} stand control terms generated by PID and neural network structure, respectively, and \hat{W} is estimate of the weight vector W . The estimation \hat{W} is updated by the following non-linear mechanism:

$$\dot{\hat{w}}_{i|1..n} = -\alpha_w |e_i| (1 + |e_i|)^{-1} \hat{w}_i + \beta_w \left(\dot{e}_i + K_2 e_i + K_1 \int e_i dt \right) \xi_i, \quad (8)$$

where α_w and β_w are learning rates.

Remark 4: The system (8) uses rich information including time-derivative, linear, and integral function of the control error to activate the learning process. The weight matrix of the neural network is automatically updated to ensure the minimum control error.

3.3 Stability analysis

In this section, we discuss the stability of the closed-loop system to ensure reliability of the proposed controller for the robotic system (3). From the above design, we have the following statements.

Theorem 1: Give a task-space model (3) of robotic manipulators, if employing a conventional neural PID control

signal (7) supported by adaptive rules (5) and (8), the following properties hold:

- 1) The control error e , activation gain \check{k}_0 and the neural weight vectors are bounded.
- 2) In the stationary phase, the control error e converges to zero.

Proof:

We first synthesize a virtual control error (e_v) as follows:

$$e_v = \dot{e} + K_2 e + K_1 \int edt \quad (9)$$

The time derivative of the new error (e_v) under dynamics (3) and the model (6) is described

$$\dot{e}_v = J(q)\bar{M}^{-1}(q)\tau + W\xi(q) + \delta - \ddot{x}_d + K_2 \dot{e} + K_1 e \quad (10)$$

By substituting the control signal Eq. 7 and the gain structure Eq. 5 into the dynamics Eq. 10, we have a simpler form:

$$\begin{aligned} \dot{e}_v &= -J(q)\bar{M}^{-1}(q)\bar{M}J^+ \left(\underbrace{K_p e + K_d \dot{e} + K_i \int edt}_{u_{PID}} + \underbrace{\hat{W}\xi(q, \dot{q})}_{u_{NN}} \right) \\ &\quad + W\xi(q, \dot{q}) + \delta - \ddot{x}_d + K_2 \dot{e} + K_1 e \\ &= -K_2 \left(\dot{e} + K_2 e + K_1 \int edt \right) - \check{k}_0 \left(\dot{e} + K_2 e + K_1 \int edt \right) - \tilde{W}\xi(q, \dot{q}) \\ &\quad + \delta - \ddot{x}_d \\ &= -K_2 e_v - \check{k}_0 e_v - \tilde{W}\xi(q, \dot{q}) + \delta - \ddot{x}_d \end{aligned} \quad (11)$$

where $\tilde{W} = \hat{W} - W$ is estimation error of the neural weight matrix W .

We now consider a new Lyapunov function:

$$L = 0.5 e_v^T e_v + 0.5 \check{k}_0 \check{k}_0 + 0.5 \sum_{i=1}^n \tilde{w}_i^T \beta_{wi}^{-1} \tilde{w}_i. \quad (12)$$

Differentiating the function Eq. 12 with respect to time and noting the dynamics Eq. 11 lead to

$$\begin{aligned} \dot{L} &= e_v^T \dot{e}_v + \check{k}_0 \beta_0^{-1} \dot{\check{k}}_0 + \sum_{i=1}^n \tilde{w}_i^T \beta_{wi}^{-1} \dot{\tilde{w}}_i = -e_v^T (K_2 e_v + \check{k}_0 e_v - \delta + \ddot{x}_d) \\ &\quad - \sum_{i=1}^n e_{vi} \tilde{w}_i^T \xi_i + \check{k}_0 \beta_0^{-1} (-\alpha_0 \text{diag}(1 + |e|)^{-1} \check{k}_0 + \beta_0 e_v^2) \\ &\quad + \sum_{i=1}^n \tilde{w}_i^T \beta_{wi}^{-1} \left(-\alpha_w |e_i| (1 + |e_i|)^{-1} \hat{w}_i + \beta_w \left(\dot{e}_i + K_2 e_i + K_1 \int e_i dt \right) \xi_i \right) \\ &= -e_v^T (K_2 e_v - \delta + \ddot{x}_d) - \check{k}_0 \alpha_0 \text{diag}(|e|) \text{diag}(1 + |e|)^{-1} \check{k}_0 \\ &\quad + \sum_{i=1}^n \tilde{w}_i^T \beta_{wi}^{-1} (-\alpha_w |e_i| (1 + |e_i|)^{-1} (w_i + \tilde{w}_i^T)) \leq -0.5 e_v^T K_2 e_v \\ &\quad - \check{k}_0 \beta_0^{-1} \alpha_0 \text{diag}(|e|) \text{diag}(1 + |e|)^{-1} \check{k}_0 + 0.5 \lambda_{\max}(K_2^{-1}) \|\delta + \ddot{x}_d\|^2 \\ &\quad - \sum_{i=1}^n \tilde{w}_i^T \beta_{wi}^{-1} \alpha_w |e_i| (1 + |e_i|)^{-1} \hat{w}_i + \sum_{i=1}^n \beta_{wi}^{-1} \alpha_w |e_i| (1 + |e_i|)^{-1} \|w_i\|^2 \end{aligned} \quad (13)$$

Applying Cauchy-Schwarz inequality, we obtain the following result:

$$\begin{aligned} \dot{L} \leq & -0.5e_v^T K_2 e_v - \check{k}_0 \beta_0^{-1} \alpha_0 \text{diag}(|e|) \text{diag}(1 + |e|)^{-1} \check{k}_0 \\ & - \sum_{i=1}^n \tilde{w}_i^T \beta_{wi}^{-1} \alpha_w |e_i| (1 + |e_i|)^{-1} \tilde{w}_i + \Delta \end{aligned} \quad (14)$$

where Δ is a lumped term defined as

$$\Delta = 0.5\lambda_{\max}(K_2^{-1})\|\delta + \ddot{x}_d\|_{\max}^2 + \sum_{i=1}^n \beta_{wi}^{-1} \alpha_w \|w_i\|_{\max}^2 \quad (15)$$

Since w_i and δ are bounded, hence Δ is bounded as well. This discussion leads to the proof of the first statement of [Theorem 1](#).

In the stationary phase, the time derivative of the virtual control error e_v converges zeros. By differentiating Eq. 9 with respect to time and applying Hurwitz criterion on the results, we can achieve the second proof of [Theorem 1](#).

Remark 5: As carefully observing on the definition (15), one could select K_2 and β_{wi} to large enough to reduce the disturbance bound Δ . However, these are still fixed values. From Eq. 14, it can be seen that the control performance could be enhanced by the learning gain k_0 for various working cases. The control idea is graphically summarized in [Figure 1A](#). The following implementation procedure could be referred for deploying the proposed control algorithm on simulation or real-time testing. 1) In the first step, all of the learning rates ($\alpha_0, \beta_0, \alpha_w$ and β_w) are set to be zeros. The positive core gains (K_1, K_2) are manually tuned for acceptable control performances. The gain K_2 are recommended to be greater than the gain K_1 . 2) In the second step, the learning rates (α_0 and β_0) of the activation gain (K_0) are adjusted to further enhance the control performance. In this step, the core gains (K_1, K_2) could be retuned in some cases for higher control precision. 3) In the third step, the regression vector $\xi(q, \dot{q})$ is built and the learning rates (α_w and β_w) of the neural network are manually selected bring the control accuracy to a higher level. The whole tuning procedure could be applied several times for seeking an excellent control outcome. Note that, from the second turn, it does not need to reset the learning rates ($\alpha_0, \beta_0, \alpha_w$ and β_w) to be zeros anymore.

4 Validation results

This section presents validation results of the proposed controller in simulations. The control algorithm was applied to a 2-degree-of-freedom (DOF) robot, as sketched in [Figure 1B](#). The manipulator was modeled as two rigid links with lengths of l_1 and l_2 . The mass was distributed at the end of each link (m_1, m_2). The robot would work in a vertical plane with downward gravitational acceleration. Viscous friction was modeled at the joints (a_1, a_2). Although this robot is quite simple, it contains all the necessary components of a general

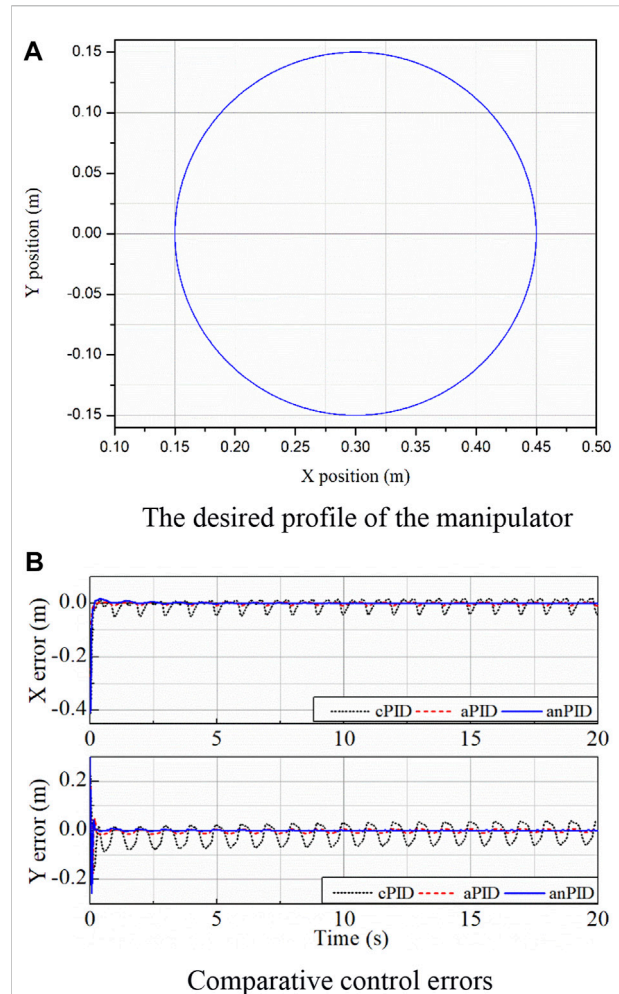


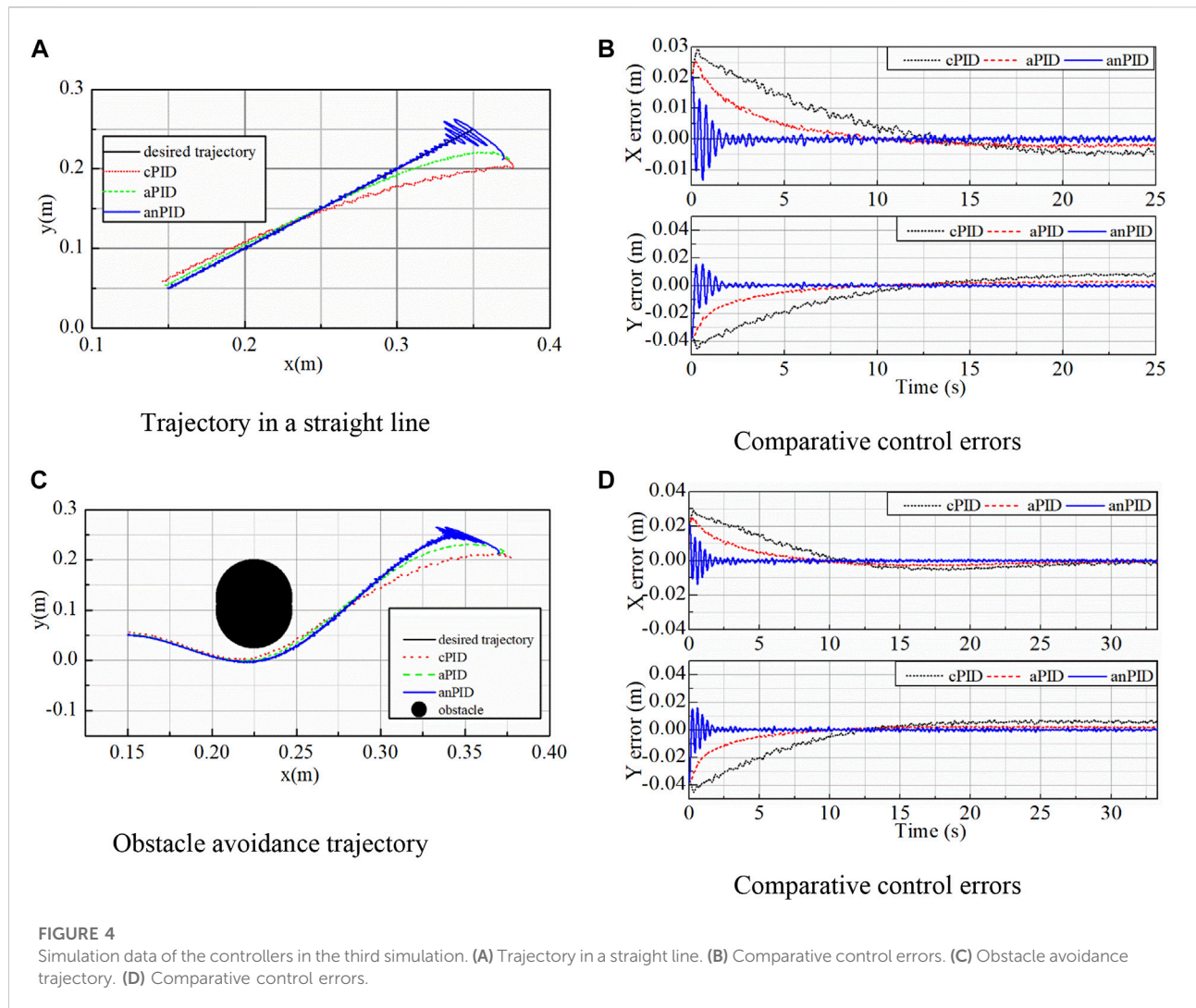
FIGURE 3
Simulation data of the controllers in the second simulation.
(A) The desired profile of the manipulator. (B) Comparative control errors.

multi-degree of freedom manipulator including moment of inertia, centrifugal terms, Coriolis terms, gravity terms and friction effects.

The detailed dynamic equations of the robot are as follows:

$$\begin{cases} \tau_1 = m_2 l_2^2 (\ddot{q}_1 + \ddot{q}_2) + l_1 l_2 m_2 \cos(q_2) (2\ddot{q}_1 + \ddot{q}_2) \\ \quad + (m_1 + m_2) l_1^2 \ddot{q}_1 \\ \quad - m_2 l_1 l_2 \sin(q_2) \dot{q}_2 (\dot{q}_2 + 2\dot{q}_1) + m_2 l_2 g \cos(q_1 + q_2) \\ \quad + (m_1 + m_2) l_1 g \cos(q_1) + a_1 \dot{q}_1 \\ \tau_2 = m_2 l_2^2 (\ddot{q}_1 + \ddot{q}_2) + l_1 l_2 m_2 \cos(q_2) \ddot{q}_1 + m_2 l_1 l_2 \sin(q_2) \dot{q}_1^2 \\ \quad + m_2 l_2 g \cos(q_1 + q_2) + a_2 \dot{q}_2 \end{cases} \quad (16)$$

To estimate the disturbances d , we used an RBF neural network with 4 input neurons, 256 hidden neurons and 2 output neurons.



The actual values of the length of links, mass and viscous friction coefficients were chosen as follows: $l_1 = 0.2$; $l_2 = 0.3$; $m_1 = 7$; $m_2 = 3.5$; $a_1 = 3$; $m_2 = 10$

To evaluate the adaptability and robustness of the controller under divergent working conditions, we compared the proposed controller (called anPID) with a conventional PID controller (referred to as cPID) and an adaptive PID controller with using only automatic tuning law for PID gains (referred to as aPID). The parameters of the controller were chosen as: $K_1 = \text{diag}([5; 5])$, $K_2 = \text{diag}([50; 50])$, $\bar{M} = \text{diag}([0.1; 0.1])$ while learning coefficients were $\alpha_0 = 0.01$, $\beta_0 = 40$ and $\alpha_w = 0.01$, $\beta_w = 50$

To carefully express the performance of the proposed controller, the robotic manipulators were simulated in three cases. In the first simulation, the robot was controlled to track the desired trajectories of smooth multi-step signals. Furthermore, process disturbances in the form of white noises, as shown in Figure 2C, were added to the output

torques of the actuators. Simulation results of the conventional and intelligent PID controllers for the tracking control mission are also shown in Figure 2.

Figures 2A,B shows that the proposed controller maintained good control errors even though the end-effector of the manipulator worked throughout a singularity point of (0.1; 0) (m). Figure 2D exhibits the control signals of the smart PID controller which had large values at the initial and singularity points in order to decrease the control errors as fast and much as possible. This superior property was the achievement of the learning laws (5) and (8) that are demonstrated by the gain and weight variations as depicted in Figures 2E, F, respectively. These terms were first started from the zero value, then their values had a large overshoot to bring the system to the steady state rapidly. It can be seen that the system adapted to the reasonable approximation of the disturbances to bring the control error to the smallest possible value. Therefore, the learning ability of the system has been confirmed with

TABLE 1 Statistical computation of the controllers from the validation results.

Control error		X position		Y Position	
		MA	RMS	MA	RMS
The 1st case	cPID	0.0045	0.0023	0.0036	0.0021
	aPID	0.0019	9.36×10^{-4}	0.0016	7.66×10^{-4}
	anPID	7.28×10^{-4}	2.45×10^{-4}	7.16×10^{-4}	2.27×10^{-4}
The 2nd case	cPID	0.0423	0.0189	0.059	0.0362
	aPID	0.0089	0.0046	0.0088	0.0059
	anPID	7.33×10^{-4}	3.14×10^{-4}	0.0016	4.91×10^{-4}
The 3rd case (no obstacle)	cPID	0.0055	0.0044	0.0089	0.0077
	aPID	0.0029	0.0021	0.0033	0.0027
	anPID	0.0013	5.24×10^{-4}	0.0012	5×10^{-4}
The 3rd case (obstacle)	cPID	0.0048	0.0037	0.0072	0.0057
	aPID	0.0026	0.0018	0.0029	0.0022
	anPID	8.37×10^{-4}	3.14×10^{-4}	0.0016	6.9×10^{-4}

uncertain non-linearities and perturbations through this simulation validation.

The manipulator was employed to draw a circle whose radius was 0.15 m and origin was at a point of (0.3; 0) (m) with a frequency of 1 Hz in the second simulation. The reference input used is shown in [Figure 3A](#).

With the application of the neural flexible PID controller for unknown environments but using the adaptive rule (7), the control results obtained are presented in [Figure 3B](#). From the data in this figure, although disturbances were not known in advance, the control qualities of the joints were good at both the transient and steady-state phases. The results were achieved thanks to the learning characteristics of the PID gains and the designed RBF neural network. There was a little overshoot in the y -direction error due to the large learning rate selected, but this overshoot might cause the system to quickly reach steady state. From the comparison of the control data in [Figure 3B](#), it can be seen that the quality of proposed controller (anPID) was better than that of the aPID controller which was employed only one learning law (5). This is possible because the more adaptive terms the controller had, the more approximation with disturbances it gained.

In the third simulation, the end effector of robot manipulator was controlled to move from a point of (0.35; 0.25) (m) to another point of (0.15; 0.05) (m). After applying the three controllers for this mission in a free condition in which the desired trajectory was planned as a straight line, their control outcomes including the actual outputs and the control errors were illustrated in [Figures 4A, B](#), respectively. In these figures, although the proposed controller (anPID) had more oscillation

in the transient state to find adaptive term quickly, it had smallest overshoot and steady state error when compared with cPID and aPID controllers.

To further challenge the controllers with a more difficult working condition, an obstacle was set on the moving trajectory of the robot in the task space. By applying the trajectory planning method and the referred avoidance collision method ([Borenstein and Koren, 1991](#)), ([Craig, 2005](#)), the desired trajectory was generated as a curve by using two third-order-segment polynomials for the position, velocity and acceleration of the end-effector. The control data in this case are shown in [Figures 4C, D](#). From the comparison of the data in these figures, it can be seen that the control quality of proposed controller (anPID) was better than that of the others (aPID and cPID) even though with the non-linear trajectory generated.

[Table 1](#) described the maximum absolute (MA) and root-mean-square (RMS) values of the control performances for a specified manipulated time (20 s–25 s). The proposed controller always provided the best MA and RMS error in all cases. These results show that the proposed control technology compensated efficiently for the non-linear uncertainties and unknown disturbances. Here, the advantages of the proposed controller have been confirmed. Therefore, the simulation results have proved that the studied control method outperform over the previous ones.

5 Conclusion

In this paper, an intelligent controller is proposed to optimize the position control performance of a 2DOF

robotic manipulator. The controller is developed based on a conventional PID structure. New advanced features designed for disturbance learning and gain adaptation are then integrated into the ordinary control signal to improve its robustness and result in high control accuracies. The control efficiency of the proposed approach was then successfully verified by theoretic proof and comparative simulations. It can confirm that the controller is model-free, simple, robust and flexible. In the near future, the proposed control algorithm will be integrated with an additional control term that could result in asymptotic control performances for dynamical trajectories. Furthermore, advanced path-planning and obstacle-avoidance algorithms will be considered to combine with the controller to increase the flexibility when the system works in complex environments.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

References

- Astrom, K., and Hagglund, K. (1995). *PID controllers: Theory, design and tuning*. Washington, DC, USA: ISA Press.
- Ba, D. X., and Bae, J. B. (2020). A nonlinear sliding mode controller of serial robot manipulators with two-level gain-learning ability. *IEEE Access* 8, 189224–189235. doi:10.1109/access.2020.3032449
- Ba, D. X., and Bae, J. B. (2021). A precise neural-disturbance learning controller of constrained robotic manipulators. *IEEE Access* 9, 50381–50390. doi:10.1109/access.2021.3069229
- Ba, D. X., Tran, M. S., Vu, V. P., Tran, V. D., Tran, M. D., Tai, N. T., et al. (2021). “A neural-network-based nonlinear controller for robot manipulators with gain-learning ability and output constraints,” in 2021 International Symp. Electrical and Electronics Engineering (ISEE), Ho Chi Minh, Vietnam, 149–153.
- Ba, D. X., Yeom, H., and Bae, J. B. (2019). A direct robust nonsingular terminal sliding mode controller based on an adaptive time-delay estimator for servomotor rigid robots. *Mechatronics* 59, May. doi:10.1016/j.mechatronics.2019.03.007
- Berscheid, L., and Kroeger, T. (2021). “Jerk-limited real-time trajectory generation with arbitrary target states,” in Proceedings of Robotics: Science and Systems, July 2021.
- Bledt, G., Powell, M. J., Katz, B., Carlo, F. D., Wensing, P. W., and Kim, S. (2018). MIT cheetah 3: Design and control of a robust, dynamic quadruped robot in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain.
- Borenstein, J., and Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom.* 7 (3), 278–288. doi:10.1109/70.88137
- Boscario, P., Gasparetto, A., and Vidoni, R. (2012). “Planning continuous-jerk trajectories for industrial manipulators,” in Proceedings of the ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis, Nantes, France, July 2012.
- Craig, J. J. (2018). *Introduction to robotics: Mechanics and control*. 4. Hoboken, NJ, USA: Pearson Prentice Hall, th ed.
- Craig, J. J. (2005). Manipulator dynamic *Introduction to robotics: Mechanics and control*. 3. Hoboken, NJ, USA: Pearson Prentice Hall, 165–200. inrd ed.s
- Cucientes, M., Moreno, D. L., Bugarin, A., and Barro, S. (2007). Design of a fuzzy controller in mobile robotics using genetic algorithms. *Appl. Soft Comput.* 7 (2), 540–546. doi:10.1016/j.asoc.2005.05.007
- Gao, X., Li, X., Sun, Y., Hao, L., Yang, H., and Xiang, C. (2022). Model-free tracking control of continuum manipulators with global stability and assigned accuracy. *IEEE Trans. Syst. Man. Cybern. Syst.* 52 (2), 1345–1355. doi:10.1109/tsmc.2020.3018756
- He, W., Sun, Y., Yan, Z., Yang, C., Li, Z., and Kaynak, O. (2020). Disturbance observer-based neural network control of cooperative multiple manipulators with input saturation. *IEEE Trans. Neural Netw. Learn. Syst.* 31 (5), 1735–1746. doi:10.1109/tnnls.2019.2923241
- Huang, S., Teo, R. S. H., and Tan, K. K. (2019). Collision avoidance of multi unmanned aerial vehicles: A review. *Annu. Rev. Control* 48, 147–164. doi:10.1016/j.arcontrol.2019.10.001
- Juang, C. F., and Chang, Y. C. (2011). Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments. *IEEE Trans. Fuzzy Syst.* 19 (2), 379–392. doi:10.1109/tfuzz.2011.2104364
- Karayiannidis, Y., Papageorgiou, D., and Doulgeri, Z. (2016). A model-free controller for guaranteed prescribed performance tracking of both robot joint positions and velocities. *IEEE Robot. Autom. Lett.* 1 (1), 267–273. doi:10.1109/lra.2016.2516245
- Kim, D. H., and Cho, J. H. (2006). A biological inspired intelligent PID controller tuning for AVR systems. *Int. J. Control, Automation, Syst.* 4 (5), 624–636.
- Minguez, J., and Montano, L. (2004). Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios. *IEEE Trans. Robot. Autom.* 20 (1), 45–59. doi:10.1109/tra.2003.820849
- Mujumdar, A., and Padhi, R. (2011). Reactive collision avoidance of using nonlinear geometric and differential geometric guidance. *J. Guid. Control Dyn.* 34 (1), 303–311. doi:10.2514/1.50923
- Neath, M. J., Swain, A. K., Madawala, U. K., and Thrimawithana, D. J. (2014). An optimal PID controller for a bidirectional inductive power transfer system using multiobjective genetic algorithm. *IEEE Trans. Power Electron.* 19 (3), 1523–1531. doi:10.1109/tpel.2013.2262953
- Park, H. W., Park, S., and Kim, S. (2015). “Variable-speed quadrupedal bounding using impulse planning: Untethered high-speed 3D Running of MIT Cheetah 2,” in

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, USA. in.,

Rocco, P. (1996). Stability of PID control for industrial robot arms. *IEEE Trans. Robot. Autom.* 12 (4), 606–614. doi:10.1109/70.508444

Shiller, Z. (2015). Off-line and on-line trajectory planning *Motion and operation planning of robotic systems, mechanisms and machine science*. Switzerland: Springer International Publishing, 29–62.

Tan, G. Z., Zeng, Q. D., and Li, W. B. (2004). Intelligent PID controller based on ant system algorithm and fuzzy inference and its application to bionic artificial leg. *J. Cent. South Univ. Technol.* 11, 316–322. doi:10.1007/s11771-004-0065-7

Thanh, T. D. C., and Ahn, K. K. (2006)., 16. *Mechatronics*. doi:10.1016/j.mechatronics.2006.03.011 Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network

Wang, M., Wang, Z., Chen, Y., and Sheng, W. (2020). Adaptive neural event-triggered control for discrete-time strict-feedback nonlinear systems. *IEEE Trans. Cybern.* 50 (7), 2946–2958. doi:10.1109/tcyb.2019.2921733

Wensing, P. M., Wang, A., Seok, S., Otten, A., Lang, J., and Kim, S. (2017). Proprioceptive actuator design in the MIT cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots, *IEEE Trans. Robot.* 33. *IEEE Transactions on Robotics*, 509–522. doi:10.1109/tro.2016.2640183

Ye, J. (2008). Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot. *Neurocomputing* 71. doi:10.1016/j.neucom.2007.04.014

Zhu, W. H. (2010). *Virtual Decomposition Control: Toward hyper degrees of freedom robots*. Berlin Heidelberg: Springer-Verlag.



OPEN ACCESS

EDITED BY

Allahyar Montazeri,
Lancaster University, United Kingdom

REVIEWED BY

Hung Pham Duy,
Vietnam National University, Hanoi,
Vietnam
Khoshnam Shojaei,
Islamic Azad University of Najafabad, Iran

*CORRESPONDENCE

Simone Monteleone,
✉ simone.monteleone@phd.unipi.it

SPECIALTY SECTION

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

RECEIVED 18 November 2021

ACCEPTED 14 December 2022

PUBLISHED 20 January 2023

CITATION

Monteleone S, Negrello F, Grioli G,
Catalano MG, Bicchi A and Garabini M
(2023), A method to benchmark the
balance resilience of robots.
Front. Robot. AI 9:817870.
doi: 10.3389/frobt.2022.817870

COPYRIGHT

© 2023 Monteleone, Negrello, Grioli,
Catalano, Bicchi and Garabini. This is an
open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). 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 method to benchmark the balance resilience of robots

Simone Monteleone^{1*}, Francesca Negrello², Giorgio Grioli²,
Manuel G. Catalano², Antonio Bicchi^{1,2} and Manolo Garabini¹

¹Centro di Ricerca E. Piaggio e Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa, Italy, ²Istituto Italiano di Tecnologia, Genova, Italy

Robots that work in unstructured scenarios are often subjected to collisions with the environment or external agents. Accordingly, recently, researchers focused on designing robust and resilient systems. This work presents a framework that quantitatively assesses the balancing resilience of self-stabilizing robots subjected to external perturbations. Our proposed framework consists of a set of novel Performance Indicators (PIs), experimental protocols for the reliable and repeatable measurement of the PIs, and a novel testbed to execute the protocols. The design of the testbed, the control structure, the post-processing software, and all the documentation related to the performance indicators and protocols are provided as open-source material so that other institutions can replicate the system. As an example of the application of our method, we report a set of experimental tests on a two-wheeled humanoid robot, with an experimental campaign of more than 1100 tests. The investigation demonstrates high repeatability and efficacy in executing reliable and precise perturbations.

KEYWORDS

benchmarking method, self-stabilizing robots, robots balance, performance assessment, robustness

1 Introduction

The growing employment of robots in real-world applications, e.g., exploration of hazardous environments (Negrello et al., 2018) and household assistance (Parmiggiani et al., 2017), emphasizes the necessity of robots safe and resilient against disturbances. In engineering, Hollnagel et al. (2006) defined resilience as “the ability of an organization (system) to keep or recover quickly to a stable state, allowing it to continue operations during and after a major mishap or in the presence of continuous significant stresses”. Zhang brought the concept of resilience into the robotic field (Zhang and Lin, 2010), while Zhang et al., 2017 proposes a set of principles for the design of soft and resilient robots.

Following Hollnagel and Zhang's interpretations, we investigate the definition of resilience for self-stabilizing robots. Self-stabilizing robots are a group of robotic systems with the common trait of possessing an unstable equilibrium stabilized continuously through control. Their increment of control and design complexity is accepted in the face of the augmented dexterity and agility that they show when compared to stable robots, such as mobile base robots (Fuchs et al., 2009). In the face of this augmented dexterity,

the possibility of facing unexpected falls, which may cause damage to the robot, the surroundings, or persons, arises and become the major issue for a self-stabilizing robot to cease operations. As a result, for these robots, the concept of resilience expressed by Zhang and Lin, 2010 should be re-defined as the ability of a system to maintain a stable state, allowing it to continue operating in the presence of continuous and significant perturbations. In this sense, studying the resilience of robots becomes closely connected to looking into balancing abilities.

However, nowadays, the measurement of robots' balancing resilience is a novel research field and still mostly relies on qualitative methods. To this aim, one of the most advanced fields is the legged locomotion research community. Nevertheless, in the related literature, it is possible to find just heuristic tests: Pushes) (Barasuol et al., 2013) (Feng et al., 2016), tilting the support surface (Li et al., 2013), balancing over a soft ground (Henze et al., 2016), or impacting with heavy masses (Kanzaki et al., 2005). These assessment methods are qualitative or hardly repeatable and do not allow comparisons between different robots. Benchmarking the performance of robotic systems offers many advantages. It allows for quantifying the performance of various systems, making comparisons possible, and fostering improvements. In industry, performance quantification makes possible standardization of technologies and regulation of the processes for manufacturing and commercialization of certified robots (Torricelli et al., 2015). Hence, a growing interest in the field of benchmarking has arisen during recent years in the research community (Negrello et al. (2020)) (Stasse et al. (2018)), especially for legged systems (Torricelli et al., 2015). Torricelli et al., 2015 and Torricelli and Pons, 2018 paved the way for benchmarking platforms for self-stabilizing robots and exoskeletons with the European Project EUROBENCH 2020¹. Seventeen sub-projects work under the Eurobench consortium, each accounting for a different aspect of robot performance. To give some valuable examples, in Taborri et al., 2020, the authors present *B.E.A.T.*, a benchmark for evaluating the static and dynamic balance of wearable human-assisting devices. In Lippi et al., 2019, Lippi et al., 2020, the authors proposed *COMTEST*, a similar framework for testing the performance of humanoid robots, as well as a set of Performance Indicators that aim to standardize the capabilities of robots on a universal level. In Vicario et al., 2021, the authors present *FORECAST*, a benchmarking method able to “define an objective score for a given force-controlled system accounting for its sensitivity to environmental uncertainties and variations.” Lastly, in Bayón et al., 2021, the authors proposed *BenchBalance*, a “Benchmarking solution proposed to conduct reproducible assessments of balance in various conditions,

mainly focused on wearable robots but also applicable to humanoids.”

We propose an evaluation framework for characterizing the resilience of self-stabilizing robots subjected to external disturbances (Monteleone et al., 2020). In Monteleone et al., 2020, we introduced the early conceptual definition of the testbed, with experimental validation on a two-wheeled robot solely on impulsive conditions, enforced via a non-actuated prototype of the testbed. In this work, we developed further the conceptual definition of the testbed, designing a framework composed of seven novel PIs to evaluate the resilience of a robot, five original experimental protocols for assessing the PIs, and a new testbed for reproducible issuing of both dynamic and static perturbations. The novel framework comprises an actuated structure equipped with a brake and clutch to perform various disturbances and protect the robot and operators against accidental impacts. The PIs, the experimental protocols, and the actuated and adjustable structure design are novelties in the state of the art. The proposed system draws inspiration from classical resilience testing machines used for the characterization of materials samples (as the Charpy test stand (ISO, 2010)), realizing a system that can apply a desired impulsive, repetitive, or static disturbance in the most straightforward and easily reproducible way. As a previous work, we designed a non-actuated benchmark structure to test the resilience of the soft hand grasping under impulsive loads (Negrello et al., 2020). The novel system integrates position and force sensors to characterize the disturbance we are applying to the robot. It is actuated to control the application of perturbations under static and dynamic conditions. As a specific use case of applying our method, we report a set of experimental tests on a particular two-wheeled base humanoid robot (Lentini et al., 2019). The main contributions of this work are the definition of the performance indicators, the testing protocols, and the mechanical design and control of the testbed. Additionally, all the materials are presented as open source and can be found on the external link in Section “Data Availability Statement”.

The resilience characterization framework we propose will pave the way for a rigorous benchmarking process of robot performance. The impact of our framework could go far beyond the balancing resilience characterization of wheeled robots (such as Alter-Ego from IIT/Research Center “E. Piaggio” or Golem Krang from Georgia Institute of Technology (Stilman et al., 2010) and could include the assessment of the balancing of autonomous legged robots. Today autonomous legged robotics is one of the most vibrant and hot research topics and is also significantly changing the industrial landscape. The population of humanoid (see, e.g., COMAN from “Istituto Italiano di Tecnologia (IIT)” (Tsagarakis et al., 2013) or Valkyrie from NASA (Radford et al., 2015) and quadrupedal (see, e.g., HyQ from IIT (Semini et al., 2011) or AnyMal from ANYbotics (Hutter et al., 2016) prototypes has dramatically

¹ <https://eurobench2020.eu/>.

increased in the last decades, as well as the related scientific publications. Moreover, today there exist several companies that develop and commercialize legged robots (e.g., agilityrobotics², unitree³, ANYbotics⁴, SoftBank⁵), with the remarkable recent acquisition of Boston Dynamics⁶ by a large automotive corporation, promising to be the core of a novel industrial segment. Finally, a strong impact of our work is expected in the field of assistive robotics (e.g., wearable robots (Kazerooni, 2005) (Khazoom et al., 2020) and personal robots (Parmiggiani et al., 2017), that is experiencing a growing trend similar to one of the autonomous legged robots.

The paper is organized as follows: **Section 2** briefly recalls the requirements necessary to define a robotic system performance. **Section 3** describes the performance indicators that we propose to define the resilience and performance of systems subjected to external perturbations. **Section 4** and **Section 5** present the design of the testbed, which we use to test the systems and the experimental protocols to execute reliable and repeatable experiments with different perturbation conditions. **Section 6** shows the results of applying this testing strategy to Alter-Ego, a two-wheeled humanoid robot. **Section 7** discusses the results based on the acquired data, exposing the quality and the possible future improvements of the benchmarking method. **Section 8** concludes the paper.

2 Methodology and concept

Given the youth of robotics (about 60 years), benchmarking the performance of robots is a novel study area. Other fields, such as biomechanics, have extensively researched this problem to assess, for example, humans' balance and locomotion capabilities. The literature proposes several methods and structures to characterize the performance of human balancing (e.g., Berg et al., 1992; Zemková, 2011; Molnar et al., 2018). They typically rely on the application of perturbations to the subject, e.g., asymmetric (Vashista et al., 2013), impulsive (Ellis et al., 2014), or active (Vashista et al., 2014) disturbances at the Center of Mass (CoM), or on the distal arts, e.g., Barasuol et al., 2013. From the literature, the importance of employing different forms of disturbances to measure the performance of a system is evident. To characterize the balancing performance of a robot, we require a system capable of providing

a variety of perturbations on robotic structures and collecting a meaningful set of data.

When applied to robotic applications, these stimuli can be helpful in evaluating performance in typical stress situations. Impacts are disturbances that naturally occur in unstructured environments and collaboration with humans. Single impacts are the most common perturbations in robotics, but various types of pushes occur periodically or that last over time. Repetitive and quasi-static perturbations are sustained during Human-Robot Interactions (HRI). This kind of stimulus helps people perceive robots as human-like entities rather than mechanical systems (Hyon et al., 2007).

By analyzing the response of systems subjected to various types of disturbances (see **Section 3**), we point out the necessity to define some indices that can easily show the limits and capability of systems. In this way, we can determine the experiments necessary to obtain the desired performance indicators (**Figure 1C**) that will constitute the core of our protocols and structure. In the case of repetitive or quasi-static perturbation, we believe it is worth dividing the stimuli between position-driven and force-driven ones. Indeed, despite their superficial similarities, robots react to CoM position perturbations and force perturbations in distinct ways. Repetitive perturbations are the finest example (see **Section 6**).

Perturbations can be applied to robots they are while moving or standing still. Balancing in a standing position may appear a trivial task, but it is the primary condition for any self-stabilizing robot to work. Balancing in the presence of perturbations is a crucial topic in the literature. As examples, in Stephens, 2007, Ott et al., 2011, and Liu and Atkeson, 2009, the authors present strategies and controllers to recover from significant disturbances and maintain an upright posture. Understanding the limits of performance in robot balancing is the basis for fostering self-stabilizing system technologies.

During data acquisition, we must rely on measurements resulting from the testbed sensors. The test bench would be used to assess multiple robots, and we can not know *a priori* which measurements are accessible from the robot side. Accordingly, the measured values of the test bench sensors should be consistent, simple, and repeatable, allowing more reliable performance computations. Measures coming from robots are not sufficient, and therefore, we must rely only on commercial sensors integrated into the test bench. We want to investigate robotic structures that may have very different dimensions. Due to that, the test platform should adapt to the size of the robot under evaluation (**Figure 1D**). The test bench should be fully modular, allowing different disturbance conditions and locations. Complete system modularity guarantees the highest flexibility during tests. The use of the proposed device could be extended to other benchmarking scenarios, especially those involving stability against disturbances on different terrains, such as walking on slopes or irregular terrain.

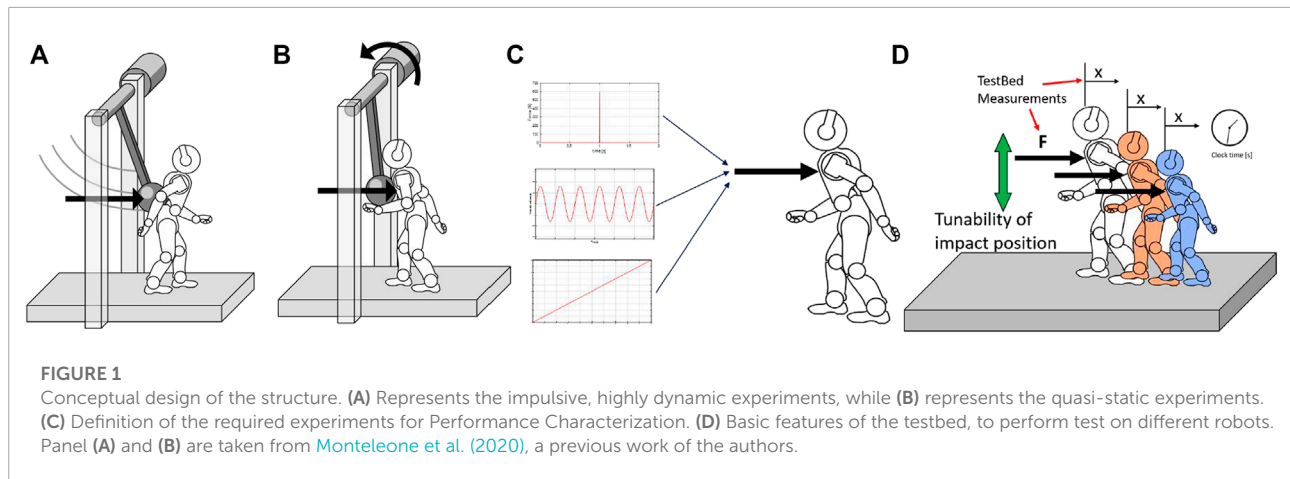
² Agility Robotics, <https://www.agilityrobotics.com>.

³ Unitree robotics, <https://www.unitree.com>.

⁴ ANYbotics, <https://www.anybotics.com>.

⁵ Soft Bank Robotics, <https://www.softbankrobotics.com>.

⁶ Boston Dynamics, <https://www.bostondynamics.com>.



During the development of the conceptual structure, we design a pendulum-like system, aiming to make its main dimensions (such as the height of the pendulum shaft, the length of the pendulum, and inertia) adjustable to match one of the different robots. In the design phase, we focus on the key features the benchmark must possess (Figure 1D). The primary purpose of this structure is to collect data from the robotic systems to define a set of Performance Indicators. These indicators are detailed in the next section.

3 Performance indicators

PIs describe the resilience of a self-stabilizing robot and allow comparisons among different robotics systems. The resilience of robots, in particular self-stabilizing systems, is influenced by structural robustness, but their balancing capabilities also cover a significant role. We divided PIs into two categories. The first contains those indices that show the limits at which the robot loses its balancing capability (see Section 3.1). The second is composed of those indices that describe the properties of systems subjected to perturbations (see Section 3.2).

3.1 Resilience limits

3.1.1 Impulsive resilience

Impulsive Resilience (IR) defines the maximum impulsive perturbation a robot can withstand without breaking or falling. Impacts are described by the impulse (I) and energy involved (E). Therefore, the IR is a diagram in which I lies on the x-axis and E on the y-axis. The resulting “resilience regions” (Figure 2B) are areas of the graph that describe the conditions at which the robot withstands the shock (light blue) or falls (red).

3.1.2 Excited resilience

The Excited Resilience (ER) defines the maximum perturbation a robot subjected to repetitive shocks can tolerate without breaking or falling. A repetitive disturbance is described by its amplitude (A) and the frequency (f) at which it is repeated. More in detail, the load can be a displacement or force perturbation. Hence, the ER are two plots in which f lies on the x-axis and A on the y-axis. The first shows the resilience regions of the robot subjected to repetitive displacement oscillations (Figure 2C), while the second displays the resilience regions related to force oscillations (Figure 2D).

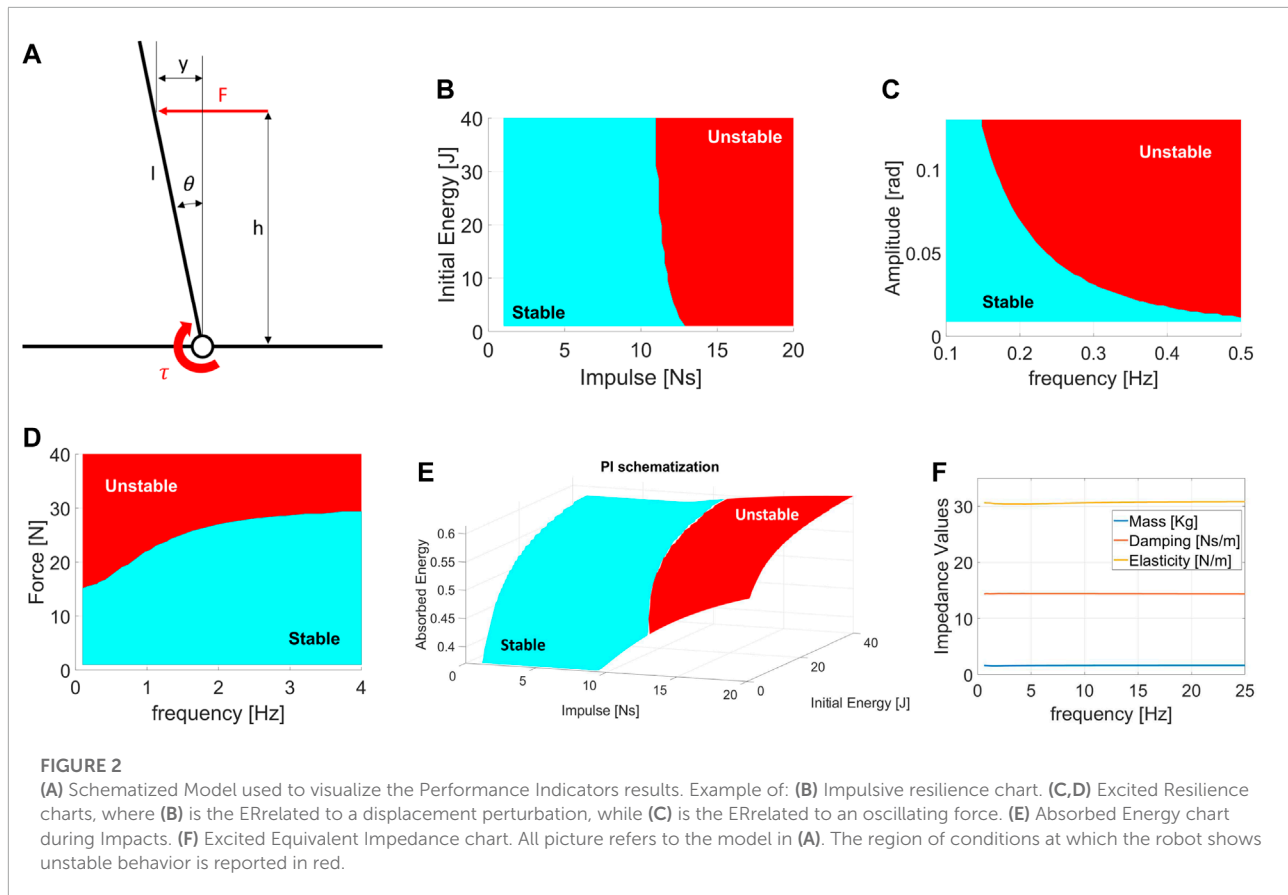
3.1.3 Quasi-static resilience

The Quasi-Static Resilience (QSR) defines the maximum perturbation a robot subjected to constant loads tolerates without breaking or falling. A constant load is described by its value in terms of force or displacement. The QSR comprehends the minimum unstabilizing constant force and displacement measures. Hence, QSR results in two scalar values. Note that in the case of a robot that can perform balancing actions, such as backward steps, these values are converted to the minimum force and displacement that induces the system to perform a complex balancing routine.

3.2 Robot properties

3.2.1 Absorbed energy

The Absorbed Energy during Impulsive perturbations (AEI) defines the capability of a robot to absorb energy during impacts. The AEI indicates the capability of the robot to oppose an impact and is expressed by the percentage of energy absorbed. Being a PI related to impulsive shocks, the parameters that describe the AEI are I and E . The result is a three-dimensional plot in which I lies on the x-axis, E is on the y-axis, and the percentage of energy absorbed by the robot is on the z-axis (Figure 2E).



3.2.2 Excited equivalent impedance

The Excited Equivalent Impedance (EEI) evaluates the dynamic behavior of a robot when subjected to repetitive disturbances. The EEI considers a simplified standard model and computes the dynamic coefficients of inertia (J), elasticity (K), and damping (B), varying f . The estimations of these parameters rely on the measurements from repetitive force perturbations. Using a dynamic regressor, we compare the robot to a second-order inertia-spring-damper system and evaluate the coefficients [J, B, K]. The EEI is a plot with f on the x-axis and the impedance coefficients on the y-axis (Figure 2F).

3.2.3 Normalization factor

Performance Indicators describe the balancing skills of systems under different loads. PIs are expressed by extensive measures (such as forces and displacements); therefore, these indices are highly dependent on the robot's size. As a result, to compare different systems, it is necessary to scale all measurements to a common reference model. Any tested robot could be used as a standard for all other systems. However, we believe it is better to refer to a more general model.

Since robots are designed to mimic human behaviors, we compare their performance to a medium-sized human.

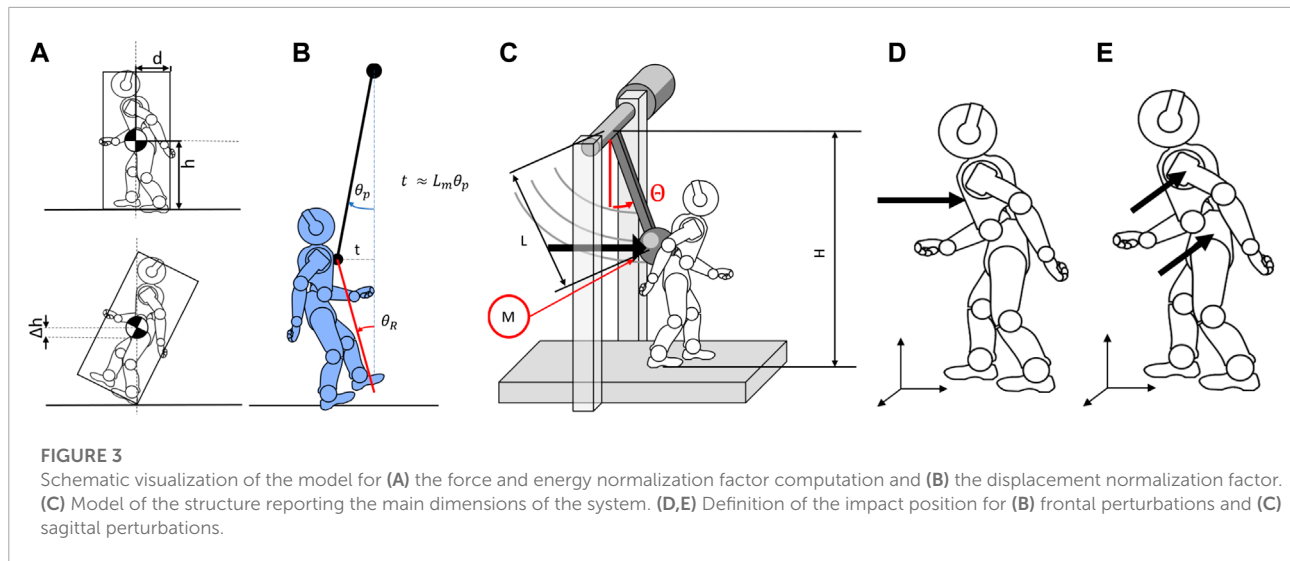
Dimensions for the human model are retrieved from [Armstrong, 1988](#). Using these values as standard dimensions, we designed some normalization factors that weigh all the previous indicators on the common model.

We define a total of six Normalization Factors: two are related to force scaling (i.e., frontal and lateral directions), two are related to the energy scaling (as before, frontal and lateral), and the last two are related to the CoM displacement. Normalization factors for energy (N_{E_f}) and force (N_{F_f}) have been calculated considering the systems rigid and approximable to parallelepipeds (Figure 3A). The force and energy normalization factors are computed as the minimum force and kinetic energy required to unstabilize the system by pivoting around one of its edges (see Figure 3A). Hereafter, we compare these values to the ones of our reference system. The normalization factors related to the force are

$$N_{F_f} = \frac{m_{robot}}{m_{human}} \frac{d_{CoM_{robot}}}{d_{CoM_{human}}} \frac{h_{CoM_{human}}}{h_{CoM_{robot}}}, \quad (1)$$

while the normalization factors related to the energy are

$$N_{E_f} = \frac{m_{robot}}{m_{human}} \frac{\sqrt{h_{CoM_{robot}}^2 - d_{CoM_{robot}}^2} - h_{CoM_{robot}}}{\sqrt{h_{CoM_{human}}^2 - d_{CoM_{human}}^2} - h_{CoM_{human}}}. \quad (2)$$



For both equations, m_i are the masses of the robot and human, h_{CoM_i} are the height of the Center of Mass from the ground, and d_{CoM_i} are the width of the bearing surface. The d_{CoM_i} dimension is different if the normalization factor is computed frontally or laterally. Subscript J indicates that the same equation holds both directions. Normalization factors are calculated by the minimum impulsive force and energy (respectively) that unstabilizes the model by pivoting it on its edge. Lastly, the normalization factor related to the displacement (N_D) compares the angular movement of the contact point relative to the robot ground, scaling it to the movement the system would have with the dimensions of a medium-sized person (Figure 3B).

$$N_D = \frac{h_{CoM_{robot}}}{h_{CoM_{human}}} \quad (3)$$

3.3 Performance indicators illustration

In this paragraph, we aim to illustrate the PIs behavior when applied to a generic robotic system. Functional to the visualization of the PIs is a dynamical examination of an actuated inverted pendulum subjected to external perturbations (Figure 2A). This example model is chosen because it resembles the dynamics of a humanoid-legged robotic system subjected to pushes when no steps are performed, allowing us to consider the feet/base of the robot fixed on the ground. Therefore, this model is similar to a basic self-balancing robot performing “ankle strategy” (Stephens, 2007; Rogers and Mille, 2018). Different models can be applied if we aim to resemble two-wheeled humanoid robots, which typically act as cart poles. The reader should note that the choice of the model does not influence the effectiveness of the PIs, but it may vary the behavior shown.

The model dynamics for the inverted pendulum is

$$J\ddot{\theta} + Mg\frac{l}{2}\sin(\theta) = \tau - Fh, \text{ with } \tau = -K_{lqr}[\theta; \dot{\theta}], \quad (4)$$

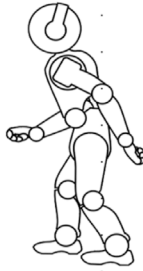
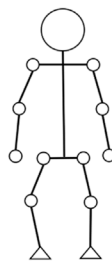
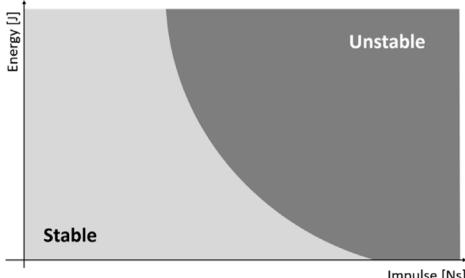
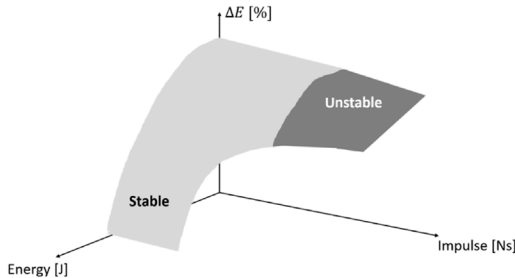
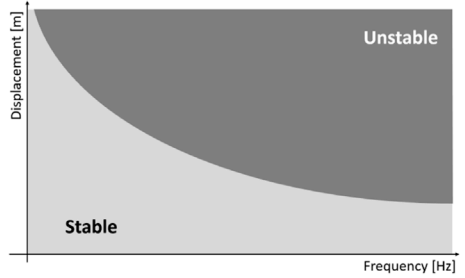
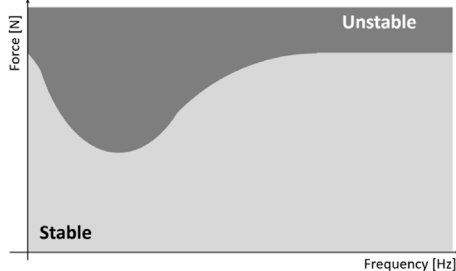
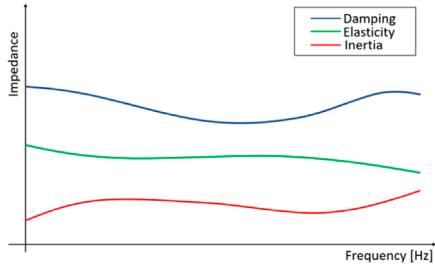
where J, M and l are the inertia, the mass, and the length of the inverse pendulum, respectively. τ is the commanded torque used to apply an LQR optimal control K_{lqr} . Lastly, F is the external force, and h is the height at which we apply the perturbation. During simulations, we adjust the input function $F(t)$ to match the different types of desired loads. Hence, the conditions under which a robot falls indicate the system’s resilience. In our model, we define the falling condition as the angle θ exceeding the limit θ_{max} . Saturation of the maximum torque makes the systems more similar to real robotic systems.

3.4 Robot resilience datasheet

Table 1 presents a datasheet that we propose to summarize the balancing performance of a robot. We hope that a datasheet can be a helpful add-on to foster system comparisons and regulation of processes.

The datasheet is organized as follows. The first and second lines contain the name and the type of the robot (e.g., legged humanoids, quadrupeds, exoskeletons) under testing. The successive 2 cells show a photo of the system and a scheme of its kinematics. The cell “Actuation” defines which types of actuators the system is built with (actuation units can be rigid, SEA, VIA, etcetera (Vanderborght et al., 2013)). “Robot Parameters” provides the main dimensions of the robot used to define the normalization Factors reported in the related cells. “Impact Position” and “Orientation” define the experimental conditions at which the experiments are executed. *Impact Position* describes the contact point location, while *Orientation* indicates if the PIs are related to the frontal or lateral perturbation on the robot.

TABLE 1 Example of datasheet containing the resilience benchmark results.

Robotic System Name				
(Humanoid/Legged System/Exoskeleton)				
		Actuation	(rigid/via/vsa/sea)	
		Robot Parameters		
		Mass	VALUE	
		Base Depth	VALUE	
		Base Width	VALUE	
		CoM Height	VALUE	
Robot Height	VALUE			
Impact Position		(CoM/Distal)	Orientation	(Frontal/Lateral)
Controller		(LQR, Whole-Body) with Reference		
Normalization Factors		IR		
Force	Frontal	VALUE		
	Lateral	VALUE		
Energy	Frontal	VALUE		
	Lateral	VALUE		
Displacement	Frontal	VALUE		
	Lateral	VALUE		
QSR				
Displacement	VALUE			
Force	VALUE			
AEI		ER- Displacement		
				
ER-Force		EEI		
				

Controller means which control is applied to the robot, and a reference on the related paper is strongly recommended. The other cells (QSR, IR, AEI, ER-Displacement, ER-Force, and EEI) report the PIs of the robot under the described conditions.

4 Test-bench design and characterization

The definition of the PIs and testing conditions provides a set of characteristics the system must possess. In Monteleone et al., 2020, we introduce the early concept definition, focusing on five design features: flexibility, reproducibility, adjustability, independency as a system, and operator safety.

4.1 Mechatronic design

Figure 4A shows the structure. The testbed is composed of two parts. The first is the central span, consisting of the actuation group and the pendulum. The second part is the external structure, mainly composed of aluminum extrusions and safety nets.

Figure 4B shows the cross-section of the actuation unit and its main components. From left to right, there is a servomotor (1), a torque sensor (2), an electromagnetic clutch (3), the pendulum connector (4), an electromagnetic brake (5), and two absolute position sensors (6–7). In component 4, we mounted the pendulum bar. We placed a piezoelectric sensor on its tip to measure the contact force between the actuation group and the robot under test. To prevent misalignments of the torque sensor, we connect it to the actuator through an Oldham joint and to the clutch through an elastic component. The compliant joint also absorbs accidental shocks transmitted to the torque sensors.

During impulsive tests, the clutch safeguards the servomotor and torque sensors. It disengages the pendulum shaft from such delicate parts, preventing the transmission of shocks. After an impact, the brake permits the system to attenuate oscillations. It also improves testbed safety by halting the pendulum in the case of emergencies.

The external structure protects the operators during test execution. We enclosed the test platform in an area accessible by two doors and surrounded by safety panels. Doors equip two electromagnetic locks. As a result, the control system can detect the status of gates (open or closed) and lock them, limiting access to the experimental area when the pendulum is moving. If the doors are unexpectedly released, the system activates the brake while simultaneously disengaging the clutch, preventing the pendulum from moving while safeguarding the motor.

Moreover, we provide an emergency button, which stops the pendulum movement in the same way as if the door opened. Figure 4C shows a picture of the physical structure, also displaying the placement of the control system (Figure 4D).

To enhance tunability, the system is equipped with components that can vary its structural dimensions. The testbed is provided with an electrical crane and four guides to change the position of the central span. Hence, it is possible to modify the height at which the pendulum impacts the robot (H). Furthermore, a two-part connector links the pendulum bar to the shaft, making it simple to vary the pendulum length (L). Lastly, we ensure that additional masses (M) can be mounted on the pendulum to increase its inertia. Table 2 reports the tunable parameters and their range of variation. We equipped the testbed with a modular floor with a series of holes equidistant from each other. This design improves the structure compatibility with other testing devices, such as treadmills or inclined planes. It also allows for the placement of obstacles to test robotic systems on uneven terrains.

Table 2 reports the main characteristic values of the testbed. Friction torque has been computed experimentally.

4.2 Control architecture

The framework is equipped with an industrial PC that is ROS compatible and three drivers for the servomotor, clutch, and brake, respectively. The IPC supervises the structure framework, generating the control inputs that are communicated to drivers. Moreover, it also acquires data through an integrated DAQ system from National Instruments (NI) (Figure 4D).

Figure 4E describes the control architecture scheme, showing each block and its physical connections. For the actuation unit components, each number corresponds to the ones shown in Figure 4B. Blue arrows indicate the power connections, black arrows show the control communication network, and purple and green arrows depict the sensor connections for the control loop and the data record, respectively.

The actuation unit can be controlled both in position or torque loop. The position control loop uses the measurements of the Renishaw absolute encoder (6) located at the output shaft. However, if the clutch is disengaged, the system relies on the servomotor encoder to move the actuator and reset the zero position of the motor control. The torque control loop relies on the measurement of the FUTEK torque sensor (2). These measurements are corrected by gravity compensation, so if we command a force trajectory, the torque measures reject the pendulum weight, providing the correct movement at the contact point.

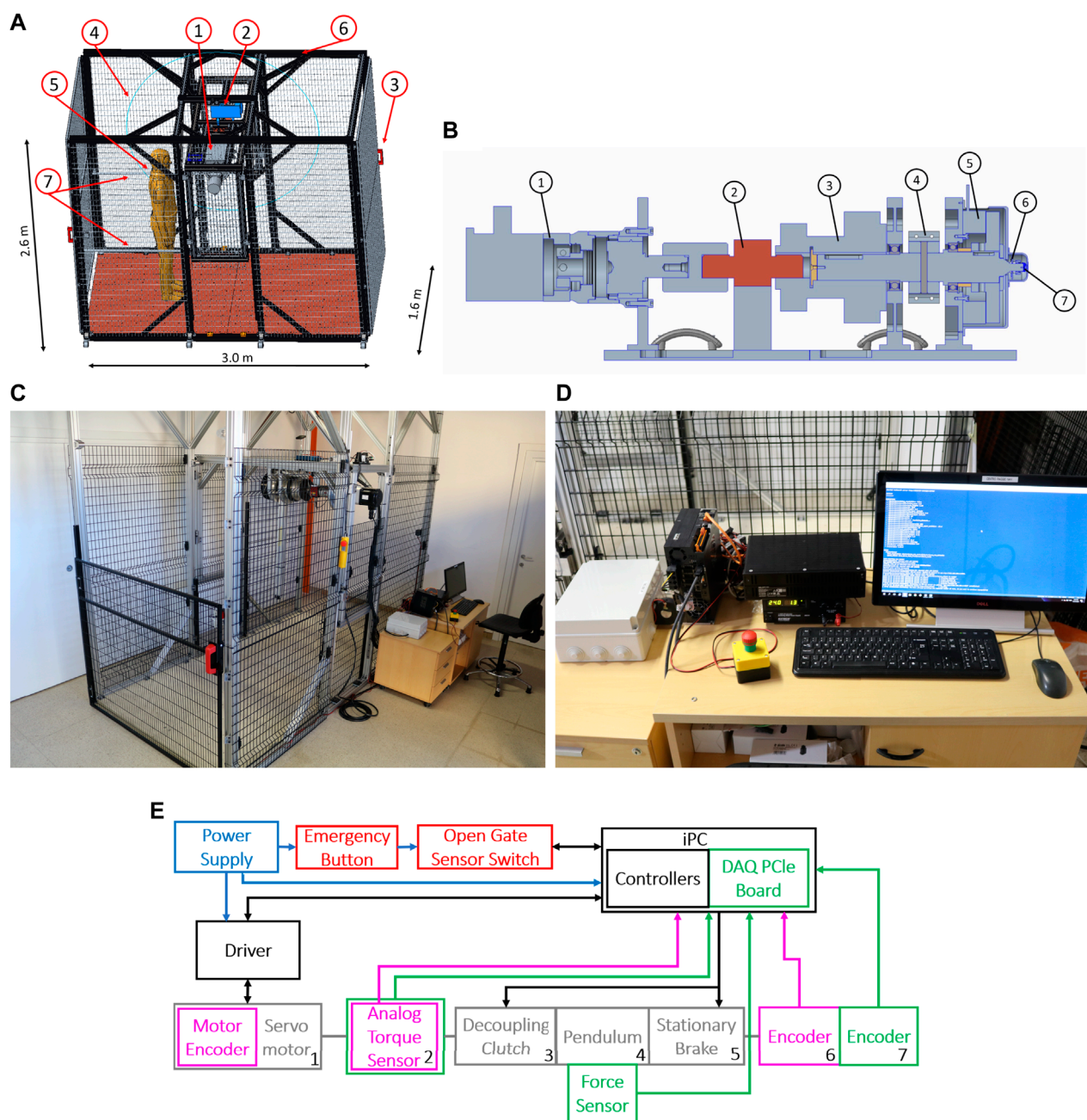


FIGURE 4

Photo of (A) Test-Bench dimensions and components: 1) Actuation group, 2) electrical crane, 3) sensorized safety doors, 4) safety panels, 5) agent positions during tests, 6) external structure, and 7) handrails. (B) Cross section of the actuation unit: From left to right, 1) servomotor, 2) torque sensor, 3) electro-magnetic clutch, 4) pendulum connector, 5) electro-magnetic brake, 6) first position sensor (absolute), 7) second position sensor (relative). (C) Testbed, and (D) industrial PC and controllers. (E) Electronic Connection of the Test-Bench. Blue arrows indicate the power connections, black arrows are the control communication network, and purple and green arrows describe sensor connections for the control loop and the data record. In red, we indicate the safety systems.

4.3 Data recording

The IPC saves data from experiments through a National Instrument data acquisition device. We acquire data from three sensors. The first is the pendulum encoder (number 7

in Figures 4B, E), which is an AMS absolute encoder with a resolution of about $.1^\circ$. The second is the force sensor. It is a DYTRAN 1051V6, a piezoelectric sensor capable of precisely measuring impacts and impulsive forces. However, when subjected to constant or slowly varying forces, it does not

TABLE 2 Test-bench characterization.

Tuning parameters	Range		Steps
Pendulum bar length	.5 ÷ 1.5 <i>m</i>		0.5 <i>m</i>
Pendulum axis height	1 ÷ 2 <i>m</i>		1 <i>cm</i>
Additional masses	0 ÷ 15 <i>kg</i>		0.5 <i>kg</i>
pendulum position	−90° ÷ 0°		.5°
Friction Experimental Estimation			
	Pendulum length		
	0.5 m	1.0 m	1.4 m
Friction torque (τ_f)	2.3 <i>Nm</i>		
Impulsive tests (Protocol 1)			
Maximum force	1200 N		
sinusoidal tests (Protocol 2/3)			
Maximum force	700 <i>N</i>	350 <i>N</i>	250 <i>Nm</i>
Maximum torque	350 <i>Nm</i>		
Maximum angular speed	100 <i>rpm</i>		
Maximum oscillation frequency	7 <i>Hz</i>	5 <i>Hz</i>	3 <i>Hz</i>
quasi-static tests (Protocol 4/5)			
Maximum force	700 <i>N</i>	350 <i>N</i>	250 <i>Nm</i>
Maximum torque	350 <i>Nm</i>		
Maximum angular speed	100 <i>rpm</i>		
sensors resolutions			
torque sensor force resolution	1 <i>N</i>	0.5 <i>N</i>	0.4 <i>N</i>
force sensor resolution	0.8 <i>N</i>		
angular encoder resolution	.09°		

perform correctly due to drift. It has a resolution of 0.3 N and a saturating value of 2224 N. The third one is the torque sensor. It is a FUTEK FSH02060, an analog sensor that measures non-impulsive forces using strain gauge technologies. The resolution of the torque sensor is 1 Nm, and its maximum measurable value is 500 Nm. We use the force sensor during impulsive tests to compute performance indicators. In contrast, during the other tests, we estimate the force exerted on the pendulum, knowing the distance of the pendulum tip to the torque sensor axis, and correcting the measure with a gravity compensation in post-processing.

The testbed acquires all data at a frequency of 10 kHz. Position and torque measurements are filtered by excluding outliers and using a symmetric moving average filter. The data from the piezoelectric sensor is not filtered because filtering would result in a loss of accuracy on the force peaks. From the force data, it is possible to identify the exact moment of an impact. However, to measure the value of the impulse, there

exist two methods. The first technique estimates the duration of the contact between the robot and the pendulum and integrates the force value. The second method relies on measuring the pendulum velocities before and after the moment of touch. It evaluates the impulse as the variation of the momentum. We saw experimentally that the second one resulted in being more reliable, as the definition of the contact duration is not trivial.

5 Experimental protocols

To measure the PIs, we developed a series of testing methods that allow the reproduction of the necessary perturbations. In the following, we define each protocol and report the detailed procedure to perform the experiments. This work focused on the definition of resilience against pushes on regular, obstacle-free terrains. The possibility of studying the effects of different terrains on the performance is left to future works.

During a protocol execution, we repeat each experiment (we call “experiment” tests with the same set of conditions) 10 times (we call each one a “run”). With this, we aim to provide the results with a certain degree of statistical validity. Indeed, we performed a high number of experiments to provide a more reliable view of how the system reacts to perturbation with a given entity. Since the system is physical, borderline values of perturbation can lead to a robot falling or not depending on other robot conditions (e.g., if the robot is impacted while the pitch angle is positive or negative). Therefore, the high number of experiments considers the fall’s statistical validity, reducing the effect of outlier situations.

We measure the pendulum angle, the torque at the motor axis, and the force at the contact point with the robot. These measures are used to obtain all the performance indicators in [Section 3](#). The force sensor employs piezo-electric technologies, allowing one to appreciate the quick variation of forces, such as peaks. On the contrary, since the torque sensor is resistive, it is more suitable to evaluate constant or slow-vary forces.

At the beginning of each protocol execution, we must adjust the structure to impact the system at the desired contact point. For frontal collisions, the designated point should be placed at the center of the chest, on the robot axis, the closest to the CoM as possible ([Figure 3D](#)). The height of the contact point is measured and saved by the platform. For side impacts, the contact point should be located on the shoulder or hip, typical contact points during accidental collisions ([Figure 3E](#)). Aside from the contact point, lateral experiments execution uses the same experimental protocols as the frontal experiment. Therefore, the following section will not further distinguish between frontal or lateral protocols. The control sets the end of the experiment when it detects that the pendulum has reached the maximum height and there has been a speed inversion or when it is motionless.

5.1 Protocol I: Impulsive disturbance protocol

The first protocol aims to assess the balancing performance of systems subjected to impacts. Impulsive loads are obtained by raising the pendulum at the desired height and successively letting it free to fall.

In the Charpy test, impacts are defined by the energy involved in the experiment (François and Pineau, 2002). Moreover, impacts can always be described by the value of forces or impulses exchanged between objects (Stronge, 2018). Hence, we decided to define impulsive tests based on the value of both impulse (I) and the initial energy (E). We discovered that these parameters could be treated as two independent values through analytical computations and experiments. To obtain the desired values during experiments, we tune the pendulum length (L), the initial position (θ), and pendulum inertia (M) (see Figure 3C). We related the pendulum parameters with potential energy and the impulse exerted on the system during an impact. Parameters of impulsive tests are defined by

$$\begin{cases} E = \left(M + \frac{\delta L}{2} \right) gL (1 - \cos \theta) \\ I = \frac{1}{L} \sqrt{2 \left(ML^2 + \frac{\delta L^3}{3} \right) \alpha E} \\ \alpha = 1 - \frac{\tau_f \theta}{E}; \end{cases} \quad (5)$$

Among them, α indicates the percentage of energy not lost due to friction and is computed experimentally by estimating the energy loss between initial height and impact height. τ_f represents the friction torque, which is assumed to be constant. δ expresses the linear density of the pendulum bar, and g is the gravity coefficient. These equations consider that the pendulum stops after the impact and that the impact occurs between rigid bodies, so there is a slight difference between theoretical and experimental values. The experimental procedure for the protocol I is reported in Table 3. It is worth noticing that H , L , and M change in response to the values of $[E, I]$, whereas the others have a fixed value for each robot under test.

The medium execution time for each run is 2/3 s. The data collected during the experiments are used to compute the Performance Indicators described in Section 3. In particular, this protocol aims to find the IRand the AEI.

Defining the parameters that describe an impact requires a preliminary testing phase on a mock model. The mock model comprises an inverted pendulum structure and a small base to stabilize it. Following an impact, we let the system fall to reduce the residual noise on the force sensor.

As a preliminary couple of parameters, we selected the energy and the maximum exerted force ($[E, F_{max}]$), assuming an impact lasted for a constant time. However, the assumptions resulted in being incorrect. Indeed, we experimentally observed that E influenced the impact duration (Figure 5D) and, consequently, F_{max} (Figure 5C). Conversely, the experiments

show no correlation between E and the impulse (I) applied to the robot. Figure 5A, B show the theoretical values of I and E computed using Eq. 5 and the one resulting from experiments. We appreciate how assumptions about shocks (rigid impulse and stationary pendulum after impacts) generate a plot in which I values are scaled by a medium scale factor of .59.

Figure 5A highlights in black the physical constraints of the system. Points outside the demarcated area are not feasible due to the range of possible pendulum inertia.

Moreover, values below the straight line are not recommended because the friction action consumes most of the energy during the pendulum swing.

5.2 Protocols II & III: Sinusoidal protocols

The purpose of the second and third protocols is to assess the balancing performance of systems subjected to periodic perturbations. Repetitive perturbation are given with a controllable position amplitude (A_D for displacements, A_F for forces), frequency (f), and number of cycles (n_C) in the form

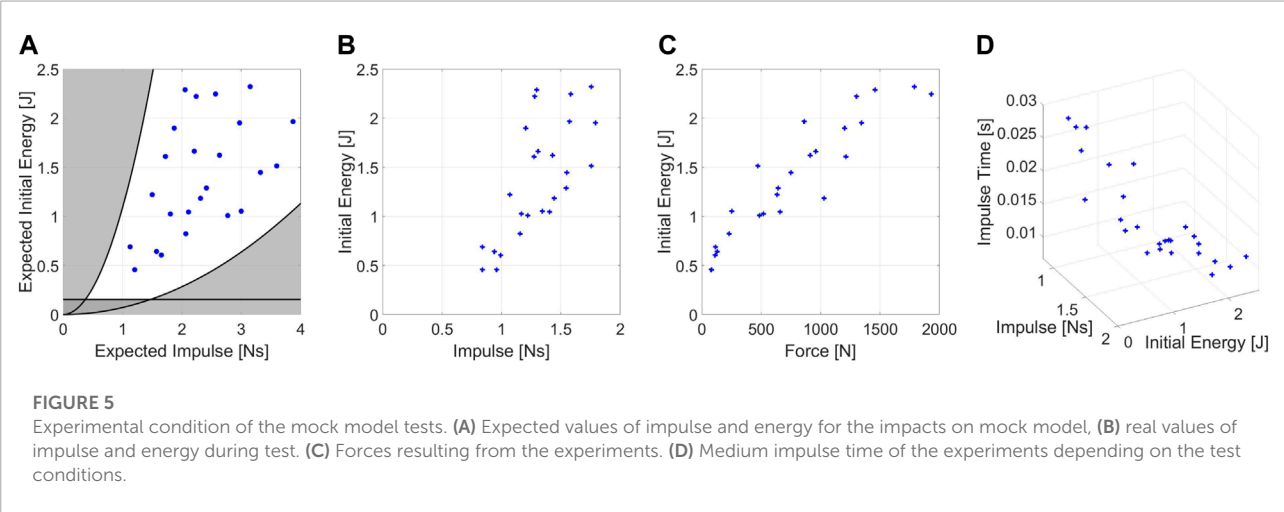
$$\begin{aligned} D(t) &= A_c + A_D + A_D \sin(2\pi f t - \pi/2), \\ F(t) &= F_c + A_F + A_F \sin(2\pi f t - \pi/2) \end{aligned} \quad (6)$$

where A_c is the position in which the pendulum starts to contact the robot, and t is the time. A_F is the force amplitude, and F_c is a small force to ensure that the robot will keep contact during the execution of the experiment. The first equation is related to protocol II, while the second one is to Protocol III. A position perturbation does not ensure that the contact lasts during all the experiment execution since we command the position of the pendulum to follow a specified path.

Table 3 reports the experimental procedure for protocol II & III. For Protocol II, the operator must stop the experiment if the robot falls. In contrast, for protocol III, if the robot falls, the framework will detect that contact with the robot is lost and consider the experiment finished. Experimental conditions ($[A_D, f]$ for protocol II, $[A_F, f]$ for protocol III) are gradually increased until the robot falls. The execution time heavily depends on the f at which the experiment is executed. The result of sinusoidal protocols is the EROf the system. Moreover, sinusoidal force protocol aims to define also the EEI, since the contact lasts along all the run execution. The main limitations are the maximum allowable frequency and amplitude during test execution. The maximum force and frequency depend on the characteristic of the actuation unit. The maximum displacement is a function of f (as it is related to the maximum allowable speed at the servomotor side) but also depends on the dimension of the robot. Indeed, since a displacement along the perpendicular direction corresponds to a height variation, the contact point should never exceed a safe height variation to avoid the system impacting sensitive parts of the robot, such as the head.

TABLE 3 The table shows the procedures for the testing protocols. If the steps are different, we divided it for each protocol.

Protocols procedure			
Steps	Protocol I	Protocol II & III	Protocol IV & V
1	Set up H , L , and M	Set up H and L	
2	Place the agent at the desired experiment position		
3	Activate the protocol		
4	Pendulum raises at desired position	Pendulum reaches contact point with the robot	
5	Data acquisition is started		
6	Pendulum performs the desired perturbation		
7	When experiment finishes, data acquisition stops and the pendulum moves to a resting position		
8	Operator reports if the robot is fallen, so that data can be saved		



5.3 Protocol IV & V: Quasi-static protocols

The fourth and fifth protocols aim to assess the resilience of systems subjected to constant or quasi-static perturbations. We provide slow varying perturbation $D(t)$ and $F(t)$, with a dynamics of

$$\begin{aligned} D(t) &= A_c + V_d t, \\ F(t) &= F_c + V_f t \end{aligned} \tag{7}$$

where V_d and V_f are the small velocity at which we execute the experiments. Tests are defined based on the value of V_d and V_f at which experiments are executed. However, the slopes of ramps provided are fixed to avoid testing the robot under non-quasi-static conditions.

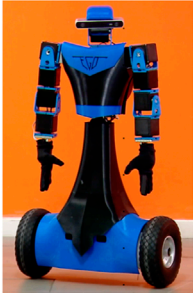
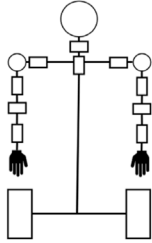
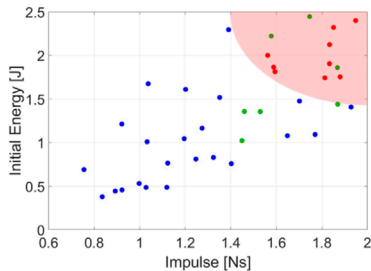
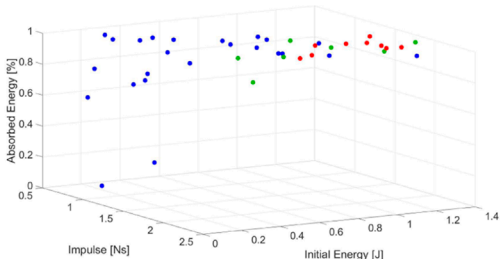
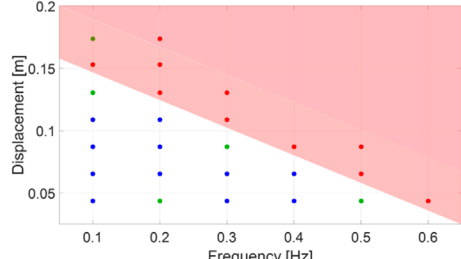
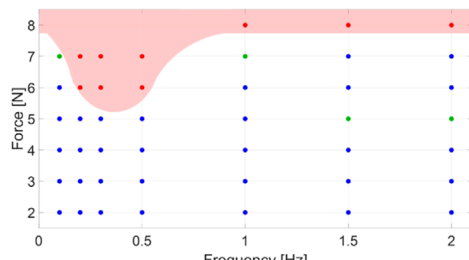
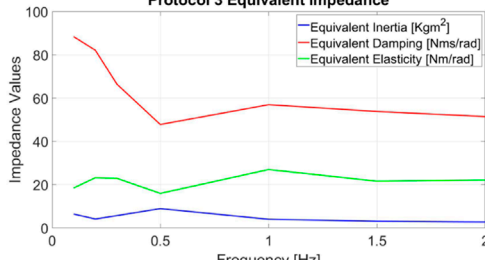
Table 3 reports the experimental procedure for protocol IV & V. The falling detection algorithm is also applied in the case of quasi-static protocols. The falling detection algorithm explained in protocol III is also applied for protocols IV and V since the

slow slopes ensure that contact is always present. Forth and fifth protocols are designed to measure the QSR. The major constraint is the maximum allowable displacement the system can do. The variation of height must be limited so that sensitive parts of the robot are kept safe. The possibility of studying the effects of different terrains on performance is left for future works.

6 Application example

To demonstrate the strength of our framework, we benchmark the performance of Alter-Ego, a robust and versatile mobile two-wheeled system with a functional anthropomorphic upper body (Lentini et al., 2019). The robot is equipped with an LQR optimal controller for lower body stability, while the upper body is controlled to stay in the rest position with a low level of stiffness. The robot is equipped with an integrated safety system (Zambella et al., 2020) that avoids breakages in case of falls. In

TABLE 4 Datasheet resulting from the experiments on AlterEgo.

Alter-Ego			
(Two-Wheeled Humanoid Robot)			
		Actuation	VSA
		Robot Parameters	
		Mass	26.0 kg
		Base Depth	0.26 m
		Base Width	0.57 m
		CoM Height	0.41 m
		Robot Height	1.0 m
Impact Position	CoM	Orientation	Frontal
Controller	LQR - control (Lentini et al. (2019))		
Normalization Factors		IR	
Force	Frontal		
	Lateral		
Energy	Frontal		
	Lateral		
Displacement	Frontal		
	Lateral		
QSR			
Displacement	N.D.		
Force	N.D.	ER- Displacement	
AEI		EEI	
			
ER-Force			
			

the paper, we also report conditions at which the robot becomes unstable.

Tests are made on the frontal plane of the robot. Every test has the same contact point in the center of Alter-Ego chest, at the height of 80 cm from the ground. As stated in Section 5, experiments stop if the displacement of the pendulum exceeds the maximum allowable for the robot. The maximum permissible displacement of Alter-Ego was experimentally selected as 40 cm with a pendulum of 1 m length, corresponding to a height displacement of the contact point of around 8 cm. If necessary, the height displacement can be reduced using the 1.5 m pendulum bar.

Lateral experiments are not reported. The reason lies in the kinematics of AlterEgo that does not allow it to move laterally. No control can be applied in that direction. As a result, while performing Impulsive perturbations, the system acted rigidly until the impact was powerful enough to break the system. The robot resisted the external perturbation in the other protocols until the force was sufficient to lift the system. However, because we can easily calculate that force value analytically, we believed that a physical evaluation was unnecessary and would be detrimental to the robot integrity. Therefore, we decided to interrupt the lateral performance evaluation since continuously damaging the robot would have been too expensive.

We collect a number of 410 runs (41 different conditions) for protocol 1, 260 runs (26 different conditions) for protocol 2, 430 runs (43 different conditions) for protocol 3, and 10 runs each for protocols 4 and 5 for a total of 1120 experiments. A full testing procedure required around 4 days for frontal experiments. In all the figures related to performance indicators, we indicated with blue dots the conditions at which the robot does not fall, with green points conditions at which the robot falls beneath 30% of times, and with red dots conditions at which the robot falls with a statistical percentage above 30%. Table 4 reports the results of the experiment in the form of datasheet, as presented in Section 3.

In the attachment to the paper, we present a video showing examples of the execution of tests.

6.1 Protocol I: Impulsive protocol

Figure 6 displays the photo-sequences of two experiments performed on Alter-Ego. The first one shows the system resisting an impact (Figure 6A), while the second shows the system failing to balance itself (Figure 6B). Impacts on the systems result in a variation in the pitch of the robot. If the pitch variation is too fast or too extended, the system cannot balance itself, failing. Tests on Alter-Ego have been executed with the conditions shown in Figure 7A. Each set of conditions ($[E, I]$) corresponds to a specific value of $[L, M, \theta_i]$ for the experiment. To test the system, we performed the protocol described in Section 5.

Figure 8 shows an example of angular position and force measurement during an impulsive test. Blue data indicates the raw data coming directly from the DAQ system, while we highlighted the filtered data in orange.

Figures 7C, F shows the IRand the AEIof Alter-Ego, respectively. AEIshows the capability of the robot to return energy to the pendulum in case of impacts with low $[E, I]$. In unstabilizing impacts, however, the robots absorb most of the energy, which becomes kinetic energy and plastic deformation of the covers. Figure 7C shows that it is possible to describe a region of conditions at which the robot cannot absorb and withstand the shock, validating our theory. In this graph, each set of data collected have a medium standard deviation from the mean value shown of around .37Ns, and therefore possess a certain degree of repeatability. Figure 7D reports the relation between E and F_{\max} during impact. The graph shows an almost linear relation between those parameters, confirming the same results achieved with the mock model. Figure 7E shows the relation between impact conditions and impulse duration. This picture also confirmed the behavior exhibited by the mock model. Although it may seem obvious, this behavior is worth reporting. Indeed, the fact that Alter-Ego possesses more complex internal dynamics than the mock model does not change the considerations made about force and impulse time. Then, we can assume that this behavior holds for other robots, validating the choice of $[E, I]$ as describing parameters for the experiment.

6.2 Protocol II: Sinusoidal displacement protocol

We began the experiments with the set of conditions $[A_D, f] = [4 \text{ cm}, 0.1 \text{ Hz}]$, gradually increasing them in ranges that goes from 4 cm to 17 cm within 7 steps for A_D , and from 0.1 Hz to 0.6 Hz within 6 steps for f .

Figure 9 shows an example of measurements for sinusoidal displacement perturbations. Both position and force measurements required filtering noise and outliers. The reason behind the raw data drift lies in the absence of gravity compensation, which is adjusted during filtering. Figure 9C shows the ERrelated to displacement perturbations of the robots. The mean standard deviation that those data possess from their related medium value is 0.2 mm.

ERdepicts the relationship between the system capability to resist recurrent disturbances to their oscillation frequency. Higher frequencies in the position perturbation domain correspond to faster movements of the systems. Rapid perturbations result in being more unstabilizing than large displacements.

Analysis of the measurements deriving from this protocol shows a high degree of repeatability on the experimental conditions. Moreover, the procedure defined in Section 5 resulted in being simple and straightforward.

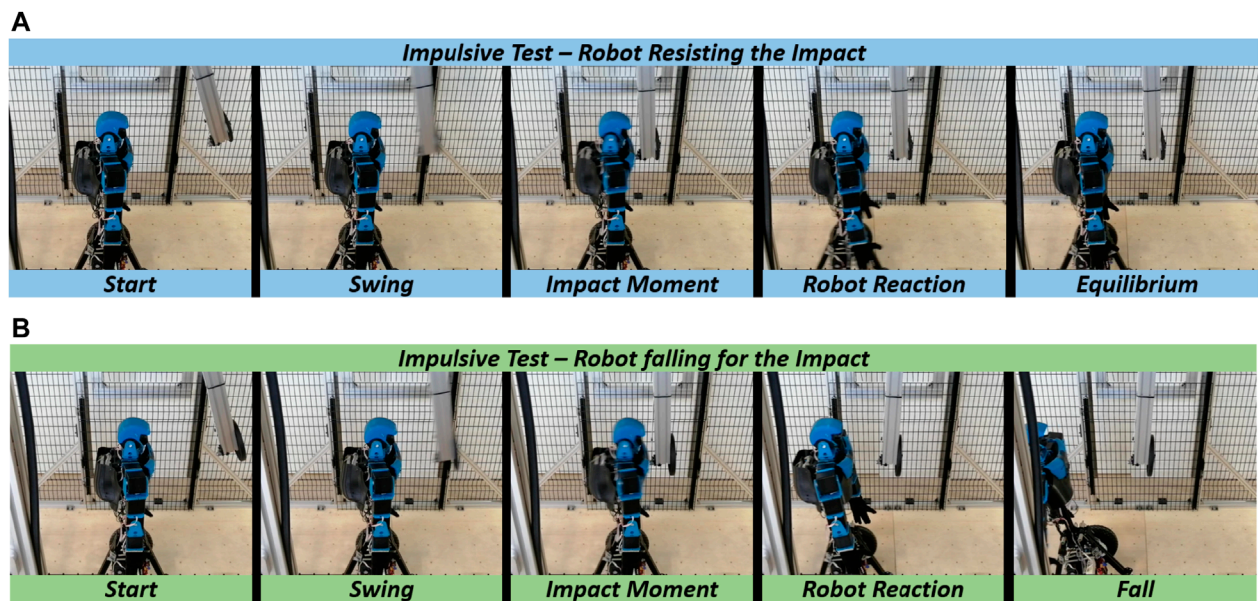


FIGURE 6
Photosequence of impulsive tests for (A) robot withstanding the impact, and (B) robot falling due to the impulse.

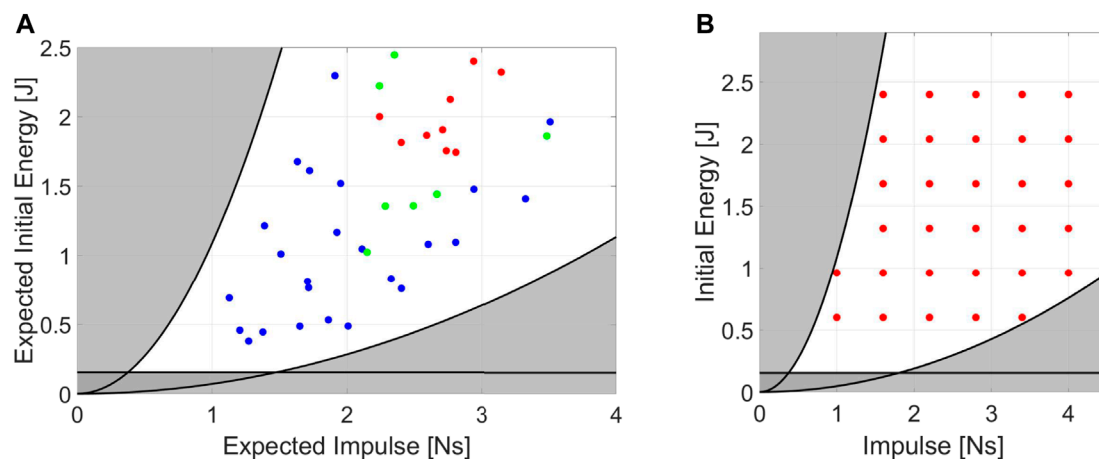


FIGURE 7
(A) Set of desired conditions at which we perform the experiments on Alter-Ego. Each point represents a set of $[E, I]$ related to 10 runs. (B) Standardization on the selection of the impulsive experiments.

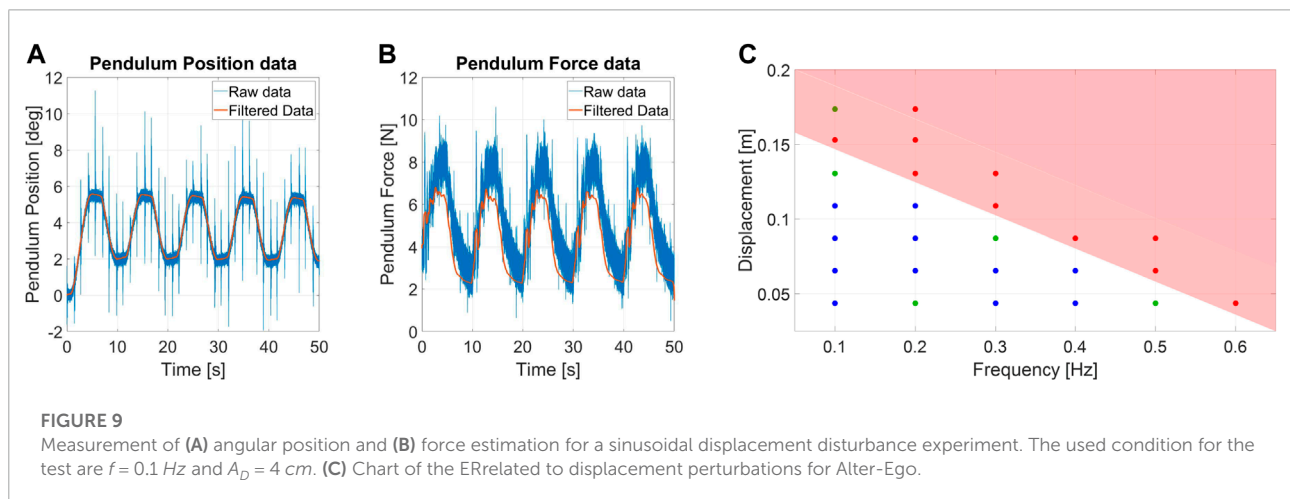
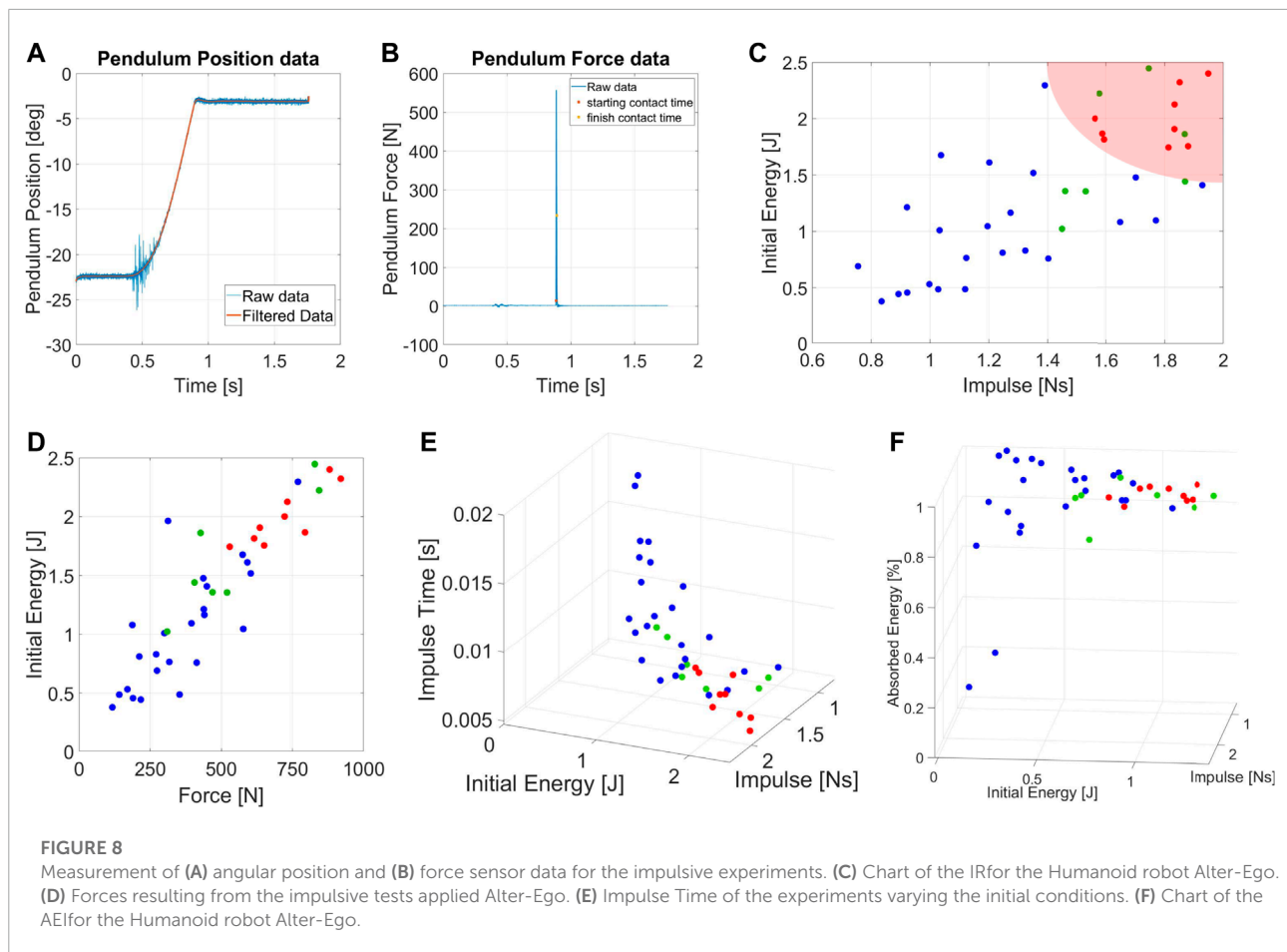
6.3 Protocol III: Sinusoidal force protocol

We began the tests with the set of conditions $[A_F, f] = [2N, 0.1Hz]$, gradually increasing them with steps of 1N for A_F , and with a span of $f = [.1, .2, .3, .5, 1.0, 1.5, 2.0]$. Experiments are performed with a $n_C = 5$.

Figure 10 shows a set of measurements for sinusoidal force perturbation. During filtering of data, we took into account the effect of gravity on the torque

measurements, and we compensated it to obtain the force exchanged between the structure and the robot.

Figure 10E shows the ERof Alter-Ego, while Figure 10C describes the EEI. For this set of data, the mean standard deviation is around 0.4N from their meadium value. The system is approximated to a second-order system (mass-spring-damper, see Section 3), and the equivalent coefficients are computed for each frequency.



ER reveals a low-frequency force range which is more destabilizing for Alter-Ego. Its dynamics act as a low pass filter, better rejecting high-frequency perturbations. In case of repetitive displacement perturbations, the pendulum provides faster and stronger pushes at higher frequencies, resulting in the robot that eventually falls when the frequency exceeds

a definite value. Conversely, in the case of repetitive force perturbations, pushes act more like vibrations than perturbations at increasing frequencies, resulting in the robot rejecting these disturbances better than at lower frequencies. EEI shows a system with almost constant inertia and elasticity while the damping lowers at higher frequencies. Being the impedance an extrinsic

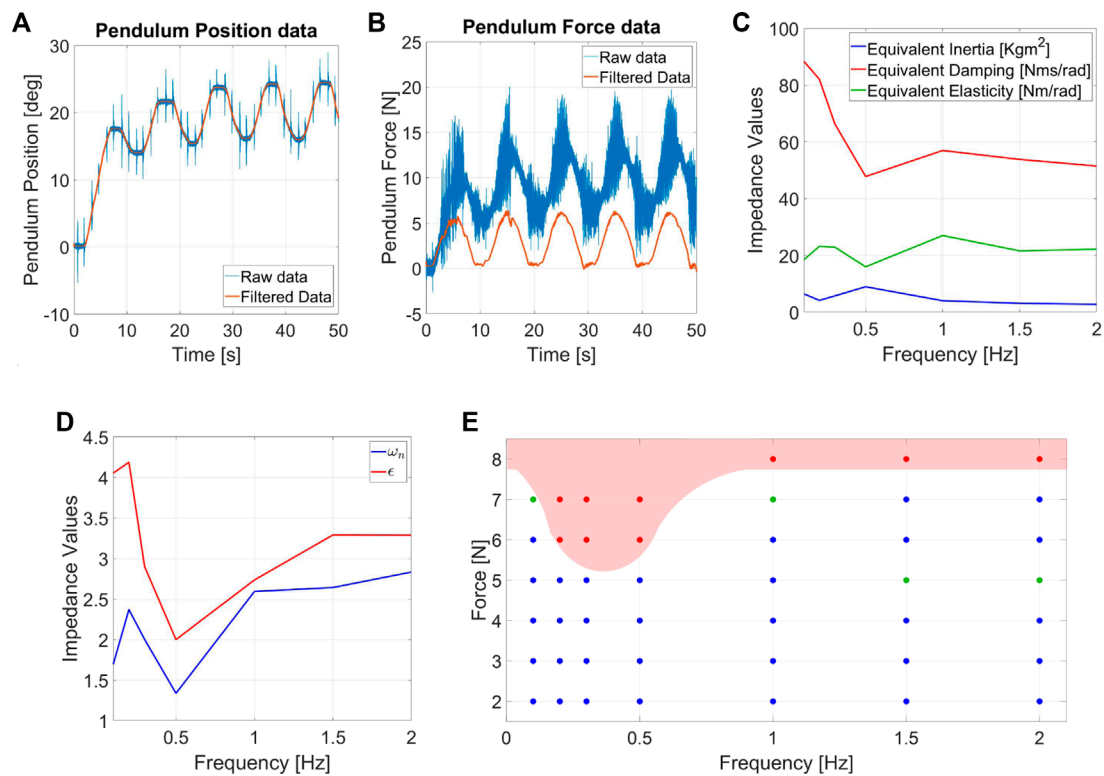


FIGURE 10

Measurement of (A) angular position and (B) force estimation for the sinusoidal force disturbance experiments. The used condition for the test are $f = 0.1\text{Hz}$ and $A_F = 3\text{N}$. (C) EEI and (D) Equivalent damping ratio (ϵ) and natural frequency (ω_n) related to sinusoidal force protocols. (E) Chart of the ER related to repetitive forces for Alter-Ego.

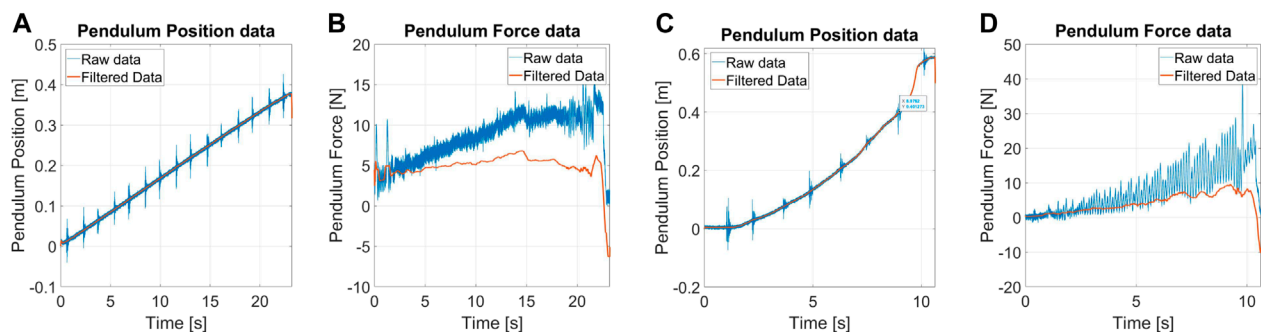


FIGURE 11

(A) angular position and (B) force estimation for the quasi-static displacement disturbance experiments, and (C) angular position and (D) force estimation for the quasi-static force disturbance experiments.

property, we also computed the damping ratio and the natural frequency, which are intrinsic properties instead (Figure 10D). Interestingly, a reduction in the damping ratio and natural frequency occurs in frequencies that are more destabilizing for the ER.

Experiments demonstrate the repeatability of testing conditions and point out how the force F_c (see Section 5.2) ensures the system maintains contact during the run duration. The protocol is straightforward, ensuring no training is necessary before performing these tests.

6.4 Protocol IV: Quasi-static displacement protocol

Tests are executed by providing a slow-varying ramp in the contact point position, with a slope of 1.5 cm/s . Experiments start after the pendulum contact algorithm and stop if the robot falls or exceeds the maximum allowable displacement. **Figure 11** shows an example of the measurements for protocol IV. The force measurements also show the gravity rejection from raw to filtered data.

During the experiments, the robot did not show any unstable behavior, so it was not possible to define the QSR. Alter-Ego acts like an inverted pendulum mounted on a Segway. The system is regulated by an LQR controller, with high weights on the pitch dynamics and low authority on the position. This design choice is because we want the system not to fall, regardless of the position. This behavior reflects on the QSR, since the robot moves away from the desired position under the action of external frontal pushes, as the main task is to maintain stability.

6.5 Protocol V: Quasi-static force protocol

Tests are executed by providing a slow varying ramp of 1 N/s . Experiments start right after the pendulum contact algorithm and stop if the contact is not preserved or the position exceeds the maximum allowable displacement.

Figure 11 shows examples of measurements for quasi-static force perturbation. As in the previous cases, we estimate the force through the torque sensor and correct the bias due to the pendulum weight. Oscillations of force are due to the friction created by sliding the pendulum tip on the robot covers. The robot did not show unstable behaviors, and, for this reason, it was not possible to define the QSR.

Regarding the discussion of the performance of Alter-Ego under quasi-static forces, the same considerations made on **Section 6.4** hold.

7 Discussion

We verified the efficacy of the benchmarking method by quantifying the balancing performance of Alter-Ego. Results reported in **Section 6** are promising in assessing systems resilience for many reasons. The experimental campaign exhibits a high degree of repeatability. Indeed, it was possible to perform a large number of experiments under the same conditions. Moreover, the standard deviation of each data set from the mean value shown in the PIs graphs is adequate to have a certain degree of statistical validity. As a result, we tested and thoroughly characterized the robot within a few days. The benchmark allows

for easily switching testbed conditions and control techniques during the experimental campaign. Efforts done by operators during the protocol selection and execution are minimal since the control routine and parameters can be chosen by software at the beginning of each test. Protocol I routine is the lone exception, requiring the operator to change the pendulum's inertia to obtain the appropriate conditions. To facilitate the procedure, we designed the system so that adding and removing masses is a simple process. The first calibration of the structure parameters (see **Section 4.1**) to match the dimensions of robots under testing requires a relatively low effort. H is adjusted by moving the structure using an electrical crane, making it a simple procedure. L and M are modified by changing the pendulum bar and adding masses. Moreover, tuning parameters is a preliminary procedure, and it is required to be performed once for a testing campaign (twice if experiments are performed both on the frontal and lateral planes) since the contact point is the same for all protocols. Lastly, the performance evaluation relies solely on the sensors integrated into the framework. Therefore, all the results are consistent, allowing us to compare different systems with a meaningful metric.

To improve the efficiency of tests, we define a method for selecting the optimal experimental conditions for the first protocol, allowing us to identify the experiments *a priori*. We used a mock model and Alter-Ego to validate the concept behind the experiment conditions. The reason behind this choice is that at least two systems must be used to ensure that this definition is reliable for most of the robots that will be tested. **Figure 7B** shows an example of conditions under which we should test the robot. The testbench automatically generates the required experiments that the operators must execute, indicating the necessary mass, pendulum length, and starting position in a matrix. By using normalization factors, the matrix is constructed by scaling a given set of initial values $[E, I]$ to the robot size. Protocols II and III, on the other hand, already have a straightforward procedure. The starting values of the amplitude are determined by the robot dimensions, especially for force ranges.

The perturbations considered in this work do not describe the totality of disturbances that can be applied to a self-stabilizing robot but the ones that are the most common to the best of our knowledge. In future works, we are planning to define more testing protocols to account for more perturbations. Some examples can be found in sudden forces and displacements that last over time, occurring when a robot impacts heavy external objects, or pseudo-random force signals, possible while interacting with external operators.

8 Conclusion

In our work, we investigated the stability characterization of robotic systems subjected to external perturbations. We propose

a benchmarking method for testing systems of different sizes with reliable and repeatable experiment conditions. To characterize robots performance quantitatively, we provide a set of protocols and performance indicators. The aim is to allow comparisons between different mechatronics solutions or the same system with distinct controllers. Finally, we propose a datasheet to summarize the balancing performance of robots resulting from experiments in our framework. We used the framework to characterize Alter-Ego, a two-wheeled robust humanoid robot, to evaluate the effectiveness of our benchmarking method. In this regard, we ran a campaign with 1120 tests. Quantitative evaluation of robot performance will promote the improvement of robots and push forward the standardization and regulation of these technologies.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: <https://www.naturalmachinemotioninitiative.com/benchmark-robot-balancing> Data can be found in download link on “Experimental data of AlterEgo characterization”.

Author contributions

All the authors contributed to the conceptual development of the framework. All authors conceived the study, and SM designed the mechanical system and developed the control framework with the expertise of FN and MG. SM designed the post-processing code of the system. All the authors contributed to the experimental design. SM and MG performed the experiments. SM executed literature research, and SM, GG, and MC analyzed the experimental data. SM wrote all sections of the articles with the constant insight of all the authors. AB contributed expertise and advice. All authors contributed to manuscript revision, read, and approved the submitted version.

References

- Armstrong, H. G. (1988). Anthropometry and mass distribution for human analogues, volume 1. military male aviators. *Aerosp. Med. Res. Lab Wright-Patterson AFB Ohio U. S. A. Tech. Rep.*
- Barasuol, V., Buchli, J., Semini, C., Frigerio, M., De Pieri, E. R., and Caldwell, D. G. (2013). “A reactive controller framework for quadrupedal locomotion on challenging terrain,” in 2013 IEEE International Conference on Robotics and Automation (IEEE), Karlsruhe, Germany, 06-10 May 2013, 2554–2561.
- Bayón, C., Delgado-Oleas, G., Avellar, L., Bentivoglio, F., Di Tommaso, F., Tagliamonte, N. L., et al. (2021). Development and evaluation of benchbalance: A system for benchmarking balance capabilities of wearable robots and their users. *Sensors* 22, 119. doi:10.3390/s22010119
- Berg, K. O., Maki, B. E., Williams, J. I., Holliday, P. J., and Wood-Dauphinee, S. L. (1992). Clinical and laboratory measures of postural balance in an elderly population. *Archives Phys. Med. rehabilitation* 73, 1073–1080.
- Ellis, R. G., Sumner, B. J., and Kram, R. (2014). Muscle contributions to propulsion and braking during walking and running: Insight from external force perturbations. *Gait posture* 40, 594–599. doi:10.1016/j.gaitpost.2014.07.002
- Feng, S., Xinjilefu, X., Atkeson, C. G., and Kim, J. (2016). “Robust dynamic walking using online foot step optimization,” in 2016 IEEE/RSJ International

Funding

This work is supported by the European Union’s Horizon 2020 research program under the projects Eurobench (No. 779963) and Natural Intelligence (No. 101016970). The content of this publication is the sole responsibility of the authors. The European Commission or its services cannot be held responsible for any use that may be made of the information it contains.

Acknowledgments

The authors would like to thank Vinicio Tincani, Cristiano Petrocelli, Mattia Poggiani from Istituto Italiano di Tecnologia and Alessandro Tondo and Fabio Bonomo from Qbrobotics for the support in developing the test-bench.

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.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2022.817870/full#supplementary-material>

Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea (South), 09–14 October 2016 (IEEE), 5373–5378.

François, D., and Pineau, A. (2002). *From Charpy to present impact testing*. Elsevier.

Fuchs, M., Borst, C., Giordano, P. R., Baumann, A., Kraemer, E., Langwald, J., et al. (2009). “Rollin’justin-design considerations and realization of a mobile platform for a humanoid upper body,” in 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009 (IEEE), 4131–4137.

Henze, B., Roa, M. A., and Ott, C. (2016). Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. *Int. J. Robotics Res.* 35, 1522–1543. doi:10.1177/0278364916653815

Hollnagel, E., Woods, D. D., and Leveson, N. (2006). *Resilience engineering: Concepts and precepts*. Farnham, UK: Lund Humphries Publishers Ltd.

Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C. D., Tsounis, V., et al. (2016). “AnyMal-a highly mobile and dynamic quadrupedal robot,” in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE), Daejeon, Korea (South), 09–14 October 2016 (IEEE), 38–44.

Hyon, S.-H., Hale, J. G., and Cheng, G. (2007). Full-body compliant human–humanoid interaction: Balancing in the presence of unknown external forces. *IEEE Trans. Robotics* 23, 884–898. doi:10.1109/tro.2007.904896

ISO, E. (2010). *148-1: Metallic materials-charpy pendulum impact test-part 1: Test method*. European Committee for Standardization.

Kanzaki, S., Okada, K., and Inaba, M. (2005). “Bracing behavior in humanoid through preview control of impact disturbance” in 5th IEEE-RAS International Conference on Humanoid Robots, 2005, IEEE, 301–305.

Kazerooni, H. (2005). “Exoskeletons for human power augmentation,” in 2005 IEEE/RSJ International conference on intelligent Robots and Systems, Edmonton, AB, Canada, 02–06 August 2005 (IEEE), 3459–3464.

Khazoom, C., Caillouette, P., Girard, A., and Plante, J.-S. (2020). A supernumerary robotic leg powered by magnetorheological actuators to assist human locomotion. *IEEE Robotics Automation Lett.* 5, 5143–5150. doi:10.1109/lra.2020.3005629

Lentini, G., Settimi, A., Caporale, D., Garabini, M., Grioli, G., Pallottino, L., et al. (2019). Alter-ego: A mobile robot with a functionally anthropomorphic upper body designed for physical interaction. *IEEE Robotics Automation Mag.* 26, 94–107. doi:10.1109/mra.2019.2943846

Li, Z., Tsagarakis, N. G., and Caldwell, D. G. (2013). “Stabilizing humanoids on slopes using terrain inclination estimation,” in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE), 4124–4129.

Lippi, V., Mergner, T., Maurer, C., and Seel, T. (2020). “Performance indicators of humanoid posture control and balance inspired by human experiments” in *The international symposium on wearable robotics (WeRob2020) and WearAcon europe*.

Lippi, V., Mergner, T., Seel, T., and Maurer, C. (2019). “Comtest project: A complete modular test stand for human and humanoid posture control and balance” in 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids). IEEE, 630–635.

Liu, C., and Atkeson, C. G. (2009). “Standing balance control using a trajectory library,” in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009 (IEEE), 3031–3036.

Molnar, C. A., Zelei, A., and Insperger, T. (2018). Human balancing on rolling balance board in the frontal plane. *IFAC-PapersOnLine* 51, 300–305. doi:10.1016/j.ifacol.2018.07.240

Monteleone, S., Negrello, F., Grioli, G., Catalano, M. G., Garabini, M., and Bicchi, A. (2020). “Dysturbance: Dynamic and static pusher to benchmark balance,” in I-RIM Conference 2020: 2nd Italian Conference on Robotics and Intelligent Machines.

Negrello, F., Friedl, W., Grioli, G., Garabini, M., Brock, O., Bicchi, A., et al. (2020). Benchmarking hand and grasp resilience to dynamic loads. *IEEE Robotics Automation Lett.* 5, 1780–1787. doi:10.1109/lra.2020.2969180

Negrello, F., Settimi, A., Caporale, D., Lentini, G., Poggiani, M., Kanoulas, D., et al. (2018). Alter-Ego: A mobile robot with a functionally anthropomorphic upper body designed for physical interaction. *IEEE Robotics Automation Mag.* 26, 94–107. doi:10.1109/mra.2019.2943846

Ott, C., Roa, M. A., and Hirzinger, G. (2011). “Posture and balance control for biped robots based on contact force optimization,” in 2011 11th IEEE-RAS

International Conference on Humanoid Robots, Bled, Slovenia, 26–28 October 2011 (IEEE), 26–33.

Parmiggiani, A., Fiorio, L., Scalzo, A., Sureshbabu, A. V., Randazzo, M., Maggiali, M., et al. (2017). “The design and validation of the r1 personal humanoid,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017 (IEEE), 674–680.

Radford, N. A., Strawser, P., Hambuchen, K., Mehling, J. S., Verdeyen, W. K., Donnan, A. S., et al. (2015). Valkyrie: Nasa’s first bipedal humanoid robot. *J. Field Robotics* 32, 397–419. doi:10.1002/rob.21560

Rogers, M. W., and Mille, M.-L. (2018). Balance perturbations. *Handb. Clin. neurology* 159, 85–105. doi:10.1016/b978-0-444-63916-5.00005-7

Semini, C., Tsagarakis, N. G., Guglielmino, E., Focchi, M., Cannella, F., and Caldwell, D. G. (2011). Design of hyq—a hydraulically and electrically actuated quadruped robot. *Proc. Institution Mech. Eng. Part I J. Syst. Control Eng.* 225, 831–849. doi:10.1177/0959651811402275

Stasse, O., Brousse, E., Naveau, M., Régnier, R., Avrin, G., Souères, P., et al. (2018). Benchmarking the hrp-2 humanoid robot during locomotion. *Front. Robotics AI* 5, 122. doi:10.3389/frobt.2018.00122

Stephens, B. (2007). “Integral control of humanoid balance,” in 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA (IEEE), 4020–4027.

Stilman, M., Olson, J., and Gloss, W. (2010). “Golem krang: Dynamically stable humanoid robot for mobile manipulation” in 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 03–07 May 2010 (IEEE), 3304–3309.

Stronge, W. J. (2018). *Impact mechanics*. Cambridge University Press.

Taborri, J., Salvatori, S., Mariani, G., Rossi, S., and Patanè, F. (2020). “Beat: Balance evaluation automated testbed for the standardization of balance assessment in human wearing exoskeleton,” in 2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT, Roma, Italy, 03–05 June 2020 (IEEE), 526–531.

Torricelli, D., Gonzalez-Vargas, J., Veneman, J. F., Mombaur, K., Tsagarakis, N., Del-Ama, A. J., et al. (2015). Benchmarking bipedal locomotion: A unified scheme for humanoids, wearable robots, and humans. *IEEE Robotics Automation Mag.* 22, 103–115. doi:10.1109/mra.2015.2448278

Torricelli, D., and Pons, J. L. (2018). “Eurobench: Preparing robots for the real world,” in *International symposium on wearable robotics* (Springer), 375–378.

Tsagarakis, N. G., Morfe, S., Cerda, G. M., Zhibin, L., and Caldwell, D. G. (2013). “Compliant humanoid coman: Optimal joint stiffness tuning for modal frequency control,” in 2013 IEEE International Conference on Robotics and Automation (IEEE), 673–678.

Vanderborght, B., Albu-Schäffer, A., Bicchi, A., Burdet, E., Caldwell, D. G., Carloni, R., et al. (2013). Variable impedance actuators: A review. *Robotics Aut. Syst.* 61, 1601–1614. doi:10.1016/j.robot.2013.06.009

Vashista, V., Jin, X., and Agrawal, S. K. (2014). “Active tethered pelvic assist device (a-tpad) to study force adaptation in human walking,” in 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China (IEEE), 718–723.

Vashista, V., Reisman, D. S., and Agrawal, S. K. (2013). “Asymmetric adaptation in human walking using the tethered pelvic assist device (tpad),” in 2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR), Seattle, WA, USA, 24–26 June 2013 (IEEE), 1–5.

Vicario, R., Calanca, A., Dima, E., Murr, N., Meneghetti, M., Ferro, R., et al. (2021). “Benchmarking force control algorithms,” in The 14th Pervasive Technologies Related to Assistive Environments Conference, 359–364.

Zambella, G., Monteleone, S., Alarcón, E. P. H., Negrello, F., Lentini, G., Caporale, D., et al. (2020). An integrated dynamic fall protection and recovery system for two-wheeled humanoids. *IEEE Robotics Automation Lett.* 5, 2138–2145. doi:10.1109/lra.2020.2970951

Zemková, E. (2011). Assessment of balance in sport: Science and reality. *Serbian Journal of Sports Sciences*.

Zhang, T., Zhang, W., and Gupta, M. M. (2017). Resilient robots: Concept, review, and future directions. *Robotics* 6, 22. doi:10.3390/robotics6040022

Zhang, W.-J., and Lin, Y. (2010). On the principle of design of resilient systems—application to enterprise information systems. *Enterp. Inf. Syst.* 4, 99–110. doi:10.1080/17517571003763380



OPEN ACCESS

EDITED BY

Holger Voos,
University of Luxembourg, Luxembourg

REVIEWED BY

Maral Partovibakhsh,
Islamic Azad University System, Iran
Claudio Cimarelli,
University of Luxembourg, Luxembourg

*CORRESPONDENCE

Allahyar Montazeri,
✉ a.montazeri@lancaster.ac.uk
Nargess Sadeghzadeh-Nokhodberiz,
✉ sadeghzadeh@qut.ac.ir

RECEIVED 05 November 2022

ACCEPTED 03 May 2023

PUBLISHED 30 May 2023

CITATION

Sadeghzadeh-Nokhodberiz N, Iranshahi M and Montazeri A (2023), Vision-based particle filtering for quad-copter attitude estimation using multirate delayed measurements.
Front. Robot. AI 10:1090174.
doi: 10.3389/frobt.2023.1090174

COPYRIGHT

© 2023 Sadeghzadeh-Nokhodberiz, Iranshahi and Montazeri. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). 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.

Vision-based particle filtering for quad-copter attitude estimation using multirate delayed measurements

Nargess Sadeghzadeh-Nokhodberiz^{1*}, Mohammad Iranshahi¹ and Allahyar Montazeri^{2*}

¹Electrical and Computer Engineering Department, Qom University of Technology, Qom, Iran,

²Engineering Department, Lancaster University, Lancaster, United Kingdom

In this paper, the problem of attitude estimation of a quad-copter system equipped with a multi-rate camera and gyroscope sensors is addressed through extension of a sampling importance re-sampling (SIR) particle filter (PF). Attitude measurement sensors, such as cameras, usually suffer from a slow sampling rate and processing time delay compared to inertial sensors, such as gyroscopes. A discretized attitude kinematics in Euler angles is employed where the gyroscope noisy measurements are considered the model input, leading to a stochastic uncertain system model. Then, a multi-rate delayed PF is proposed so that when no camera measurement is available, the sampling part is performed only. In this case, the delayed camera measurements are used for weight computation and re-sampling. Finally, the efficiency of the proposed method is demonstrated through both numerical simulation and experimental work on the DJI Tello quad-copter system. The images captured by the camera are processed using the ORB feature extraction method and the homography method in Python-OpenCV, which is used to calculate the rotation matrix from the Tello's image frames.

KEYWORDS

UAV, quad-copter, particle filtering, multi-rate sensor fusion, attitude estimation, camera, gyroscope (gyro)

1 Introduction

Autonomous quad-copter UAVs are increasingly employed in various industries, especially in applications with extreme environments where humans cannot access narrow, high altitude, far reaching, and confined spaces for further operation and inspection (Montazeri et al., 2021). Of particular importance is the ability of quad-copters to accurately maneuver in hazardous and unstructured environments such as those existing in the nuclear decommissioning applications. One of the challenging tasks for navigation of drones in such GPS-denied environments is finding the exact position and orientation of the quad-copters for feedback control and characterization of the environment (Burrell et al., 2018). Nowadays, the inertial navigation system (INS) including inertial measurement units (IMU) is widely used for navigation of UAVs. Toward this, first of all, a robust and reliable attitude estimator is required which should be able to execute on low-cost computational hardware and using measurements from light-weight sensors (Bassolillo et al., 2022).

Attitude estimation is the procedure of estimating orientation of the vehicle with respect to a reference frame using sensory measurements such as inertial and attitude

sensors. Although least square error (LSE) and maximum likelihood (ML) approaches can be classified as early attitude estimation methods, model-based Bayesian approaches are most common and precise approaches can be found in [Dhahbane et al. \(2021\)](#). Model-based approaches normally employ vehicle kinematics and/or dynamics to provide a prediction from the orientation, and the predicted attitude is updated through the sensory measurements ([Sadeghzadeh-Nokhodberiz and Poshtan, 2016](#); [Ozaki and Kuroda, 2021](#)). There are an increasing number of research studies devoted to attitude estimation ([Moutinho et al., 2015](#); [Nokhodberiz et al., 2019](#); [Liang, 2017](#)). The commonly used stochastic approaches are the Kalman filter (KF) and extended Kalman filter (EKF) ([Nemati and Montazeri, 2019](#)). However, in KF-based methods (KF, EKF, and UKF), only Gaussian noise processes are considered and EKF suffers from the linearization issue. Therefore, in some research studies, particle filters (PF) are used to overcome the problem in attitude estimation of UAVs ([Cheng and Crassidis, 2004](#); [Sadeghzadeh-Nokhodberiz et al., 2014b](#)). The gyroscope measurements in the body frame are normally incorporated in the attitude kinematics to obtain the orientation in the inertial frame. The gyroscope noises can be modeled through a probability distribution function, making the kinematics a stochastic model as it is employed as an input vector in it. Therefore, it is necessary to employ a stochastic approach such as PF that works directly with the non-linear dynamic model of the system. PFs are appropriate for attitude estimation of quad-copters due to non-linear and non-stochastic nature of the system model. PF is an optimal non-linear filtering method in which the posterior probability density function (pdf) is approximated through sample point (particles) generation as it is not possible to be computed analytically for non-Gaussian systems. This posterior pdf is required for Bayesian minimum mean square error (MMSE) estimation, and it is the main advantage of PF over other non-linear Bayesian MMSE estimators such as EKF ([Sadeghzadeh-Nokhodberiz and Meskin, 2020](#)).

Additional sensors such as cameras are commonly employed together with the low-cost inertial sensors. This greatly helps mitigate the effect of errors and noises in the gyroscope measurement and facilitates designing a vision-based navigation technique ([Sadeghzadeh-Nokhodberiz et al., 2014a](#); [Sadeghzadeh-Nokhodberiz et al., 2014c](#)). Although cameras can provide highly accurate measurements from the quad-copter orientation compared to low-cost gyroscopes, they suffer from a slow sampling rate and delay problems with respect to the gyroscope measurements due to heavy computation load required. In the vision-based navigation, feature points extracted from the camera images are tracked and the camera motion, mounted on the UAV, is related to the locations of tracked planar feature points in the image plane using the homography relationship ([Wang et al., 2013](#); [White and Beard, 2019](#)). Homography-based state estimation of a quad-copter system using EKF is presented in [Chavez et al. \(2017\)](#). The images captured by cameras should be highly processed for feature extraction including detection, description, and matching ([Csurka et al., 2018](#)). Although a recently developed ORB ([Rublee et al., 2011](#)) method can significantly reduce the processing time compared to the popular SIFT ([Lowe, 2004](#)) and SURF ([Bay et al., 2006](#)) approaches, it still needs almost 33 ms for feature extraction per image ([Mur-Artal et al., 2015](#)). This processing time not only leads

to a much slower sampling rate but also the measured values are received with a significant delay for the attitude estimation procedure.

The problem of multi-rate delayed state estimation has been studied in [Lin and Sun \(2021\)](#), [Comellini et al. \(2020\)](#), [Fatehi and Huang \(2017\)](#), and [Khosravian et al. \(2015\)](#). [Lin and Sun \(2021\)](#) and [Khosravian et al. \(2015\)](#) proposed a cascaded output predictor and an attitude observer where the effect of sampling and delays are compensated in the predictor. The delayed measurements are extrapolated to present time using past and present estimates of the KF in [Larsen et al. \(1998\)](#), where an optimal gain is derived for this extrapolated measurement. In [Lin and Sun \(2021\)](#), the system with delayed and multi-rate measurements is transformed into a delay-free and single-rate system using a state iterating method, and a non-augmented recursive optimal linear state filter is presented for the system by utilizing projection theory. In [Fatehi and Huang \(2017\)](#), different KFs are employed for each type of measurement and the estimates are fused considering the correlation between them in the next step. The cross-covariance matrix between the estimation errors of KFs is obtained iteratively to be employed in the fusion process. Due to the non-linear attitude kinematics with respect to the Euler angles and its stochastic nature due to the incorporation of the gyroscope noise in the model, the PF is an appropriate choice for the attitude estimation. It is worth mentioning that as long as staying away from singularity points ($\pm 90^\circ$ rotations of pitch angle), the Euler angle representation of the attitude is preferred to the quaternion representation as the quaternion must obey its normalization constraint, which can cause issues in the filtering ([Markley and Crassidis, 2014](#)). In [Bassolillo et al. \(2022\)](#), a KF-based sensor fusion algorithm, using a low-cost navigation platform that contains an inertial measurement unit (IMU), five ultrasonic ranging sensors, and an optical flow camera is proposed to improve navigation of a UAV system in indoor GPS-denied environments. A multi-rate version of the EKF is employed to deal with the use of heterogeneous sensors with different sampling rates and the presence of non-linearity in the model.

To the best of the authors' knowledge, the problem of PF-based attitude estimation using PF with multi-rate delayed sensors has not yet been studied in the literature. Accordingly, in this paper, a multi-rate delayed PF is proposed to estimate the orientation with a discretized attitude kinematics in Euler angles. It is shown that the corresponding weights of the generated particles are the likelihood of generally non-Gaussian delayed camera measurements. The result is then validated through simulation and experiments on a UAV quad-copter system. For the experimental work, a DJI Tello quad-copter system is employed where the images are processed using the ORB feature extraction method and Python-OpenCV is employed to calculate the rotation matrix using the homography approach.

The organization of the paper proceeds as follows. The system and measurement models including the attitude kinematic model, and gyroscope and camera measurement models are presented in [Section 2](#). [Section 3](#) provides with the PF with multi-rate delayed measurements. Simulation results are presented in [Section 4](#) to demonstrate the accuracy of the presented PF. The experimental data gathered from DJI Tello quad-copter systems are analyzed in [Section 5](#). Finally, conclusion is provided in [Section 6](#).

2 System and measurement models

2.1 System model

The quad-copter attitude kinematics which represents the relationship between angular velocities in the body and inertial frames are described as follows [Sadeghzadeh-Nokhodberiz et al. \(2021\)](#):

$$\begin{aligned}\dot{\phi}(t) &= p(t) + \sin(\phi(t)) \tan(\theta(t)) q(t) + \cos(\phi(t)) \tan(\theta(t)) r(t), \\ \dot{\theta}(t) &= \cos(\theta(t)) q(t) - \sin(\phi(t)) r(t), \\ \dot{\psi}(t) &= \frac{\sin(\phi(t))}{\cos(\theta(t))} q(t) + \frac{\cos(\phi(t))}{\cos(\theta(t))} r(t),\end{aligned}\quad (1)$$

where $x(t) = [\phi(t)\theta(t)\psi(t)]^T$ is the attitude vector of quad-copter which is defined in the inertial frame where roll angle $\phi(t)$, pitch angle $\theta(t)$, and yaw angle $\psi(t)$ determine rotations around x -axis, y -axis, and z -axis, respectively. In addition, $p(t)$, $q(t)$, and $r(t)$ are angular velocities rotating around x -axis, y -axis, and z -axis in the body frame, respectively, and t refers to time.

2.2 Measurement models

2.2.1 The gyroscope measurement model

The gyroscope measurement model can be written as follows:

$$\omega_m(k) = \omega_b(k) + v_\omega(k), \quad (2)$$

where $\omega_m(k) \in \mathbb{R}^3$ and $\omega_b(k) \in \mathbb{R}^3$ are the vectors of measured and true angular velocities in the body frame at sample time k and $\omega_b = [p \ q \ r]^T$ (see Eq. 1). Moreover, $v_\omega \in \mathbb{R}^3$ is a zero-mean generally

non-Gaussian measurement noise with a known probability density function (pdf) with the covariance matrix of R_ω .

2.2.2 The camera measurement model

Homography is used to obtain the measurement model of camera by providing the transformation (including scale, rotation, and translation) between two images. Toward this, two consecutive frames from a camera mounted on a moving body viewing a fixed point P are considered. The fixed point is considered a feature extracted from the images using some feature extraction approaches such as ORB ([Rublee et al., 2011](#)).

singular value decomposition (SVD) is then performed with the feature pairs that pass the RANSAC test to calculate the homography matrix H . Let m_1 and m_2 be the two projections of point P in the camera coordinates with R_{12} and t_{12} as the corresponding rotation matrix and translation vector, respectively, in the camera frame transforming m_1 to m_2 (see [Figure 1](#)). In [Figure 1](#), d is the Euclidean distance between the plane π , with the unit normal vector n , and position 1. The relationships between the homography matrix and the transformation between two images can be found in [Wang et al. \(2013\)](#). Finally, through the SVD of the homography matrix, R_{12} and t_{12} can be obtained which can be transformed to the direction cosine matrix (DCM), R , iteratively. The Euler angles can be then computed using the DCM.

Therefore, the camera measurement model without delay consideration can be represented as

$$y_e(k) = x(k) + v_e(k), \quad (3)$$

where v_e is a zero-mean generally non-Gaussian measurement noise with the covariance matrix of R_e . It is worth mentioning that the index e refers to the Euler angles.

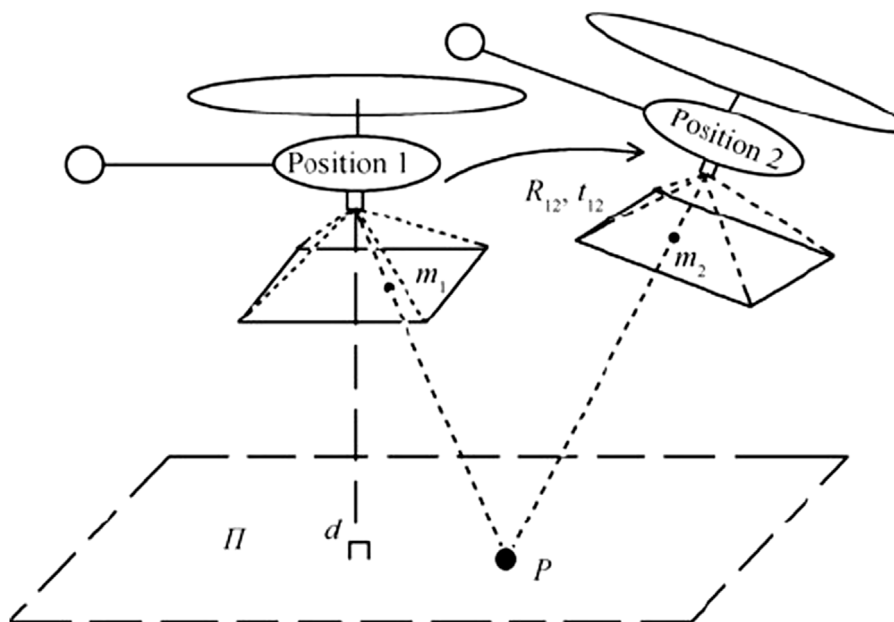


FIGURE 1

Same fixed point, P , viewed from two different positions of a moving quad-copter for the homography purpose [Wang et al. \(2013\)](#).

3 The proposed multirate particle filter

In this section, a PF for the system and measurement models introduced in the previous section is presented when the sensors collect data using a multi-rate sampling frequency procedure.

3.1 The general model

Toward this, let the gyroscope and camera sampling times in seconds be represented with T and sT , respectively, with $s \in \mathbb{N}$. Moreover, we consider the camera processing delay time as dT with $d \in \mathbb{N}$ and $d < s$, as depicted in **Figure 2**.

Moreover, a discretized form of the kinematic model presented in Eq. 1 is also considered with the general non-linear discrete state space model as follows:

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{\bar{w}}_{k-1}, \quad (4)$$

where $\mathbf{u}_k = [p(k)q(k)r(k)]^T$ and \mathbf{x}_k is defined in Equation 1 and $\mathbf{\bar{w}}_{k-1}$ is the additive process noise resulting from the gyroscope measurement noise $\mathbf{v}_\omega(k)$ with the same distribution but a scaled covariance matrix \mathbf{Q}_x . The index k refers to the sampling instant.

In other words, the components of the discretized model of (4) are as follows:

$$\begin{aligned} \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) &= \mathbf{x}_{k-1} + T\mathbf{A}_{k-1}\mathbf{u}_k, \\ \mathbf{\bar{w}}_{k-1} &= T\mathbf{A}_{k-1}\mathbf{v}_\omega(k), \end{aligned} \quad (5)$$

$$\text{where } \mathbf{A}_{k-1} = \begin{bmatrix} 1 & \sin(\phi_{k-1})\tan(\theta_{k-1}) & \sin(\phi_{k-1})\tan(\theta_{k-1}) \\ 0 & \cos(\theta_{k-1}) & \cos(\theta_{k-1}) \\ 0 & \frac{\sin(\phi_{k-1})}{\cos(\theta_{k-1})} & \frac{\cos(\phi_{k-1})}{\cos(\theta_{k-1})} \end{bmatrix}.$$

3.2 The modified particle filter

The general approach in the PF is to compute the posterior pdf using the Monte Carlo (MC) method used in the Bayesian estimation of the stochastic process \mathbf{x}_k by having the measurement history $\mathbf{y}_{1:k} = \{\mathbf{y}_1, \dots, \mathbf{y}_k\}$ and the current sample of input at time k . The goal in standard PF is to approximate the posterior pdf $p(\mathbf{x}_k | \mathbf{y}_{1:k})$ by generating particles from a known distribution and estimating the target pdf through attribution of the normalized weights for each particle.

However, in case of this study as explained earlier, the measurements are delayed and a slower sampling rate is considered

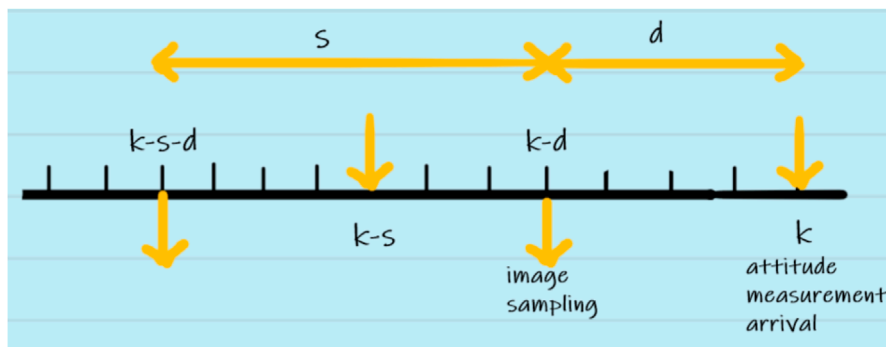
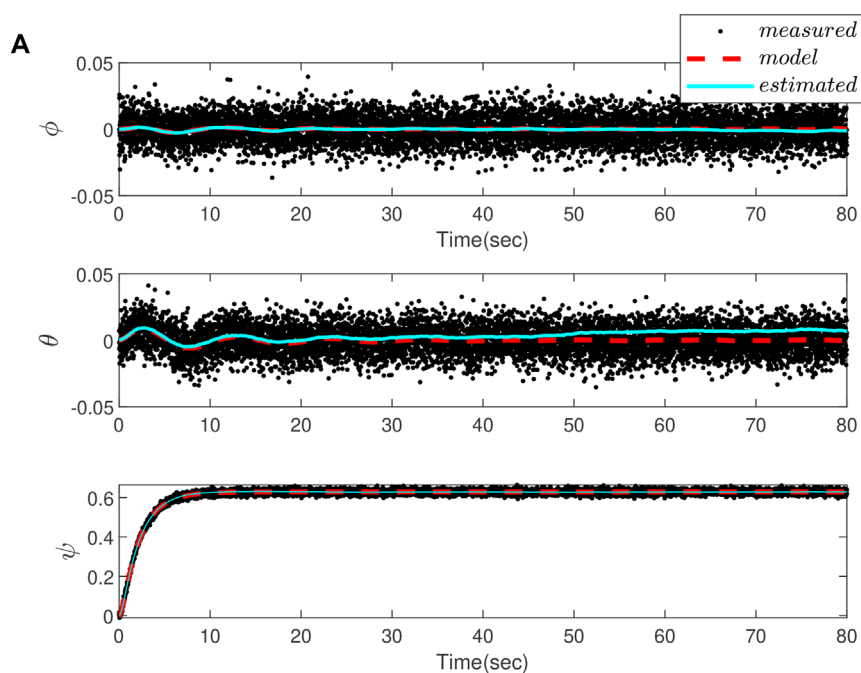


FIGURE 2

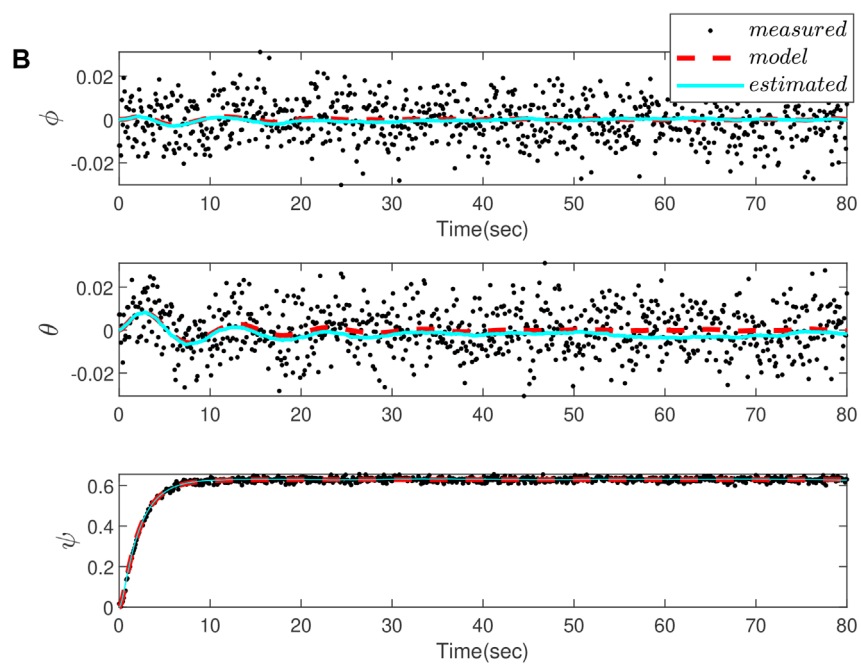
Camera sampling and delay scenario.

TABLE 1 Pseudo code corresponding to the proposed PF for attitude estimation.

Step 0: Initialization: Sample initial particles, that is, $\{\mathbf{x}_0^i\}_{i=1}^N$, using a known initial distributions of states ($p(\mathbf{x}_0)$), where $\mathbf{x}_0 \sim p(\mathbf{x}_0)$
At the time instant k
Step 1: Prior estimate: Generate the prior state particles $\{\mathbf{x}_k^{i-}\}_{i=1}^N$ using the system model, that is, $\mathbf{x}_k^{i-} \sim p(\mathbf{x}_k \mathbf{x}_{k-1}^{i+})$, for $i = 1:N$
In other words for the attitude system of (4) and 5: $\mathbf{x}_k^{i-} = \mathbf{x}_{k-1}^{i+} + T\mathbf{A}_{k-1}^i \mathbf{\bar{w}}_{k-1}^i - \mathbf{\bar{w}}_{k-1}^i$, where \mathbf{x}_k^{i-} is the prior estimate at the time sample k and $\mathbf{x}_{k-1}^{i+} = \mathbf{A}_{k-1}^i \mathbf{x}_{k-1}^{i-}$ and $\mathbf{\bar{w}}_{k-1}^i$ is the particle generated using a known pdf of $\mathbf{\bar{w}}_{k-1}$ which is zero mean generally non-Gaussian noise with the covariance of $\mathbf{Q}_x = T^2 \mathbf{A}_{k-1}^i \mathbf{R}_\omega (\mathbf{A}_{k-1}^i)^T$
Step 2: Posterior estimate: Compute $\{\mathbf{x}_k^{i+}\}_{i=1}^N$ as follows.
IF the camera measurement, $\mathbf{y}_{e,k-d}$, is available, the weights are computed using (10) and the particles are re-sampled to generate posterior estimates with equal weights of $\frac{1}{N}$, that is, $\{\mathbf{x}_k^{i+}, \frac{1}{N}\}_{i=1}^N$
ELSE let $\mathbf{x}_k^{i+} = \mathbf{x}_k^{i-}$, $i = 1, \dots, N$ with the corresponding weights of $\frac{1}{N}$ for each sample
END
Step 3: State estimation: Estimate the system states as $\hat{\mathbf{x}}_k = \sum_{i=1}^N \frac{1}{N} \mathbf{x}_k^{i+}$

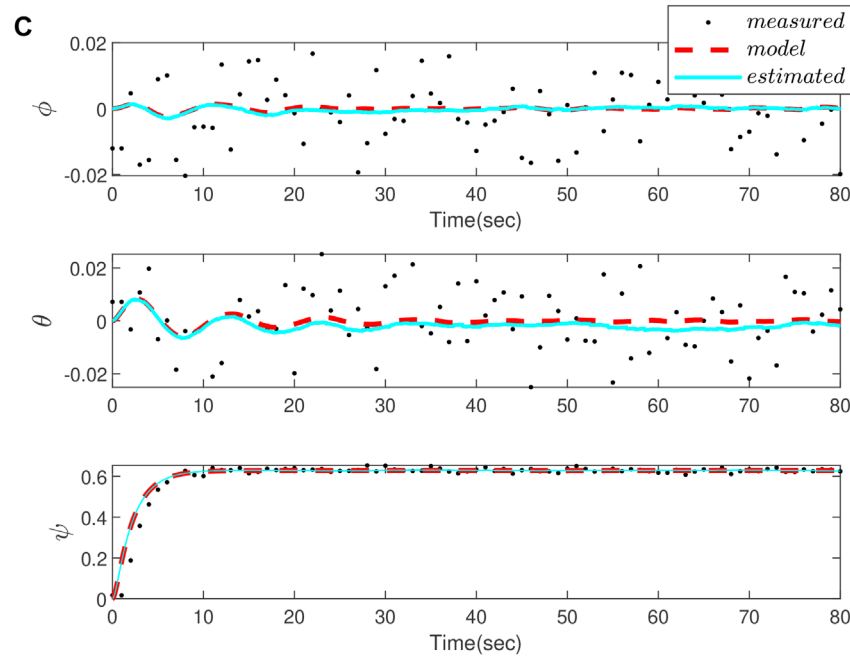


$s = 1$ and $d = 0$.



$s = 10$ and $d = 5$.

FIGURE 3
(Conitnued).



$$s = 100 \text{ and } d = 50.$$

FIGURE 3

(Continued). Comparing attitude estimation of a drone for a different delay d and sampling time s , using measurement by a camera (dots), estimation by the proposed particle filter (blue dot-line), and the kinematic model (red dot-line). (A) No delay and sampling rate for both gyro and camera (B) Sampling rate of the camera is 10 times slower than the gyro and the delay in the camera measurements is 5 samples (C) Sampling rate of the camera is 100 times slower than the gyro and the delay in the camera measurements is 50 samples.

for the camera compared with the data measured from the gyroscope. Therefore, two different cases may happen at each sampling step. The delayed camera measurements are arrived or there are no camera measurements. In this section, the PF for these two cases are derived.

3.2.1 Case I: camera measurement available

In this case, as depicted in Figure 2, the delayed camera measurement is available at sample time k . Therefore, the posterior distribution $p(\mathbf{x}_{1:k}|\mathbf{y}_{e,1:s;k-d})$ should be approximated using the MC method such that $\mathbf{y}_{e,1:s;k-d}$ refers to the historical data collected by camera every s sampling instant and with the initial time delay of d . Here, the proposal distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is employed for the particle generation. By applying the Bayes' rule and the Markov property, it can be concluded that

$$p(\mathbf{x}_{1:k}|\mathbf{y}_{e,1:s;k-d}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}) \times \dots \times p(\mathbf{x}_{k-d+1}|\mathbf{x}_{k-d}) \times p(\mathbf{x}_{1:k-d}|\mathbf{y}_{e,1:s;k-d}). \quad (6)$$

Here, $p(\mathbf{x}_{1:k-d}|\mathbf{y}_{e,1:s;k-d})$ using the Bayes' rule and statistical independencies can be rewritten as follows:

$$p(\mathbf{x}_{1:k-d}|\mathbf{y}_{e,1:s;k-d}) \propto p(\mathbf{y}_{e,k-d}|\mathbf{x}_{k-d}) \times p(\mathbf{x}_{1:k-d}|\mathbf{y}_{e,1:s;k-d-s}). \quad (7)$$

The term $p(\mathbf{x}_{1:k-d}|\mathbf{y}_{e,1:s;k-d-s})$ is also extended as follows:

$$p(\mathbf{x}_{1:k-d}|\mathbf{y}_{e,1:s;k-d-s}) = p(\mathbf{x}_{k-d}|\mathbf{x}_{k-d-1}) \times \dots \times p(\mathbf{x}_{k-s+1}|\mathbf{x}_{k-s}) \times p(\mathbf{x}_{1:k-s}|\mathbf{y}_{e,1:s;k-d-s}). \quad (8)$$

Using Eqs 6–8, the weight function is computed as follows:

$$w(\mathbf{x}_k) = p(\mathbf{y}_{e,k-d}|\mathbf{x}_{k-d}) w(\mathbf{x}_{k-s}). \quad (9)$$

The weight functions are evaluated for the particles generated using the proposal distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ and using the system probabilistic model represented by the kinematics model in Eq. 1. So the particles $\mathbf{x}_k^{i=1:N}$ are generated where N refers to the number of particles. Since re-sampling should be carried out as the next step, the weights of the particles at sample time $k-s$ are transformed to $\frac{1}{N}$ after re-sampling with re-sampled particles of $\mathbf{x}_{k-s}^{i=1:N}$.

Therefore, the normalized weights for re-sampling for each particle are computed as follows:

$$w_k^{i*} = \frac{p(\mathbf{y}_{e,k-d}|\mathbf{x}_{k-d}^{i*})}{\sum_{j=1}^N p(\mathbf{y}_{e,k-d}|\mathbf{x}_{k-d}^{j*})}. \quad (10)$$

3.2.2 No camera measurement available

In this case, no camera data are available at sample time k . Therefore, $p(\mathbf{x}_{1:k}|\mathbf{y}_{e,1:s;k-d-s})$ should be approximated. Therefore, the weight function is $w(\mathbf{x}_k) = w(\mathbf{x}_{k-s})$ as it is proved in the following. Since the last received measurement from the camera at the sampling instant k is \mathbf{y}_{k-d-s} and received at $k-s$ sampling instant, therefore

$$p(\mathbf{x}_{1:k}|\mathbf{y}_{e,1:s;k-d-s}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}) \times \dots \times p(\mathbf{x}_{k-d+1}|\mathbf{x}_{k-d}) \times p(\mathbf{x}_{1:k-d}|\mathbf{y}_{e,1:s;k-d-s}), \quad (11)$$

TABLE 2 RMSE for different scenarios.

Scenario	$s = 1, d = 0$	$s = 10, d = 5$	$s = 100, d = 50$
RMSE	9.7001×10^{-6}	3.2284×10^{-5}	8.2284×10^{-5}

where the term $p(\mathbf{x}_{1:k-d} | \mathbf{y}_{e,1:s,k-d-s})$ is computed in Eq. 8 which states that the corresponding weight of the i^{th} particle \mathbf{x}_{k-1}^{i-} is $w(\mathbf{x}_{k-1}^{i-})$ and after normalization and re-sampling the corresponding weight of each particle would become $\frac{1}{N}$.

In order to clarify the proposed method, it is presented in Pseudo code in Table 1.

4 Simulation results

In this section, simulation results are provided in the MATLAB/SIMULINK environment to show the efficiency of the proposed method. The simulated AR Drone Parrot 2.0 quad-copter is stabilized using a non-linear robust sliding mode control technique presented in Nemati and Montazeri (2018a) and Nemati and Montazeri (2018b). The physical parameters of the quad-copter are listed as follows:

$$I_{xx} = 7.72 \times 10^{-2} \text{ kg } m^2, \quad I_{yy} = 7.64 \times 10^{-2} \text{ kg } m^2,$$

$$I_{zz} = 0.1031 \text{ kg } m^2, \quad I_r = 1.8 \times 10^{-5} \text{ kg } m^2, \quad m = 2.5 \text{ kg}.$$

In order to evaluate the performance of the proposed PF, different scenarios for delay and sampling rate values are considered. Figure 3A depicts the attitude estimation result using the gyro and camera measurements, and the results are compared with those obtained from the kinematics model when camera has no delay and the sampling rates are equal, that is, $s = 1$ and $d = 1$. Figures 3B,C are related to the cases with $s = 10, d = 5$ and $s = 100, d = 50$, respectively. It is obvious from the figures that although the camera slows the sampling rate and processing delay, and deteriorates the estimation accuracy, the proposed method is still successful to provide an accurate estimation. The root mean square error (RMSE) criterion is also employed to provide a numerical measure for a comparative study of the results. The results are summarized in Table 2, which also confirms the aforementioned discussion.

5 Experimental results

The experimental results are provided using the DJI Tello drone illustrated in Figure 4. It is a small ($99 \text{ mm} \times 92.5 \text{ mm} \times 41 \text{ mm}$) and lightweight (80g) drone with a maximum speed of 8 m/s which uses the 2.4 GHz Wi-Fi communication channel to be connected to a PC or laptop for sending and receiving telemetry data and commands, respectively. The drone is equipped with an IMU and a 720p camera and an SDK is provided to help developers for implementation of their algorithms. Although the camera information is available, the SDK cannot read the IMU data and instead the results of internal positioning data can be read Steenbeek (2020).

To perform the homography, first, the ORB feature matching method is applied on the frames captured by the DJI Tello camera. Toward this, the ORB algorithm uses the improved FAST



FIGURE 4
Image of the DJI Tello mini-drone.

algorithm, used in image feature point detection, the feature point screening, image pyramid building, and the feature point direction determination. After that, the ORB algorithm uses the improved BRIEF algorithm to generate binary feature point descriptors, and then, the descriptors are corrected using the steer BRIEF method to include the direction information. Finally, in the process of feature point matching, the points are matched based on their descriptor similarities. Toward this, the Brute-Force matcher method applied in Hamming distance is used to measure the distance between the binary descriptors and to choose the nearest ones as the matched points. Finally, by employing the PROSAC algorithm, the matched points with larger matching errors are rejected, which significantly improves the accuracy of matching Luo et al. (2019).

For this purpose, in Python-OpenCV, we have employed the following command for the feature point detection and generation of descriptors:

```
[keypoints, descriptors] = orb.detectAndCompute(img)
```

The result of this command for a sample frame is depicted in Figure 5A. In addition, for feature matching, the following command is employed:

```
matches = bf.match(des1, des2)
```

where *des1* and *des2* are descriptor vectors of two successive frames. Finally, the matched points are sorted to find the best matches. The result of feature matching is depicted in Figure 5B for the same sample frames.

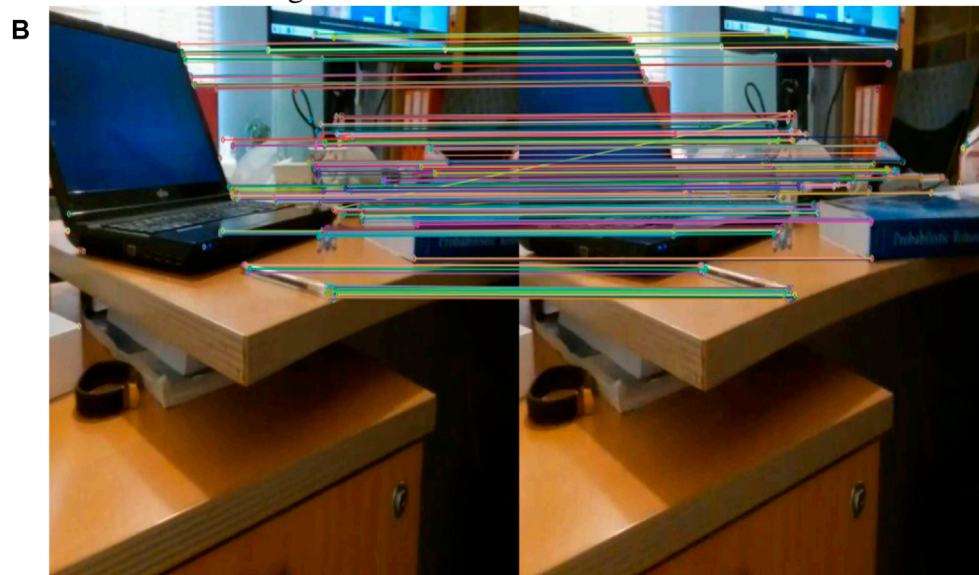
To compute the homography matrix, the following command is employed:

```
H = cv.findHomography(srcPoints, dstPoints),
```

where *srcPoints* are coordinates of the points in the original plane, which is a cell array of 2-element vectors $\{[x, y], \dots\}$ with single floating-point precision and *dstPoints* are coordinates of the points in the target plane, of the same size and type as *srcPoints*. Now, in



An example of feature detection using ORB algorithm.



An example of feature matching using ORB algorithm.

FIGURE 5

ORB feature extraction results achieved by the drone camera in a hover position. (A) Feature extraction (B) Feature matching.

order to compute the DCM from the computed homography matrix, the following command is employed:

$$[n, Rs, Ts, Ns] = cv2.decomposeHomographyMat(H, K).$$

Here, H is the input homography matrix between two images, K is the input camera intrinsic matrix, Rs are array of rotation matrices, Ts are array of translation matrices, and Ns are array of plane normal matrices. In addition, n is the number of possible solutions and returned as the function output. The set of four solutions is returned using this command which can be reduced to two or one using the method explained in [Malis and Vargas \(2007\)](#). Reducing the number of solutions to two can be achieved by using additional constraints. For this purpose, a set of reference image points p^* is selected and by

using the camera intrinsic matrix K , the points are projected using the relation as follows:

$$m^* = K^{-1} p^*.$$

In this case, the valid solutions are those satisfying the projection inequality as follows for all points in the plane determined by the normal vector m^* and n^* is the normal vector of the corresponding plane

$$m^{*T} n^* > 0.$$

The frame rate of the camera is 30 frames per sec, and we have processed the frames on an 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.8 GHz processor. The results of the attitude estimation

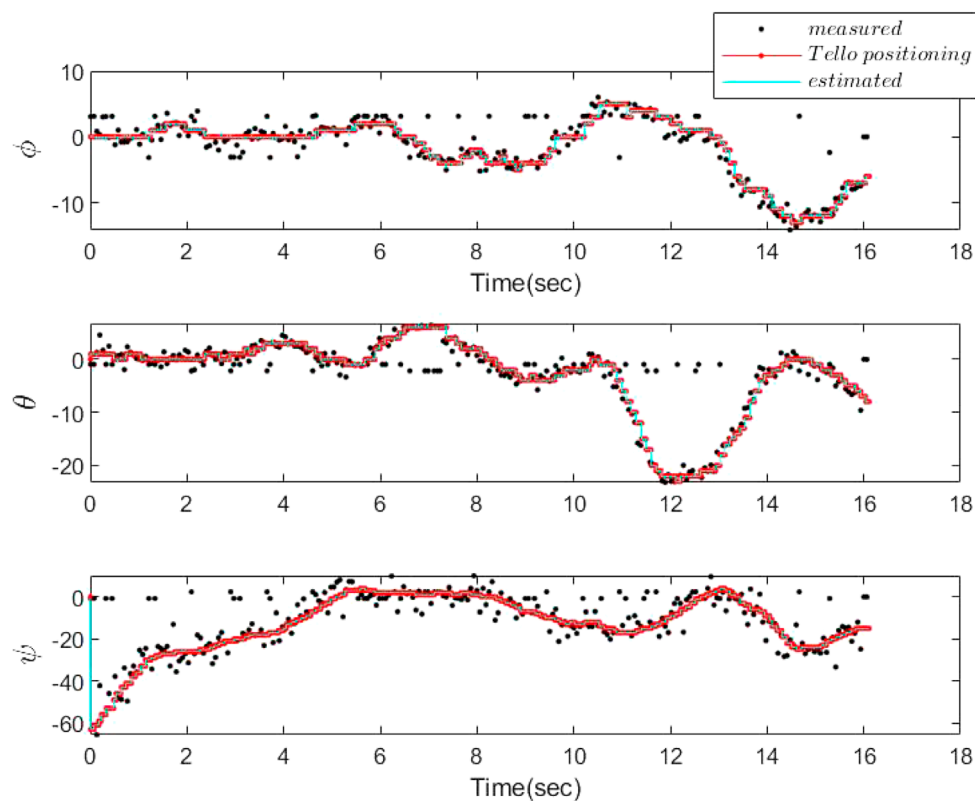


FIGURE 6

Attitude estimation versus the experimental data from the DJI Tello with varying d and s .

using the proposed PF compared to the camera homography (measurement results) and the positioning results, recorded by DJI Tello, are depicted in Figure 6. It is worth mentioning that due to experimental limitations the sampling time and delay in the recorded experimental data have not been constant, but they are known. However, the delay caused by the camera processing time has always been less than these varying sampling periods. Therefore, the proposed approach is applicable to the problem at hand. In other words, the camera frames are sampled after the results of the ORB and homography procedures of the last sampled frames are available. It took on average 0.03 s for processing of each frame for our processor and 0.02 s for the ORB feature extraction and matching, and the homography procedure only needs 4×10^{-4} s. The sampling rate of the positioning procedure on average is 17×10^{-4} s. Therefore, on average $d = 30$ and $s = 40$. In other words, the sampling rate of the camera is 0.068 s, and thus, only one frame is missed due to the frame rate of 30 frames per sec. Accordingly, the processing delay of the camera information is less than the sampling time of the camera. Although we have processed the gathered data from the camera offline, it is also possible to be processed online. In other words, sampling two successive frames is fast enough such that still there exists features to be matched between two sample frames (only one frame is missed). Moreover, in practical implementation of the particle filter estimation algorithm, since the IMU data are not directly available for measurement, the gathered positioning data by camera are used for the particle generation.

This is replaced by the kinematic model used in the simulation results.

6 Conclusion

An extension of the sampling importance re-sampling (SIR) particle filter (PF) was proposed in this paper to solve the problem of attitude estimation of a quad-copter system equipped with a multi-rate camera and gyroscope sensors. In the proposed PF, the delayed camera measurements are used for weight computation and re-sampling and when no camera measurement is available, only the sampling is performed. It was shown through simulation and experimental data that the method is successful to estimate the attitude truly in the presence of delayed multi-rate camera measurements. In the experimental part, the ORB feature matching method was employed for image processing in Python-OpenCv, and after that, the DCM was computed using homography. As our future research topic, we intend to solve the problem of attitude estimation using particle filtering, in the presence of gyroscope faults and errors such as sensory biases and drifts, as well as delayed multi-rate camera measurements.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Author contributions

All authors contributed to the paper at different preparation stages. MI helped in conducting the experiments and preparation of results. NS-N contributed in writing the main manuscript, supervising the students, and generating ideas. AM contributed by developing the research ideas, editing and reviewing the manuscript, supervising the project, and providing fund for the research. All authors contributed to the article and approved the submitted version.

Funding

Herewith, all funding resources to support this research are declared. The fund helps support open-access publication through the agreement between Lancaster University and Frontiers.

References

- Bassolillo, S. R., D'Amato, E., Notaro, I., Ariante, G., Del Core, G., and Mattei, M. (2022). Enhanced attitude and altitude estimation for indoor autonomous uavs. *Drones* 6, 18. doi:10.3390/drones6010018
- Bay, H., Tuytelaars, T., and Van Gool, L. (2006). "Surf: Speeded up robust features," in *European conference on computer vision* (Berlin, Germany: Springer), 404–417.
- Burrell, T., West, C., Monk, S. D., Montazeri, A., and Taylor, C. J. (2018). "Towards a cooperative robotic system for autonomous pipe cutting in nuclear decommissioning," in *Proceedings of the 2018 UKACC 12th International Conference on Control (CONTROL)*, Sheffield, UK, 05-07 September 2018, 283–288. doi:10.1109/CONTROL.2018.8516841
- Chavez, A., L'Heureux, D., Prabhakar, N., Clark, M., Law, W.-L., and Prazenica, R. J. (2017). "Homography-based state estimation for autonomous uav landing," in *Proceedings of the AIAA Information Systems-AIAA Infotech@ Aerospace*, Kissimmee, Florida, 8–12 January 2018.
- Cheng, Y., and Crassidis, J. (2004). "Particle filtering for sequential spacecraft attitude estimation," in *Aiaa guidance, navigation, and control conference and exhibit* (New York: University at Buffalo), 5337.
- Comellini, A., Casu, D., Zenou, E., Dubanchet, V., and Espinosa, C. (2020). Incorporating delayed and multirate measurements in navigation filter for autonomous space rendezvous. *J. Guid. Control, Dyn.* 43, 1164–1172. doi:10.2514/1.g005034
- Csurka, G., Dance, C. R., and Humenberger, M. (2018). From handcrafted to deep local features. *arXiv preprint arXiv:1807.10254*
- Dhahbane, D., Nemra, A., and Sakhi, S. (2021). "Attitude determination and attitude estimation in aircraft and spacecraft navigation a survey," in *In 2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH) (IEEE)*, Boumerdes, Algeria, 09-10 February 2021.
- Fatehi, A., and Huang, B. (2017). Kalman filtering approach to multi-rate information fusion in the presence of irregular sampling rate and variable measurement delay. *J. Process Control* 53, 15–25. doi:10.1016/j.jprocont.2017.02.010
- Khosravi, A., Trumpf, J., Mahony, R., and Hamel, T. (2015). "Recursive attitude estimation in the presence of multi-rate and multi-delay vector measurements," in *Proceedings of the 2015 American Control Conference*, Chicago, IL, July 1–3.
- Larsen, T. D., Andersen, N. A., Ravn, O., and Poulsen, N. K. (1998). "Incorporation of time delayed measurements in a discrete-time kalman filter," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, Tampa, FL, USA, 18-18 December 1998, 3972–3977.4
- Liang, W. (2017). *Attitude estimation of quadcopter through extended kalman filter*. Ph.D. thesis, Lehigh University, Bethlehem, PA
- Lin, H., and Sun, S. (2021). Estimator for multirate sampling systems with multiple random measurement time delays. *IEEE Trans. Automatic Control* 67, 1589–1596. doi:10.1109/tac.2021.3070299
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60, 91–110. doi:10.1023/b:visi.0000029664.99615.94
- Luo, C., Yang, W., Huang, P., and Zhou, J. (2019). "Overview of image matching based on orb algorithm," in *Journal of physics: Conference series* (Bristol, United Kingdom: IOP Publishing), 1237, 032020.
- Malis, E., and Vargas, M. (2007). Deeper understanding of the homography decomposition for vision-based control. Ph.D. thesis, INRIA
- Markley, F. L., and Crassidis, J. L. (2014). *Fundamentals of spacecraft attitude determination and control*. Springer.
- Montazeri, A., Can, A., and Imran, I. H. (2021). Chapter 3 - unmanned aerial systems: Autonomy, cognition, and control. In *Unmanned aerial systems*, eds. A. Koubaa, and A. T. Azar (Academic Press), Massachusetts, United States. doi:10.1016/B978-0-12-820276-0.00010-8
- Moutinho, A., Figueiró, M., and Azinheira, J. R. (2015). Attitude estimation in so (3): A comparative uav case study. *J. Intelligent Robotic Syst.* 80, 375–384. doi:10.1007/s10846-014-0170-2
- Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orb-slam: A versatile and accurate monocular slam system. *IEEE Trans. robotics* 31, 1147–1163. doi:10.1109/tro.2015.2463671
- Nemati, H., and Montazeri, A. (2018a). Analysis and design of a multi-channel time-varying sliding mode controller and its application in unmanned aerial vehicles. *IFAC-PapersOnLine* 51, 244–249. doi:10.1016/j.ifacol.2018.11.549
- Nemati, H., and Montazeri, A. (2018b). "Design and development of a novel controller for robust attitude stabilisation of an unmanned air vehicle for nuclear environments," in *Proceedings of the 2018 UKACC 12th International Conference on Control (CONTROL)*, Sheffield, UK, 05-07 September 2018 (IEEE), 373–378.
- Nemati, H., and Montazeri, A. (2019). Output feedback sliding mode control of quadcopter using imu navigation. *2019 IEEE Int. Conf. Mechatronics (ICM)*. vol. 1, 634–639. doi:10.1109/ICMECH.2019.8722899
- Nokhodberiz, N. S., Nemati, H., and Montazeri, A. (2019). Event-triggered based state estimation for autonomous operation of an aerial robotic vehicle. *IFAC-PapersOnLine* 52, 2348–2353. doi:10.1016/j.ifacol.2019.11.557
- Ozaki, R., and Kuroda, Y. (2021). EKF-based real-time self-attitude estimation with camera dnn learning landscape regularities. *IEEE Robotics Automation Lett.* 6, 1737–1744. doi:10.1109/Lra.2021.3060442
- Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). "Orb: An efficient alternative to sift or surf," in *Proceedings of the 2011 International conference on computer vision*, Barcelona, Spain, 06-13 November 2011.
- Sadeghzadeh-Nokhodberiz, N., Can, A., Stolkin, R., and Montazeri, A. (2021). Dynamics-based modified fast simultaneous localization and mapping for unmanned aerial vehicles with joint inertial sensor bias and drift estimation. *IEEE Access* 9, 120247–120260. doi:10.1109/access.2021.3106864
- Sadeghzadeh-Nokhodberiz, N., and Meskin, N. (2020). Protocol-based particle filtering and divergence estimation. *IEEE Syst. J.* 15, 4537–4544. doi:10.1109/jsyst.2020.3002907

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- Sadeghzadeh-Nokhodberiz, N., and Poshtan, J. (2016). Distributed interacting multiple filters for fault diagnosis of navigation sensors in a robotic system. *IEEE Trans. Syst. Man, Cybern. Syst.* 47, 1383–1393. doi:10.1109/tsmc.2016.2598782
- Sadeghzadeh-Nokhodberiz, N., Poshtan, J., and Shahrokhi, Z. (2014a). "Particle filtering based gyroscope fault and attitude estimation with uncertain dynamics fusing camera information," in Proceedings of the 2014 22nd Iranian Conference on Electrical Engineering, Tehran, Iran, 20–22 May 2014, 1221–1226.
- Sadeghzadeh-Nokhodberiz, N., Poshtan, J., Wagner, A., Nordheimer, E., and Badreddin, E. (2014b). Distributed observers for pose estimation in the presence of inertial sensory soft faults. *ISA Trans.* 53, 1307–1319. doi:10.1016/j.isatra.2014.04.002
- Sadeghzadeh-Nokhodberiz, N., Poshtan, J., Wagner, A., Nordheimer, E., and Badreddin, E. (2014c). Cascaded kalman and particle filters for photogrammetry based gyroscope drift and robot attitude estimation. *ISA Trans.* 53, 524–532. doi:10.1016/j.isatra.2013.10.002
- Steenbeek, A. (2020). *CNN based dense monocular visual SLAM for indoor mapping and autonomous exploration*. Master's thesis. Enschede, Netherlands: University of Twente.
- Wang, C.-L., Wang, T.-M., Liang, J.-H., Zhang, Y.-C., and Zhou, Y. (2013). Bearing-only visual slam for small unmanned aerial vehicles in gps-denied environments. *Int. J. Automation Comput.* 10, 387–396. doi:10.1007/s11633-013-0735-8
- White, J. H., and Beard, R. W. (2019). *The homography as a state transformation between frames in visual multi-target tracking*, Brigham Young University, Provo, UT



OPEN ACCESS

EDITED BY

Allahyar Montazeri,
Lancaster University, United Kingdom

REVIEWED BY

Roohollah Barzamani,
Islamic Azad University Central Tehran
Branch, Iran
Khoshnam Shojaei,
Islamic Azad University of Najafabad, Iran

*CORRESPONDENCE

Alireza Rastegarpanah,
✉ a.rastegarpanah@bham.ac.uk

[†]These authors have contributed equally
to this work

RECEIVED 15 May 2023

ACCEPTED 11 August 2023

PUBLISHED 29 August 2023

CITATION

Hathaway J, Shaarawy A, Akdeniz C,
Aflakian A, Stolkin R and Rastegarpanah A
(2023), Towards reuse and recycling of
lithium-ion batteries: tele-robotics for
disassembly of electric vehicle batteries.
Front. Robot. AI 10:1179296.
doi: 10.3389/frobt.2023.1179296

COPYRIGHT

© 2023 Hathaway, Shaarawy, Akdeniz,
Aflakian, Stolkin and Rastegarpanah. This
is an open-access article distributed
under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/).
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.

Towards reuse and recycling of lithium-ion batteries: tele-robotics for disassembly of electric vehicle batteries

Jamie Hathaway^{1,2†}, Abdelaziz Shaarawy^{1†}, Cansu Akdeniz¹,
Ali Aflakian^{1,2}, Rustam Stolkin^{1,2} and Alireza Rastegarpanah^{1,2*}

¹Department of Metallurgy and Materials Science, University of Birmingham, Birmingham, United Kingdom, ²The Faraday Institution, Quad One, Harwell Science and Innovation Campus, Didcot, United Kingdom

Disassembly of electric vehicle batteries is a critical stage in recovery, recycling and re-use of high-value battery materials, but is complicated by limited standardisation, design complexity, compounded by uncertainty and safety issues from varying end-of-life condition. Telerobotics presents an avenue for semi-autonomous robotic disassembly that addresses these challenges. However, it is suggested that quality and realism of the user's haptic interactions with the environment is important for precise, contact-rich and safety-critical tasks. To investigate this proposition, we demonstrate the disassembly of a Nissan Leaf 2011 module stack as a basis for a comparative study between a traditional asymmetric haptic-"cobot" master-slave framework and identical master and slave cobots based on task completion time and success rate metrics. We demonstrate across a range of disassembly tasks a time reduction of 22%–57% is achieved using identical cobots, yet this improvement arises chiefly from an expanded workspace and 1:1 positional mapping, and suffers a 10%–30% reduction in first attempt success rate. For unbolting and grasping, the realism of force feedback was comparatively less important than directional information encoded in the interaction, however, 1:1 force mapping strengthened environmental tactile cues for vacuum pick-and-place and contact cutting tasks.

KEYWORDS

robotic disassembly, telerobotics, lithium-ion batteries, EV batteries, haptic, teleoperation

1 Introduction

As a result of the increasing demand for electric vehicles (EVs) (Rietmann et al., 2020), a large number of EV batteries are expected to reach end of life. Owing to a combination of contained high-value materials such as lithium, nickel and cobalt (Thies et al., 2018), and a limited lifespan of 10–15 years (Ai et al., 2019), there is an increasing research interest towards EV battery disposal. Battery disassembly is a critical step to enable gateway testing and sorting of end-of-life (EoL) battery components for re-use, and recovery of high-purity materials for recycling. This remains a predominantly manual process for trained personnel, requiring a high degree of precision and attention (Tang and Zhou, 2004; Tan et al., 2021). While previous studies such as Lander et al. (2023) have emphasised the importance of

autonomous disassembly for reducing disassembly cost, limitations are presented due to the high degree of variability in EV battery models (Pehlken et al., 2017), as well as the lack of dexterity alongside the highly dynamic, unstructured work environments—chiefly due to variety of models and manufacturers and lack of design standardisation. Consequently, hybrid frameworks for EV battery disassembly have been proposed in which robot and human work closely and collaboratively (Tan et al., 2021). However, EV batteries have numerous associated thermal and chemical hazards due to residual charge held within the battery and risk of thermal runaway. The associated risks are further compounded for damaged batteries, which are challenging to disassemble autonomously due to further uncertainty in component end-of-life condition.

Telerobotics aims to mitigate the hazards of disassembly by enabling a human operator to carry out disassembly tasks remotely through a local interface while imparting some of the human operator's dexterity and fine motor control to the robot. Haptic devices are commonly employed in telemanipulation studies due to their ability to deliver force feedback to the operator while carrying out tasks. Nevertheless, one key challenge in haptic devices can be the master-slave asymmetry in regards of the kinematics (Li et al., 2020); hence, there exist a range of mapping schemes by which the motion of the master device can be mapped to that of the slave. Previously, it has been emphasised the nature of this interface is important for carrying out dangerous/sensitive tasks (Bernold, 2007). Furthermore, for disassembly in hazardous settings, it is suggested that the quality and realism of the feedback, and interface provided to the operator can greatly impact the level of task performance (Bolarinwa, 2022). Nonetheless, the effect of the user interface on such tasks has not been evaluated, and limited comparative studies exist for telerobotics in non-hazardous settings (Gliesche et al., 2020; Singh et al., 2020). Specifically, using telerobotics for the application of EV battery disassembly is not well explored, particularly, where damaged batteries present further critical challenges to the process of disassembly.

The main contribution of this work is a comparative study of the effect of differing haptic master devices on the level of task performance for sensitive disassembly tasks. This study further demonstrates potential application areas for telerobotics for battery disassembly, which thus far has been sparsely explored. The results provide insights into the feasibility of telerobotic disassembly of EV batteries, as well as areas of improvement for specific tasks. We establish a battery disassembly case study encompassing a range of disassembly tasks, such as unbolting, sorting and cutting, based on the sequence of pack-to-cell disassembly operations for a Nissan Leaf 2011. A series of tools were custom-designed in order to complete the tasks. We evaluate and compare the success rate and task execution time between a high-cost platform using two identical collaborative robots ("cobots") and a relatively low-cost platform using a haptic device paired with a single cobot. We examine causative factors for differences in performance between these two platforms for each task.

The remainder of the paper is structured accordingly: Section 2 provides a survey of related studies in disassembly and telerobotics. Section 3 introduces the experimental methodologies, including the experimental setup for carrying out disassembly tasks (Section 3.1), the control architecture for bilateral telemanipulation for both

platforms (Section 3.2), and the experimental case studies for different EV battery disassembly tasks (Section 3.3). Evaluation of each telemanipulation platform on the basis of objective performance measures is carried out in Section 4, and finally Section 5 concludes the paper.

2 Related work

Methods of battery disassembly can be broadly categorised into fully-manual, fully-autonomous, and semi-autonomous approaches. Given the hazards that EV battery disassembly environments pose against human operators, the literature has been more oriented towards developing autonomous and semi-autonomous approaches on the basis of potential improvements in efficiency and safety. Autonomous approaches aim to increase the efficiency of disassembly by allowing robots to plan and carry out repetitive tasks in unstructured environments through the use of visual and tactile feedback. Examples of such works can be found in Zorn et al. (2022); Zhang et al. (2023); Choux et al. (2021); Farhan et al. (2021). A common factor in these approaches is the use of labelling and detection methods to autonomously identify components and fasteners and construction of plans accordingly. However, this suffers from dependency on data and prior knowledge, and the risk of misidentification of battery components. For damaged batteries, existing prior knowledge datasets may not be suitable, or autonomous disassembly processes may fail; e.g., a fastener must be dislodged manually, or autonomous grasping fails due to structural deformation of components, requiring manual alignment. Furthermore, for sorting and lifting applications, the complexity of planning and collision avoidance remains an outstanding problem Zhou et al. (2019); Tang et al. (2022). Other approaches have aimed to reduce complexity of disassembly in various settings through tool design, such as gripper extensions Borràs et al. (2018) or flexible grippers Schmitt et al. (2011), to assist with unscrewing, grasping and other contact-rich manipulation tasks. However, even with such adaptations, autonomous approaches suffer from difficulties with generalisation to a range of tasks and across differing battery designs. Other studies such as Baazouzi et al. (2021) have examined optimisation of the EV battery disassembly process, emphasising the importance of battery design considerations for disassembly. As these are currently lacking for EVs, this introduces a high degree of variability and uncertainty into the disassembly process.

Recently, human-robot collaboration (HRC) has garnered attention for disassembly of end-of-life products, integrating both the robot's high efficiency in repetitive tasks and the human flexibility with higher cognition. However, there are few studies on this subject in the literature. In (Kay et al., 2022), a high-speed rotary cutting wheel was adapted to perform robotic cutting at various points of the battery module casing. The proposed framework allowed a robot to efficiently carry out semi-destructive disassembly processes while allowing human operators to rapidly sort the battery components and remove connectors. Frameworks have further been proposed to semi-automate the process of extracting and sorting different objects from an EV battery pack using a mobile manipulator (Rastegarpanah et al., 2021). This study uses the behaviour tree model, which connects different robot capabilities,

including navigation, object tracking and motion planning, for cognitive task execution and tracking in a modular format. Another study looked into human-/multi-robot-collaborations (Gil et al., 2007) in a disassembly scenario, where a technician and a number of robots coexist in the same environment interacting with each other to complete disassembly tasks. These presented articles demonstrate that HRC scenarios can offer both safety and time improvements over manual and autonomous disassembly strategies for EV LIBs. Nonetheless, issues such as coordination and allocation of tasks between robot and human and handling camera occlusion (Fan et al., 2022) remain outstanding topics of research. Furthermore, in some cases, it is not possible for robots to be in the same environment as humans, as the environment is considered hazardous for humans.

Telemanipulation has been commonly employed in the nuclear industry for delicate handling of wastes that present a radiation hazard for human personnel. For example, the use of gloveboxes is examined in Tokatli et al. (2021) in the context of telerobotics as a means of reducing radiation exposure risks to human personnel. Another study conducted real world experiments of a user performing nuclear decommissioning task via a unilateral teleoperated robotic system (Mizuno et al., 2023). The study conducted a comparison analysis between three teleoperation scenarios: fully manual, teaching-based, and planning-based, in terms of safety, cognitive demand, and preparation time. Findings showed that planning-based is the most time efficient, yet lacks safety. Although these articles presented successful use cases of teleoperation in hazardous environments, bilateral systems were not investigated to show the importance of force feedback to the user on performance. Haptic devices have been extensively used in many studies developing bilateral teleoperation systems for various applications (Giri et al., 2021). Haptic devices allow the operator to feel external force/torque, and further haptic cues that can be obtained in complex virtual environments. One study employed haptic devices as master devices to control dual 7-DOF serial arm manipulators to perform maintenance and repair tasks in nuclear power plants (Ju and Son, 2022). They developed a shared bilateral teleoperation system including three elementary technologies: egocentric teleoperation, virtual fixture, and vibration suppression control to assist the human operator in performing shaft-and clutch-based peg-in-hole tasks. Furthermore, they conducted human-centric evaluations to measure the performance in terms of completion time, trajectory length, and human effort. The results of this study showed that haptic cues improved task performance significantly. Additionally, in Shen et al. (2021), the authors propose a framework for a single-master-multiple-slave manipulator system that enables remote regulation of cooperative tasks. They utilize an adaptive non-singular terminal sliding-mode (ANTSM) method to address challenges such as time-varying delays, external disturbances, and modeling uncertainties. Other approaches have aimed to examine the effect of virtual constraints and force feedback guidance on task performance. Such approaches have been proposed in He and Chen (2008), based on imposition of rotational constraints on a component when axially aligned with its target. A related concept is proposed in Abi-Farraj et al. (2016), based on a shared control approach to remote object manipulation using visual information, and extended to a real environment in (Abi-Farraj et al., 2018). In this way, the user is assisted during

a grasping task by constraining the gripper orientation to the surface normal of a virtual sphere, centred on the object centre of mass. While performance improvements for these tasks have been documented, these approaches limit the flexibility provided to the operator; such constraints are often task-specific and susceptible to failure in edge cases, for example, if the autonomously identified grasp point is not suitable. Alternative approaches for sorting and separation of objects in an unstructured environment have been proposed based on haptic guidance cues Abi-Farraj et al. (2020). In this way, a visual inspection of the object is used to generate suitable grasp points, and the user is guided towards the suggested grasp point through tactile cues and vibration without constraining the free manipulation of the robot.

The majority of considered studies have focused on telerobotics for assembly tasks, nuclear decommissioning and waste handling applications. However, there is a paucity of research surrounding the use of telerobotics in an EV disassembly context. The authors in Meng et al. (2022) highlight the potential of teleoperation in handling hazardous manufacturing environments and discusses various autonomy levels of intelligent teleoperation for disassembly. They have also provided a forward-looking overview on how to use telerobotics for the application of disassembly of electric vehicle lithium-ion batteries. Furthermore, while many studies have explored the effect of shared autonomy and control system design on task performance, a limited number of works have investigated the nature of the interface exposed to the operator and its effects on task performance. Such a comparison has been explored in Gliesche et al. (2020) between a keyboard-mouse setup, XBOX controller and a bilateral setup using a haptic device. For a range of patient care tasks, involving grasping and moving a remote and adjusting a camera view, a haptic device was found to improve the speed at which operators carried out tasks, however no subjective differences in mental load were identified. In Singh et al. (2020), a framework was presented using a pair of identical cobots, and explores human-centric performance metrics between methods of rendering guidance and environmental forces to the user on the performance of a peg-in-hole task. However, the range of tasks remains limited and the effects of the interface remain unclear for the wider range of tasks featuring in a disassembly environment.

3 Methodology

3.1 Experimental setup

This study focuses on the Nissan Leaf 2011 battery pack as a case study to demonstrate a range of disassembly tasks using telemanipulation. The Nissan Leaf 2011 pack comprises 192 cells enclosed within 48 modules, arranged with two forward vertical stacks of 12 modules and a single rear horizontal stack of 24 modules. An overview of the main disassembly stages for pack-to-cell disassembly is given in Table 1; for brevity, this is provided for only the forward module stacks. The level of autonomy presented in Table 1 was inferred from understanding of the current manual disassembly process and use of design features, such as types of fasteners and accessibility of components. In general, the range of semi-/fully autonomous tasks will vary between battery designs and manufacturers, however, overall, there is a motivation to increase

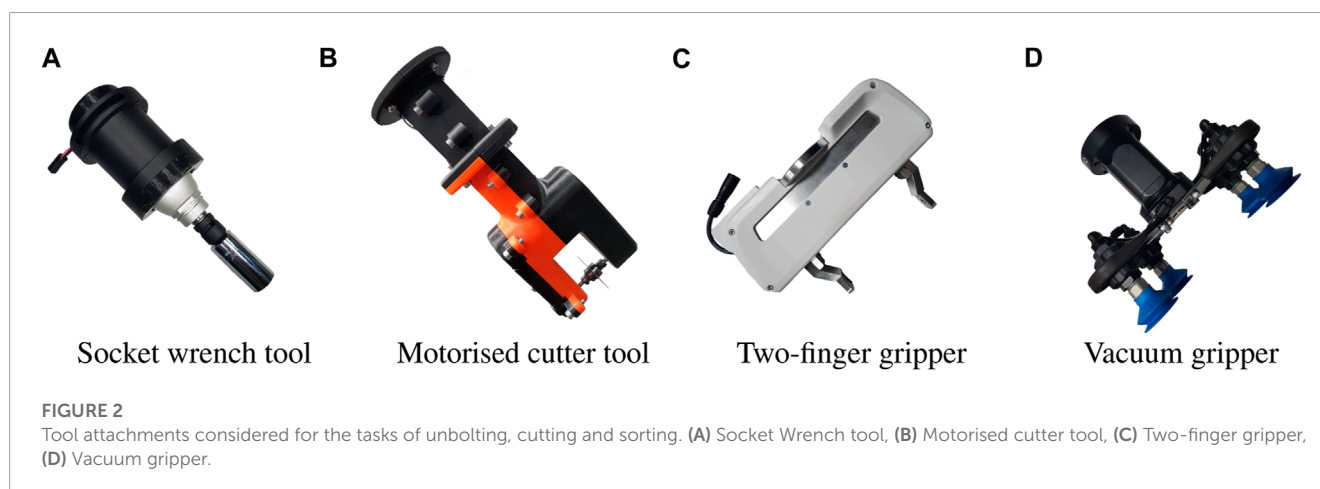
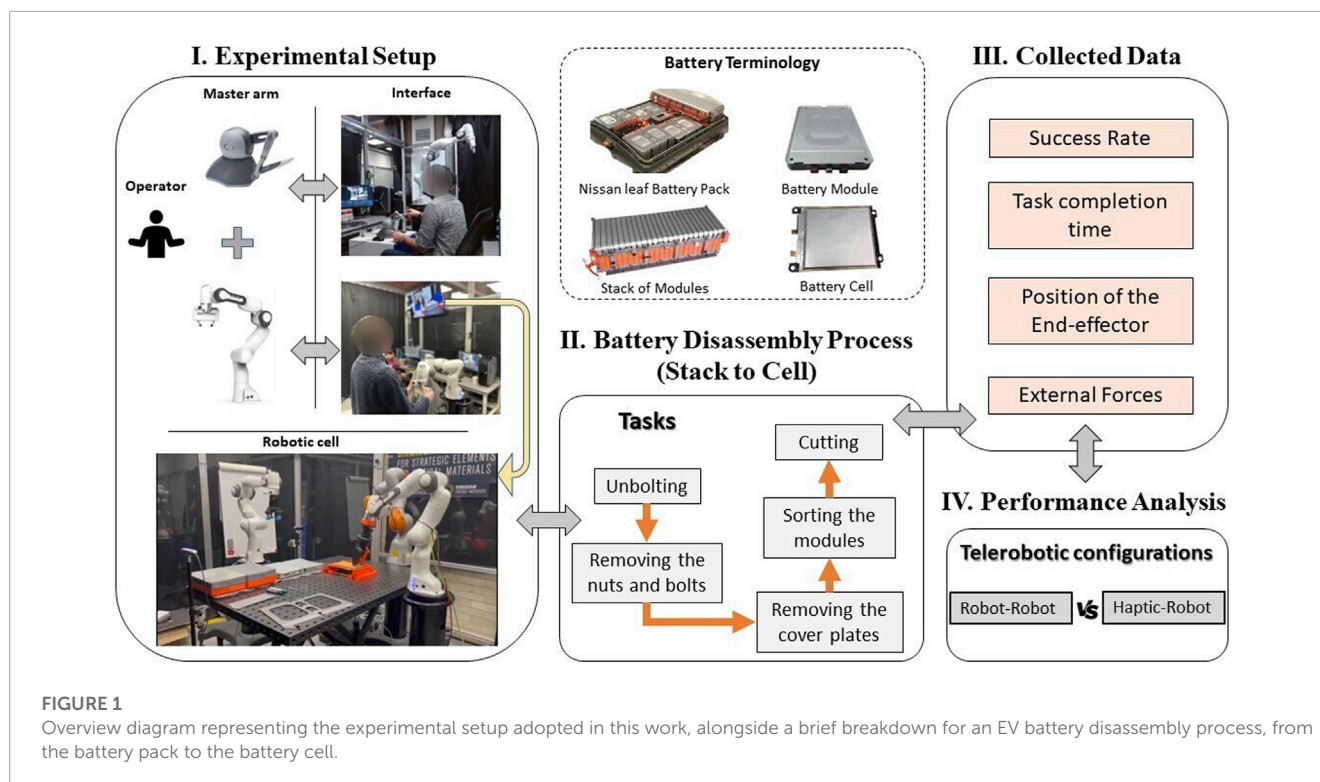
TABLE 1 Sequence of disassembly operations for pack-to-cell disassembly of Nissan Leaf 2011. For brevity, only disassembly of the front module stack is considered. The type of task can be classified as fully manual (M), requiring specialised tools or dexterous hand-manipulation, semi-autonomous (S-At), where a robot can accomplish the task with the assistance of a human operator, and fully autonomous (At) tasks accomplishable without human intervention. A contiguous sequence of semi-autonomous processes can be observed for disassembly and sorting of the module stack; tasks considered in this study are emphasised.

Step #	Disassembly task		Type
1	Top case	Remove service plug retainer	M
2		Remove upper case bolts and lift top case	At
3	Battery controller	Remove mounting bolts	S-At
4		Disconnect harness connectors and remove battery controller	M
5	Junction box and harnesses	Disconnect interlock circuit harness and heater harness connectors	M
6		Remove mounting nuts and front stack connecting bus-bar	S-At
7		Remove battery member pipe	S-At
8		Remove junction box cover	M
9		Remove central bus bar bolts and remove central bus bar	S-At
10		Remove current sensor bus bar mounting bolt	S-At
11		Remove switch bracket mounting bolts	S-At
12		Invert switch bracket, disconnect harnesses and remove switch bracket	M
13		Remove high voltage (HV) harness bolts and remove HV harnesses	S-At
14		Disconnect voltage and temperature sensor harnesses	M
15		Remove junction box mounting nuts and junction box	M
16	Heaters	Disconnect harness connectors from heater and heater relay unit	M
17		Remove heater and heater relay mounting nuts	S-At
18		Remove heater controller unit and heaters	M
19	Front module stack(s)	Remove stack mounting nuts	At
20		Extract module stack	At
21		Remove bus bar cover	M
22		Remove bus bar terminal mounting bolts and mounting screws	S-At
23		Remove end plate bolts	At
24		Remove end plate	S-At
25		Electrical test and sort modules	S-At
26	Module	Separate module cover	S-At
27		Glue separation	At
28		Separate cell tabs from terminal assembly	S-At

the level of autonomy to decrease disassembly costs (Lander et al., 2023). We formulate a basic battery disassembly workstation design consisting of 2 Franka Emika Panda cobots and a Phantom Omni haptic device. The Franka Emika Panda is a 7 degree of freedom (DoF) robot with a 3 kg payload. A combination of torque control and onboard torque sensing capabilities and low payload makes this robot suitable for accomplishing precise assembly and disassembly tasks in a shared human-robot workspace. The Phantom Omni comprises a 6 DoF handheld articulated arm with independent control of force feedback along 3 axes, with a maximum force capability of 3.3N. All devices were connected in a ROS network to a desktop with Intel(R) Core(TM) i7-8086K 8-core processor with

4 GHz base clock and 32 GB RAM. An overview of the experimental setup is shown in Figure 1.

Based on the disassembly sequence in Table 1, we consider a selection of repetitive tasks identified in related studies (Tan et al., 2021) where robots provide an advantage over manual disassembly. These consist of unbolting, such as of fasteners connecting a stack of modules; removal and sorting (pick and place) of disassembled waste components, and cutting to mechanically separate components where fasteners cannot be removed non-destructively. To deal with the range of tasks presented in the case of battery disassembly, a range of commercially available and custom designed tools were employed, shown in Figure 2. For this



study, we design a custom socket wrench tool that can be employed with a range of fastener sizes. For cutting, a motorised slitting saw tool was designed specifically for the low-payload cobots presented to accomplish low-power cutting tasks. For pick and place, the Franka Hand two-finger gripper was employed for grasping thin, light wastes such as bolts and plates, while the Robotiq EPick suction gripper is applied for larger, heavier materials such as battery modules.

3.1.1 Master-slave telemanipulation using a haptic device as master and a cobot as slave

A common framework to accomplish telemanipulation tasks is to use a lower-cost, handheld platform such as a haptic device in combination with a robot in a master-slave configuration. In this case, we consider the case of a Phantom Omni haptic device

and a Panda cobot (Figure 3). Due to a mismatch in the DoFs of the Omni and Panda robot - more generally, any over-actuated robot—there is a limitation that not all degrees of freedom of the robot can be controlled independently, reducing the amount of control provided to the user. Typically, this requires a definition of a mapping function that maps the joint positions of the haptic device to that of the slave arm. Such a mapping can be achieved by mirroring the Cartesian pose or twist of the haptic end-effector to that of the slave robot. This provides a more intuitive framework for the user to position the robot to accomplish tasks. However, this typically renders the control vulnerable to singularities and joint limits, due to the solution of inverse kinematics/dynamics to compute the required joint-space motion. Furthermore, the smaller scale of motion afforded by the Omni in both joint and task space implies a compromise between speed of coarse motion, where the

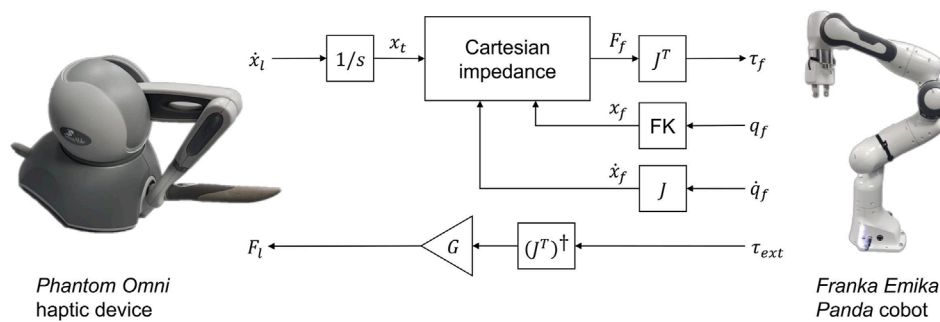


FIGURE 3

Outline of telemanipulation framework consisting of Phantom Omni haptic device and Franka Emika Panda cobot. FK refers to the forward kinematic mapping of follower joint configuration q_f to end-effector pose x_f . Note the transformation matrices relating the base frames of the haptic and cobot are omitted for simplicity.

motion of the haptic device is scaled to achieve larger motions on the robot; or precision of fine positional alignment, where the motion of the haptic device is mapped directly to the robot or even reduced. Moreover, due to the limited force capabilities of the haptic device, the force feedback cannot be mapped from the robot to the haptic 1:1. Therefore, the feedback can potentially feel unnatural or deliver insufficient cues for the operator, such as when exceeding safety limits imposed on force.

3.1.2 Master-slave telemanipulation using two identical cobots

Alternatively, in this work we consider a platform consisting of two identical Franka Emika Panda cobots operating in a master-slave configuration. With this approach, the user directly manipulates the master Franka arm, whose motions are directly mirrored to that of the slave arm. Owing to the identical configuration of both robots, both joint position and torques can be mapped between the robots 1:1. This 1:1 mapping results in natural and responsive feedback being delivered to the user. However, this also poses hazards as the user will be potentially exposed to the full forces involved in a specific task. The joint space control of both arms furthermore has advantages in the problem of singularities and joint limits, which are present in mapping the Cartesian end-effector pose of the master to the task space pose of the slave robot. However, the lack of task-space control has potential drawbacks when accomplishing tasks along specific task directions, such as removing bolts or cutting, more difficult and less intuitive for the user.

3.2 Control strategy

To achieve a virtual coupling between master and slave devices, we first consider the dynamic equation of a rigid N -link manipulator in joint space

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q}) + \mathbf{g}(q) = \tau_{\text{ext}} + \tau \quad (1)$$

where $\mathbf{M} \in \mathbb{R}^{N \times N}$, $\mathbf{C}(q, \dot{q}) \in \mathbb{R}^{N \times N}$, $\mathbf{g}(q) \in \mathbb{R}^{1 \times N}$ are the joint-space inertia matrix, Coriolis and centrifugal matrix and gravitational torques respectively, and $\tau_{\text{ext}} \in \mathbb{R}^{1 \times N}$, $\tau \in \mathbb{R}^{1 \times N}$ are the vectors of

external and control torques acting on each link respectively. We denote the joint configurations q and command torques τ by subscripts l, f for master and slave respectively.

For the haptic device, 1:1 mapping between q_l , q_f cannot be achieved in practice due to a mismatch in the number of kinematic degrees of freedom and joint ranges compared with the Franka arm. Hence, it is required to either map the joint space onto a reduced subset of the full joint space of the robot, or operate in a mutual task space. In this work, 6DoF Cartesian mapping is employed for motion control of the slave Franka arm with the Phantom Omni master arm. As the human operator moves the master arm in its workspace, delta Cartesian pose ${}^P\mathbf{T}_\Delta$ is computed and then mapped to the end-effector delta pose ${}^F\mathbf{T}_\Delta$ via the workspace transformation (Figure 4):

$${}^F\mathbf{T}_\Delta = {}^P_F\mathbf{T} \cdot {}^P\mathbf{T}_\Delta = \begin{bmatrix} {}^F\mathbf{R}_\Delta & {}^F\mathbf{t}_\Delta \\ 0 & 1 \end{bmatrix} \quad (2)$$

where ${}^P_F\mathbf{T}$ is the homogeneous transformation matrix from Franka arm base frame to Phantom Omni base frame, ${}^F\mathbf{T}_\Delta$, and ${}^P\mathbf{T}_\Delta$ represent the delta transformation matrices for Franka and Phantom respectively. Position and orientation components of ${}^P\mathbf{T}_\Delta$ are assigned to the desired velocity \dot{x}_l to compute the task space pose error $e_x = x_f - x_l$ for the slave arm. Thus, the slave arm control law is:

$$\tau_f = \mathbf{J}^T (-\mathbf{K}_p e_x - \mathbf{K}_d \dot{q}_f) + \mathbf{C}(q_f) + \mathbf{g}(q_f) \quad (3)$$

where $\mathbf{J} \in \mathbb{R}^{N \times 6}$ is the slave manipulator Jacobian mapping joint to end-effector velocities and \mathbf{K}_p , \mathbf{K}_d are controller stiffness and damping matrices respectively. This results in the desired closed-loop dynamic behaviour from Eq. 1 (where $\Lambda \in \mathbb{R}^{6 \times 6}$ is the operational space inertia matrix):

$$\Lambda \ddot{x}_f + \mathbf{K}_d \dot{x}_f + \mathbf{K}_p e_x = F_{\text{ext}} \quad (4)$$

Force feedback is an essential part of a bilateral teleoperation system in which a force feedback is maintained at control frequency. This helps the user to have a tactile perception of the slave robot's environment. In a similar way to (2), the external force vector F_{ext} experienced at the Franka end-effector is transformed with respect to the Phantom so the user is able to perceive it as force F_l as follows:

$$F_l = G_F {}^P_F\mathbf{T}^{-1} \cdot F_{\text{ext}} \quad (5)$$

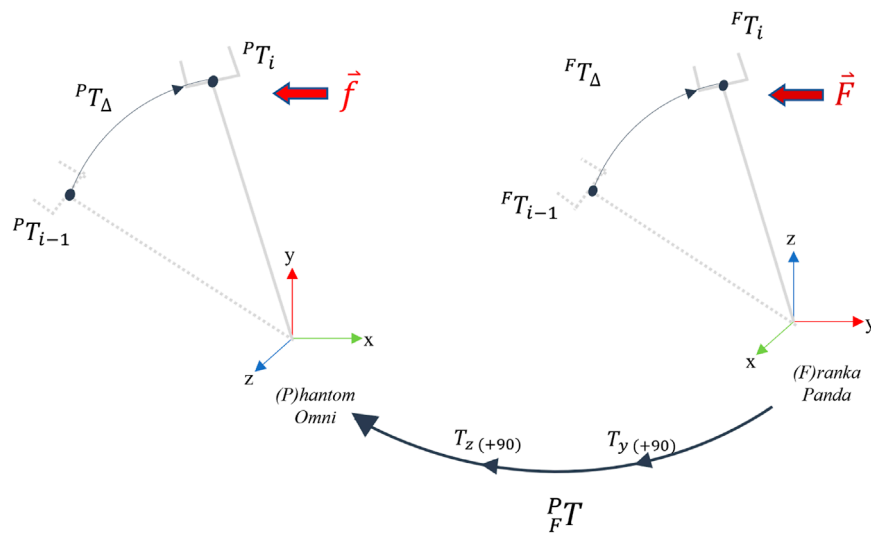


FIGURE 4

Cartesian mapping for the master-slave telemanipulation setup between the *Phantom Omni* haptic device and *Franka Panda* robot arm.

Due to the mismatch in force capabilities between master and slave devices, the feedback is scaled by a factor $G = 0.1$. This factor was determined experimentally by comparing the maximum expected force across all tasks from preliminary data and scaling to the maximum force capabilities of the haptic device (3.3N). To distort the force feedback to a minimal extent, and to maintain consistency with the constant 1:1 feedback of the identical cobot setup, this factor was held constant across all trials. A necessity of the task space control scheme is the remediation of singularities and joint limits. In this case, a null space position regulation term is added to (3) to avoid joint limits. Unlike pseudoinverse control where large solution values can be obtained, with the controller design in Eq. 3, singularities will be observed as the torque command τ_f tending to zero along the singular directions.

For the Franka arm, a joint impedance control scheme is used that directly maps the joint configuration of the master arm to the slave arm (Figure 5). The control law is defined as follows:

$$\tau_f = -K_p e_q - K_d \dot{e}_q + C(q_f) + g(q_f) \quad (6)$$

where $e_q = q_f - q_l$ is the joint space error. This results in the closed-loop dynamics in joint space for the slave arm

$$M\ddot{q}_f + K_d \dot{e}_q + K_p e_q = \tau_{\text{ext}} \quad (7)$$

To provide force feedback to the user, the master control torques are calculated as:

$$\tau_l = \tau_{\text{ext}} - K_{d,l} \dot{q}_l \quad (8)$$

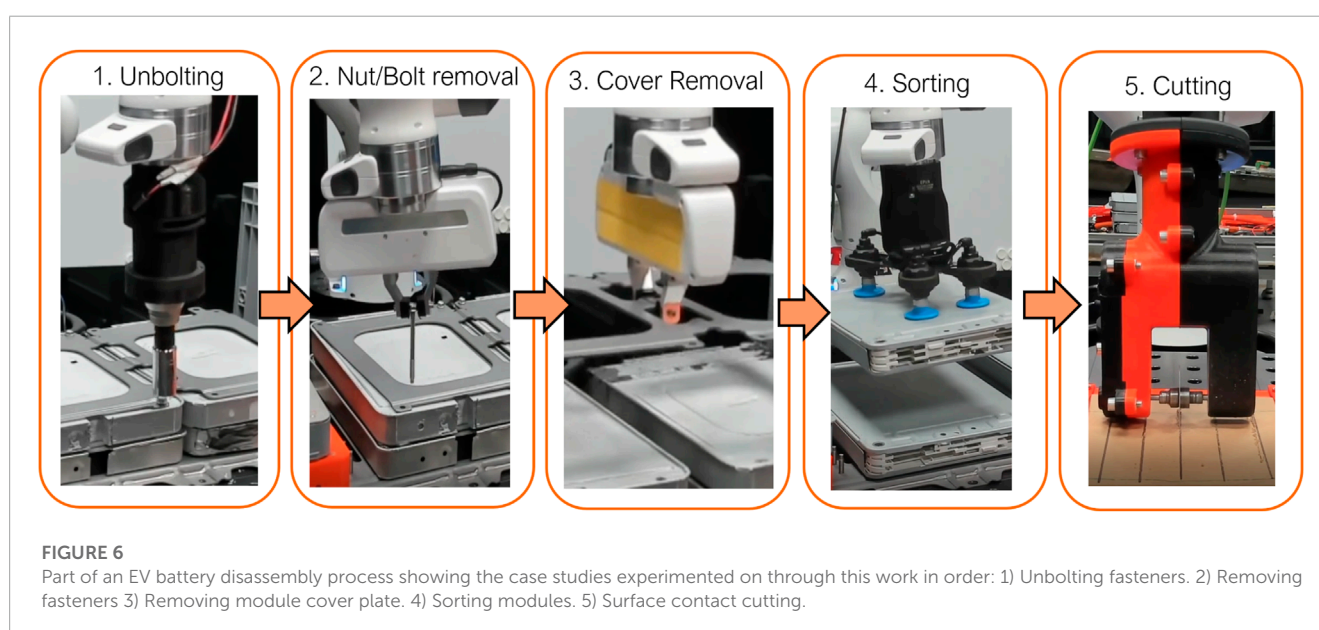
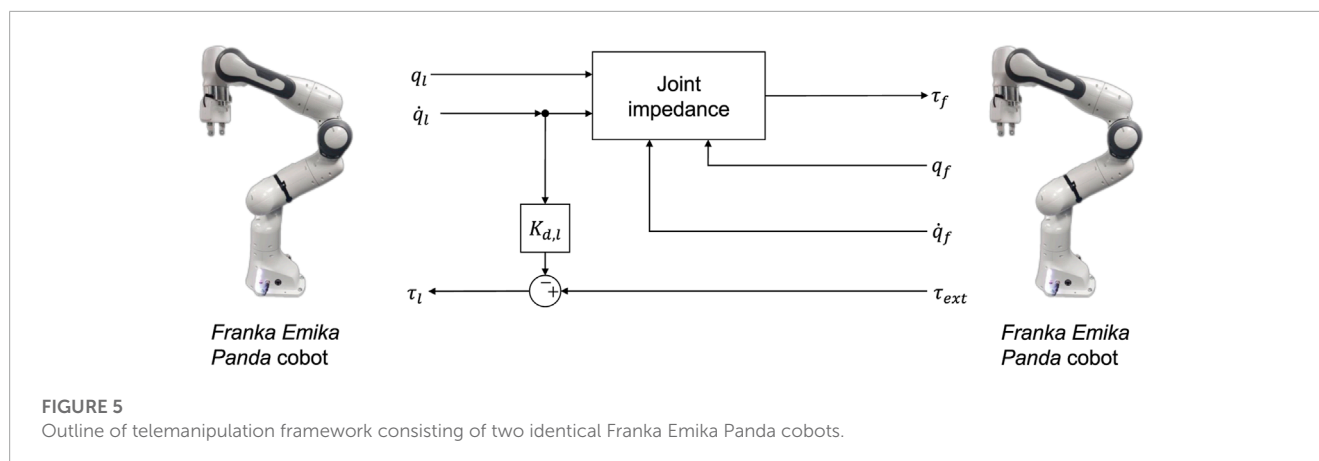
where the force feedback is computed directly as the estimated external torque applied to the slave arm. Note in general the model parameters C , g are subject to uncertainty, which results in reduced tracking performance of both controllers and the presence of steady-state error. Tuning of the control gains K_p , K_d was accomplished similarly to G via preliminary experiments, similarly to other comparative studies such as Nakanishi et al. (2008). In this case, the

value of K_p was increased for each of the control strategies on the slave arm in isolation before instability occurred, while K_d was set to obtain a critically damped behaviour. This provides the best tracking performance in free space and reduces the influence of uncertainties in the model parameters C and g . For the joint control strategy, the gains were scaled down according to the torque capabilities of each joint. Due to the combination of higher forces applied to the master with the low stiffness of the human operator, an additional damping term $K_{d,l}$ is added to the response of the master. The value of $K_{d,l}$ was selected by identifying the minimal damping value assigned to all joints to suppress oscillations arising from feedback effects from the bilateral master-slave coupling.

3.3 Disassembly case studies

A contiguous sequence of tasks relating to the disassembly and extraction of a stack of modules from the Nissan Leaf 2011 battery pack were identified as feasible targets for robotic disassembly, presented in Figure 6. The considered case study encapsulates a range of contact-rich tasks required to disassemble and recover the battery modules from an individual stack. The modules contain the active material within the battery and hence are considered a safety-critical aspect of the disassembly process due to the associated handling risks, such as risk of shorting the terminals and risk of mechanical shock from mishandling. Each task was attempted five times (trials). We evaluate the task completion time and success rate for each task for both platforms. Additionally, to evaluate the difference in task completion times in a timescale-invariant manner, the standardised mean difference (SMD) effect size metric was used. Given the mean and standard deviation $(\bar{t}_h, \sigma_{t_h})$, $(\bar{t}_f, \sigma_{t_f})$ for each platform respectively, the SMD can be defined as:

$$\text{SMD} = \frac{\bar{t}_h - \bar{t}_f}{\sqrt{\sigma_{t_h}^2 + \sigma_{t_f}^2}} \quad (9)$$



and provides an evaluation of the effect of each platform on completion time relative to the amount trial-to-trial variation. During each trial, the robot tool tip pose and estimated external wrench were recorded to examine the operator behaviour during each trial and semantically identify key stages of the task.

For each task, the operator is provided with two fixed camera views of the module stack, and otherwise does not have a direct view of the scene. These views are held constant for all tasks with the exception of cutting, where the operator is instead provided two camera views of a material holder containing the cutting workpiece. Prior to commencing each task, both master and slave robots are initialised to a home position in joint space held constant for all tasks. As the focus of this work is not on human factor analysis and subjective measures of performance based on end-user evaluations, but on objective measures of task performance, the operator is assumed to have familiarity and prior experience with both teleoperation platforms. In addition, each operator was given 10 min of training with each task with both telemanipulation systems before first attempting each task. Four expert operators participated in the study. For each trial, a single operator was selected

randomly to perform the task. A more detailed description of each task is provided as follows.

3.3.1 Unbolting

This task considers the unbolting of a set of 4 fasteners constraining an individual stack of modules. The operator has access to a motorised universal socket wrench tool mounted at the robot's wrist. The operator is responsible for aligning the socket wrench with the bolt and operating the tool to unscrew the fastener. The task is considered successful if the bolt is removable by hand without any further unscrewing action. If the success condition is not met after the first attempt, or the configured force thresholds (40N) of the robot are exceeded, the task is considered a failure.

3.3.2 Removing the fasteners

This case study follows from the previous unbolting stage by removing each fastener from the stack before removal of the cover. For this task, the Franka hand was used with a configured grasping force of 50N. The operator was required to remove a set of 8 fasteners by maneuvering the hand towards each of the fasteners, grasping and

removing the bolt from the stack and depositing into a container. Due to the limitation of the camera views of the object, the operator must use a mixture of tactile and visual exploration to successfully remove the fasteners. For each bolt, the failure condition is met if the bolt is not grasped after the first grasp attempt, or if the grasp is lost outside of the target container.

3.3.3 Removing the module cover plate

After the removal of the fasteners, the operator must then remove the module cover plate to access the underlying module stack. Due to the weight and geometry of the cover plate, it is essential for the operator to find and manoeuvre towards a good grasp point to ensure safe transportation of the cover during removal. The configured grasping force was increased to 60N for all experiments. The failure condition is met if the first grasp is unsuccessful or the grasp is lost during the transportation of the cover.

3.3.4 Sorting modules

This case study considers unstacking and sorting of the EV battery modules using a vacuum suction gripper. The operator must remove a pair of modules from a stack and deposit them into a container. This task requires visual positioning of the gripper onto a suitable surface for grasping, while furthermore maintaining contact to engage the suction cups with the material without exceeding the force limits of the robot. Similarly to the cover removal case study, the failure condition is met if the first grasp is unsuccessful, or if the grasp is lost during transportation.

3.3.5 Contact cutting

Additional semi-destructive disassembly tasks may be carried out to further disassemble the individual modules to access the Li-ion cells or to separate and remove fasteners or connectors that are not amenable to non-destructive disassembly. We consider the case study of contact cutting of a planar material along a predefined visually marked desired path using a slitting saw tool. Due to the limited availability of battery materials and the power limitations of the spindle, we consider cutting a cardboard sheet as a benchmark for the more general process of cutting different planar materials. The operator must use tactile feedback to control the cutting force while also using visual feedback to achieve precise positioning along the path. The desired cutting path consists of a window of ± 2.5 mm measured from the centroid of the marked path. If the cut deviates from the path greater than this threshold, or if the cut is incomplete along parts of the desired path, the failure condition is met.

4 Results and discussion

4.1 Success rate and completion time

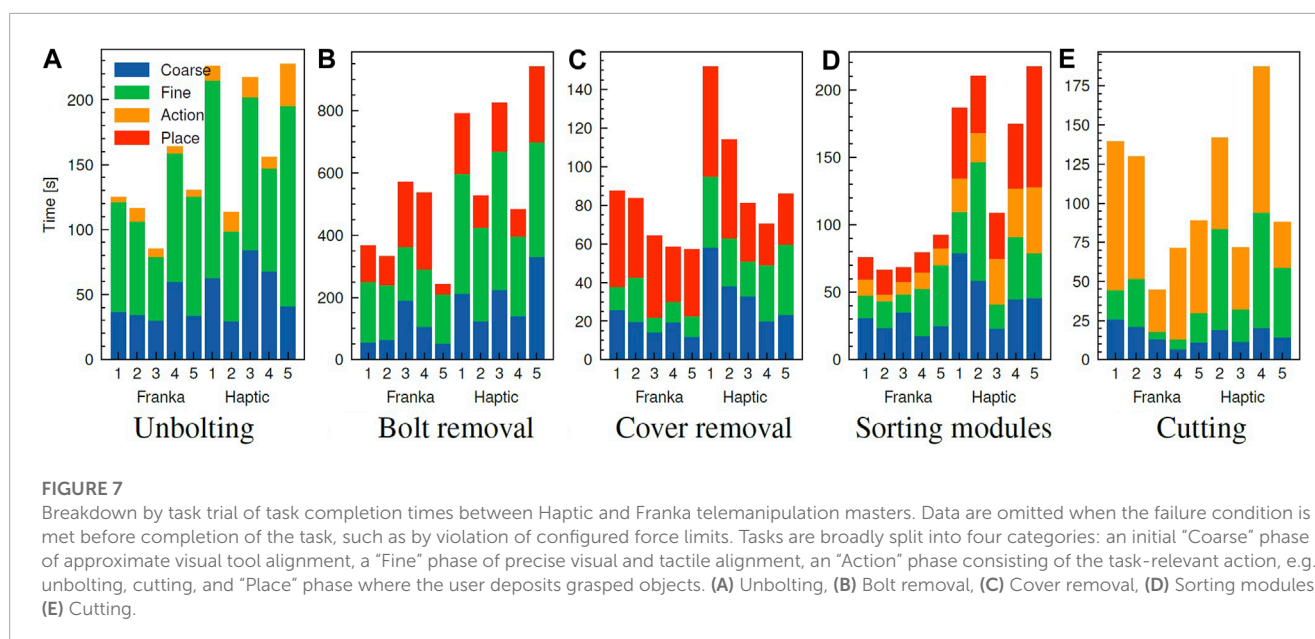
For each case study, the overall *success rate* and average *task completion time* over all trials are summarised in Table 2. For all tasks, the success rate is observed to exceed or equal 50% with both platforms, with the lowest success rate of 50% observed for the task of bolt removal with an identical cobot. For the majority of the tasks, a success rate of over 75% is achieved, demonstrating the

feasibility of the module stack disassembly process with both cobot and haptic device telerobotics platforms. Between each platform, the success rate is broadly comparable for unbolting and bolt removal, where the operator success rate is 10%–15% higher than the identical cobot case. Of the total bolt removal trials, failure to grasp the bolt due to misalignment contributed to 17% of the failures, increasing to 30% with identical cobots. Other causes of failure were due to loss of grasp during bolt extraction (15% for both platforms) and releasing the bolt outside of the box (5%, 2.5% for haptic and cobot respectively). Cover removal is notably the most simple task due to the near-planar approach trajectory; given that the operator managed to reach a firm grasp point on the cover plate, which was achieved 100% in all operators' trials with both platforms. On the other hand, the largest differences are observed with unstacking and cutting, where in the former case the identical cobot is outperformed by the haptic device by 30%, whereas in the latter case, the inverse is true, with a 20% improvement in success rate. Unstacking findings can be justified due to the design mechanism of the vacuum suction gripper, where the gripper has to be in a direct contact and, more importantly, perpendicular orientation with respect to the module surface to achieve a successful grip. Hence, manipulating the suction gripper's position/orientation to achieve this in Cartesian space with the haptic platform showed higher success rate than in joint space with the identical cobot platform, where gripping failure cases occurred due to orientation misalignment. Furthermore, the findings of cutting trials emphasize the significance of force feedback. Identical cobot platform showed higher success rate because the force feedback is maintained at a 1:1 scale, whilst in haptic platform, it is scaled down due to limited force capabilities as (5). Therefore, the user did not experience the full scaled forces exerted on the end-effector while cutting. This is corroborated by the causes of failure; which were due to deviations from the desired path for the Franka, while the haptic platform failures were due to shallow, incomplete cutting of the material or exceeding the force limits of the robot.

In Figure 7, a detailed breakdown of task completion times by trial is presented, indicating the time spent in each stage of the trials. Depending on the task, these stages subdivide each task consecutively, from start to completion, into categories: a "Coarse" stage where the operator telemanipulates the robot through rough visual alignment of the tool, followed by a "Fine" stage utilising tactile and visual feedback for precise positioning of the tool towards the interaction point. In the "Action" stage, the operator interacts with the component to perform a task after successful alignment of the tool. These actions correspond to unbolting, gripping and engagement of the cutter for nut/bolt removal, sorting and cutting respectively. Finally, the "Place" stage is encountered for grasping tasks after a successful grasp of the target object, during which the operator moves the grasped object and releases at a target position. The coarse and fine stages were annotated according to the proximity of the tool to the task-dependent target; a distance of the tool within 5 cm of the target—for example, bolts and cover grasp points—indicates the fine alignment stage. The place stage was specified for coarse motions performed after grasping, which reverts to the coarse phase of motion after release of the grasped object. The action phase was annotated based on task-specific position and force thresholds. For example, for the sorting task, the task-relevant action is engagement and evacuation of the suction cups to grasp the

TABLE 2 Comparison of performance metrics for disassembly tasks for telemanipulation with Phantom Omni haptic device and identical cobot (Franka) platforms.

	Haptic		Franka	
	Success rate [%]	Avg. Task time [s]	Success rate [%]	Avg. Task time [s]
Unbolting (4×)	95	188 ± 23	85	124 ± 13
Nut/bolt removal (8×)	63	713 ± 89	50	410 ± 63
Cover removal	100	101 ± 15	100	70 ± 6
Sorting modules (2×)	90	179 ± 19	60	77 ± 5
Cutting	60	122 ± 26	80	95 ± 18



module. From the 4 stages breakdown presented in Figure 7, one can have a clear interpretation of what stages the operators spent most of the task time at, indicating their exerted efforts in completing each of the stages with respect to the two comparable platforms.

From an overview of Figure 7, the identical cobot platform achieved shorter completion times on average across all tasks, compared to the haptic platform. Additionally, the highest difference in average time between the two platforms is observed for the bolt removal and sorting tasks, on average requiring ~ 1.7 – $2.3\times$ longer to complete with the haptic device respectively. Absolute task completion times were more consistent between trials with the identical cobot platform for all tasks. For a comparative overview of the effect on completion time across tasks, we employ the SMD effect size metric. Values of $|SMD| \geq 0.8$ implies a significant effect, and the converse indicates a small or marginal effect. For the considered module stack disassembly case study, the highest proportion of time was spent unbolting and removing the retaining fasteners. This is observed from Figures 7A,B, where the “fine” stage occupies 60%, 63%, and 47%, 50% of the average task completion time, respectively, in both platforms; therefore, the fine alignment stage was the primary contributor to the reduction in overall task completion speed. This indicates that regardless of the platform used, these two tasks, among the other disassembly tasks, were the most effort

demanding; requiring a combination of precise visual and tactile alignment of the tool to successfully unbolt/remove the fasteners. However, the SMD for unbolting and bolt removal respectively were 1.09 and 1.24, which indicates large improvements afforded by the identical cobot setup.

4.2 Case studies

Figures 8–11 present the position graphs of the Franka slave’s end-effector in Cartesian space, as well as the external forces experienced during a sampled trial from each task in the case study with both platforms. The figures also display the 4 categorised stages (“Coarse”, “Fine”, “Action”, “Place”) color-shaded in each graph’s background corresponding to each stage, as introduced in Figure 7, throughout the time span of the presented trial. In Figure 8, spikes in the force z plot enclosed in the yellow region (“action” stage) correspond to the action of *unbolting* of the fasteners using the wrench tool. Similarly, Figure 9 represents the *bolt removal* task in which fasteners are grasped, lifted up from their holes, and placed in a container; thus, the position z spikes in the red region (“place” stage). Interestingly, despite the differing scale of force feedback between the two platforms, the magnitude of external force observed

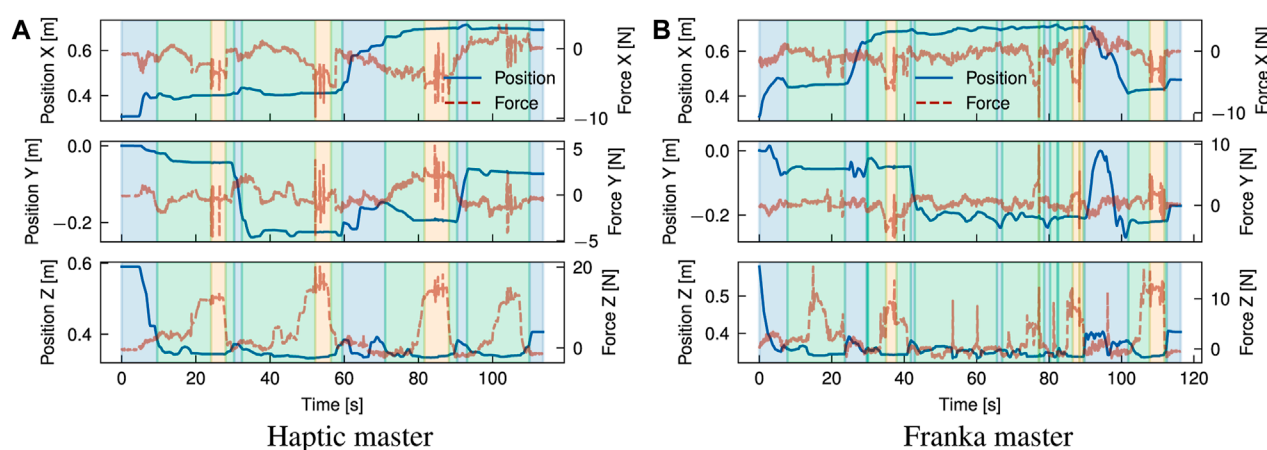


FIGURE 8

Overlay of position and estimated external forces at the end-effector for the unbolting task (trial 2) using a motorised universal socket wrench tool. Increases in force (~ 10 – 15 N) in the surface normal direction (Z) indicate establishment of contact with the bolt and tactile exploration to identify correct alignment of the tool. Execution of unbolting is marked by rapid force fluctuations around these peaks. (A) Haptic master, (B) Franka master.

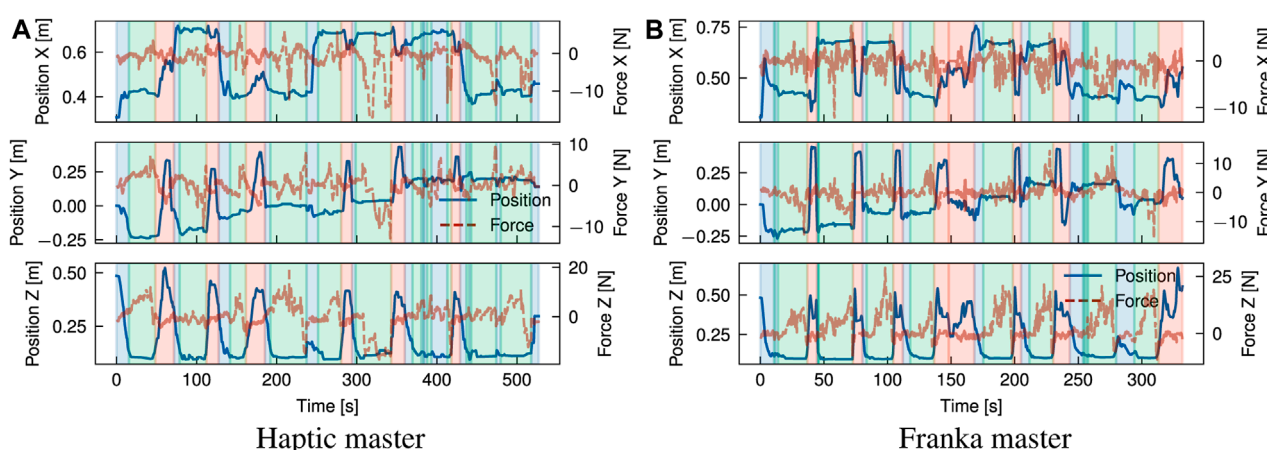


FIGURE 9

Overlay of position and estimated external forces at the end-effector for the bolt removal task (trial 2) using a 2-finger gripper. The position alignment can be categorised broadly into rapid visual alignment (indicated by a rapid decrease in Z position), followed by a slower phase of precise visual and tactile alignment (plateau in Z position). Variations in force along the X-Y plane are indicative of tactile exploration to find suitable grasp points for the bolt. (A) Haptic master, (B) Franka master.

over time for the bolt removal task is similar for both platforms. This suggests that the realism and scaling of the force feedback is less important for accomplishing precise grasping tasks for disassembly than the directional guidance provided by the force feedback during contact.

Some tasks involved placing components after disassembly into a waste/recycle bin, like in the cover removal task, where most of the time was spent on post-grasp manipulation, i.e., the “place” stage. Operators spent on average 39, and 37 s placing the objects which averaged 56%, and 35% of the task completion time with cobot and haptic platform, respectively. Around 30% of the time was spent in the “place” stage for the cobot platform; due to the lack of Cartesian control, the operator reached joint configurations where large reconfigurations were required between tasks to access

the workspace. This is evident in Figure 9 around 150 s. Comparing trial 4 in Figure 10, for which a smaller difference in task time was observed (SMD 0.85) suggests that coarse alignment with the correct grasp point is achieved on a similar timescale of ~ 20 s. For the haptic, a greater proportion of time is spent on fine alignment with the correct grasp point, while for the Franka, this time is predominantly spent on post-grasp manipulation.

For the module sorting task, significant improvements in the task completion time were observed, with an SMD of 2.31. Similarly to other tasks, improvements in coarse and fine alignment speed are observed, although the reduction in completion time is driven by time reductions over all stages of the task, including the “action” stage. Examining Figure 11 shows that after positional alignment, an increase in the force (up to ~ 25 N) occurs indicating the attempt

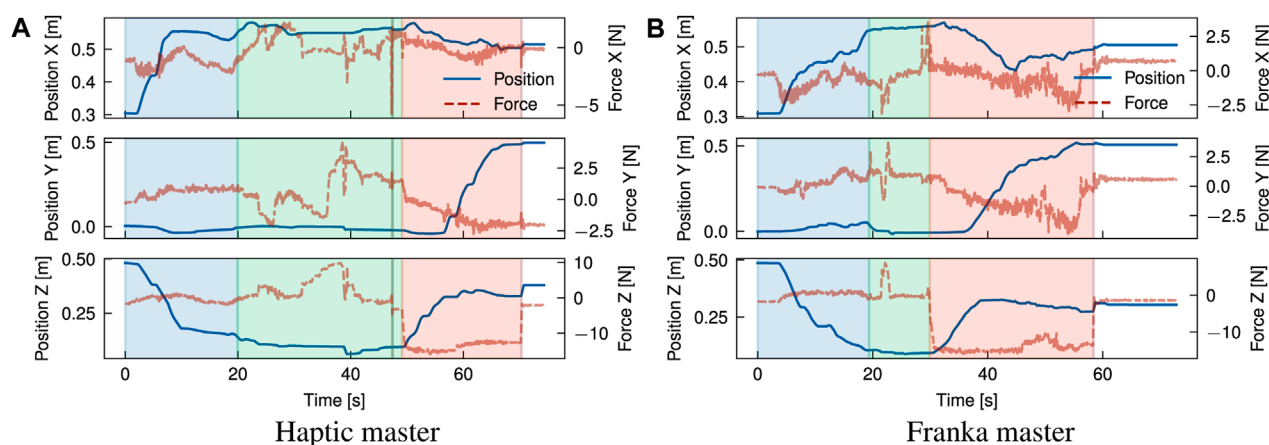


FIGURE 10

Overlay of position and estimated external forces at the end-effector for the cover removal task (trial 4) using a 2-finger gripper. An initial coarse alignment stage is marked by a rapid decrease in Z position, followed by a fine alignment guided by tactile feedback. Grasping and releasing of the cover is observed as application and release of a ~ 14 N force in the-Z direction. (A) Haptic master, (B) Franka master.

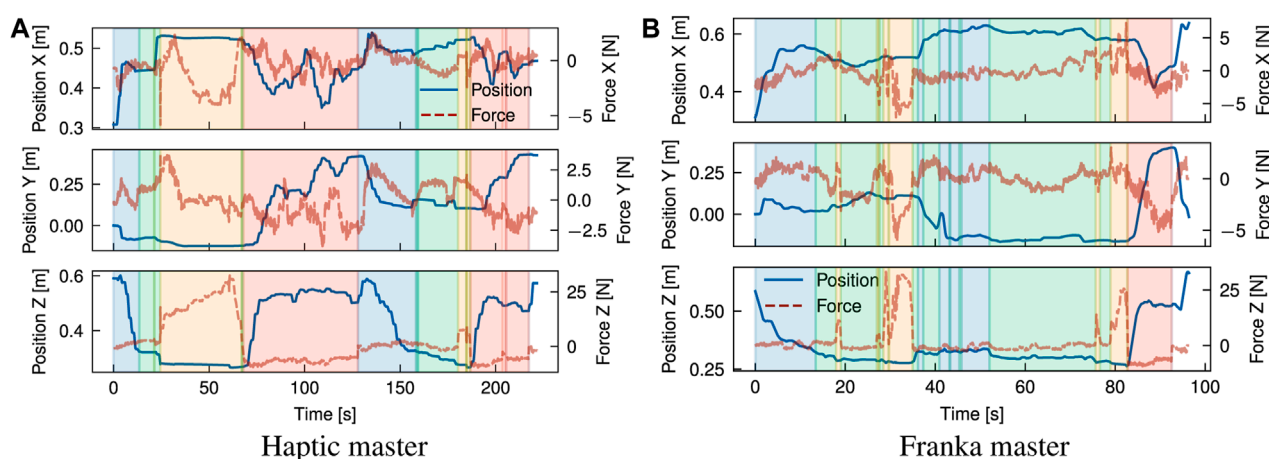


FIGURE 11

Overlay of position and estimated external forces at the end-effector for the sorting modules task (trial 5) using a vacuum suction gripper. An initial coarse alignment stage is marked by a rapid decrease in Z position, followed by a fine alignment guided by tactile feedback. Grasping and releasing of the module is observed as application and release of a ~ 8 N force in the-Z direction. (A) Haptic master, (B) Franka master.

to reach a successful suction grip at the module surface. As with the previous tasks, the magnitude of force feedback is largely consistent between both platforms; however, for the action phase, the operator exhibited a more slow and conservative approach with the haptic device, characterised by a slow ramp in the normal (+Z) force. Intuitively, the operator uses a mix of visual and tactile cues to identify engagement of the suction cups with the surface. This implies a combination of factors can be attributed to this behaviour; in the first instance, from similar factors contributing to faster positional alignment with the Franka, such as 1:1 position mapping, while in the second instance, the operator can identify the correct force threshold more intuitively, as with direct manipulation of the slave arm, through 1:1 force mapping. Next, the “action” stage is followed by the “place” stage (red region) where the module is removed from the scene. Contrasting the cover removal task,

a much shorter period of time is spent both proportionally and in absolute terms on post-grasp manipulation with the Franka. Comparison of the trial breakdown in Figures 10, 11 for these stages implies an effort-related component to the task performance, due to the higher load exerted by the cover (notable in the Z direction). Furthermore, the cover exhibits a more dispersed mass distribution in contrast with the module, where a majority of the weight is concentrated about the grasp point. Hence, positioning of the cover imposes a greater demand due to higher torque required to rotate the joints, which is largely abstracted for the Cartesian controlled haptic-Franka platform. This is corroborated by the more gradual and tortuous path seen in Figure 10. Therefore, realism of force feedback may be considered beneficial for the contact-rich stages of the task, but a telemanipulation framework incorporating 1:1 feedback of external torque should

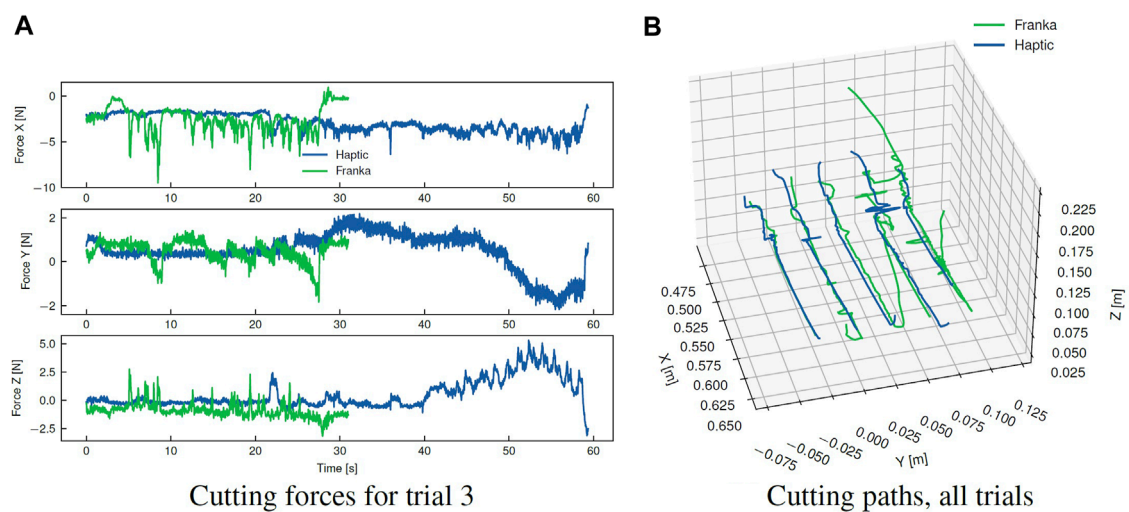


FIGURE 12

Comparison of cutting forces between each telerobotics platform for single trial, and overall paths adopted by each operator for all trials. (A) Cutting forces for trial 3, (B) Cutting paths, all trials.

be aware of detrimental effects on manipulations in free space.

For cutting, a small SMD of 0.41 suggests the effect on task completion time for the Franka platform is marginal relative to the trial-to-trial variation. Cutting is a complex task dependent on a wide range of material and tool-specific parameters, and a wide variation in completion times is observed between task trials. For the Franka, a greater proportion of time is spent performing the cutting task proper. This is on average achieved faster with the haptic device, however, a much greater proportion of time (55% vs. 32%) is required for initial alignment of the tool with the desired path. The fine alignment stage in particular is unique for the cutting task in that the operator must rely purely on visual alignment with the desired path; this further suggests the reduction in completion time for the presented tasks is in large part related to the expanded workspace and 1:1 *positional* mapping afforded with the Franka, rather than improvements in realism of the tactile feedback. However, during the cutting task, the realism of feedback is comparatively more important. This can be examined in Figure 12A, showing the force profile for the contact cutting task for both platforms. Notably, the force for the haptic trial steadily increases towards the end of the cutting operation, which can be attributed to drift of the operator setpoint away from the path in the transverse (Y) direction and an increase in the depth of cut (-Z). A similar effect can be observed in the transverse direction for the Franka, however, these are rapidly corrected during the course of the cutting task. This force feedback provides important cues to the operator that were otherwise ignored, or insufficient to prompt a response in the haptic case; similar force profiles were observed over the majority of the haptic trials.

The increased feedback capability of a setup with identical cobots has some notable disadvantages. From Figure 12B, it is clear that the cutting path adopted by the user in the Franka case study is subject to small variations along the length of the desired path. This is posited to be due to the lack of task space control, the

larger motions achievable with the Franka, and low stiffness of the human arm. Owing to these factors, the master arm responds readily to the force feedback created by the executed motions during the cutting process, causing small deflections from the operator's desired path. This can be observed from the comparatively more variable and discontinuous force profile opposing the feed direction (+X) in Figure 12A. For higher strength materials, this effect is expected to increase in significance due to an increase in the feed-rate dependent cutting forces. In the absence of remediating strategies, this consequently reduces the quality of cut and increases physical demand on the user. However, remediating strategies such as scaling the force feedback provided to the user, or applying a Cartesian impedance behaviour to the master arm to guide the user along the path would have the effect of distorting the feedback, thus reducing the realism of the interaction.

For the module stack disassembly case study comprising 4 modules, total disassembly times averaged 25.8 min and 14.7 min for the haptic and Franka platforms respectively, representing a 43% reduction in overall completion time when using two identical cobots. Considering module cover separation consisting of 4 linear cutting operations, it is estimated to require 8 min per module to expose the battery cells, reducing by 25% for the identical cobot case. Overall disassembly times demonstrate that while disassembly via telemanipulation is feasible, several efficiency improvements and assistive strategies are required to improve disassembly speed. However, there is large variation in disassembly time estimates across battery models and techno-economic analyses in literature. For example, an estimate of 12 s for manual unscrewing and 6 s for manual removal of free components across multiple battery designs (Lander et al., 2023) implies large reductions in efficiency versus manual disassembly. Another techno-economic analysis by Alfaro-Algaba and Ramirez (2020) for the Audi e-Tron hybrid estimates a total disassembly time of 30 min for pack-to-module disassembly. However, in Baazouzi et al. (2021), extraction of the battery modules from the

Smart ForFour was estimated at 10 min per module, and 135 min for disassembly of individual modules with 2 human personnel. Beyond module level, disassembly is hampered significantly by expensive cutting operations required to expose the individual battery cells. Broadly, this is corroborated across battery designs, chiefly due to the high level of design compartmentalisation and use of welds, glue and interference fit fasteners that are difficult to remove non-destructively (Baazouzi et al., 2021; Lander et al., 2023).

Limitations of the present work are that only objective measures of task performance are considered under the assumption of familiarity with the necessary tasks and teleoperation platforms (expert operators). However, to obtain a holistic comparison it is necessary to consider operators of different experience levels, as well as consideration of subjective performance measures, such as those established in NASA TLX that establish measures of cognitive load and effort for the sequence of disassembly tasks. Furthermore, while the semantic breakdown of the demonstrated tasks can provide insights into the most challenging parts of each task, the effect of variable autonomy on task performance with respect to each platform could still be explored. For example, differences in success rate for tasks such as bolt removal could be addressed by guiding the user towards suitable grasp points. Therefore, future works will consider the effect of shared control strategies on the completion time and success rate of the presented disassembly tasks.

5 Conclusion

This study demonstrates the telerobotic disassembly of a stack of modules from the Nissan Leaf 2011 battery pack. A comparative study between a master-slave setup consisting of a haptic device paired with a cobot, and two identical cobots examined the success rate and completion time for accomplishing unbolting, extraction of bolts, grasping and removal of the cover plate, sorting of modules with a suction gripper and contact cutting. While overall success rate was higher with the haptic device, a setup with identical paired cobots was found to significantly improve task completion times and time consistency across the entire set of disassembly tasks. This suggests that quality and realism of force feedback is comparatively less important for accomplishing precise manipulation tasks, but instead the 1:1 position mapping between master interface and slave and expanded workspace were main contributors to the efficiency improvements. While 1:1 mapping of torques between identical cobots was beneficial for grasping with a vacuum gripper and cutting, chiefly due to the enhanced realism of the interaction, providing strengthened tactile cues to the operator, this further had detrimental effects on operator physical effort and quality of cut. While a semantic breakdown of the presented disassembly tasks and objective performance measures provides insights into the key stages, feasibility and difficulty of each disassembly task with each platform, the effect of differing levels of operator expertise, subjective task load measures and variable autonomy on task performance was not considered. Further work will focus on evaluating the effect of variable autonomy on task

completion performance in the context of the presented disassembly case studies.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Author contributions

Conceptualization, AR; methodology, JH, AS, and AR; software, AS, JH, and AA; investigation, AS, JH, AA, and AR; validation, JH, AS, and AR; data curation, JH and AS; formal analysis, JH and AS; visualisation, JH and AS; writing—original draft, CA, JH, AS, and AR; writing—review and editing, AR, CA, and JH; resources, AR and RS; supervision AR and RS; funding acquisition, AR and RS; project administration, AR. All authors contributed to the article and approved the submitted version.

Funding

This work was supported in part by the project called “Research and Development of a Highly Automated and Safe Streamlined Process for Increase Lithium-ion Battery Repurposing and Recycling” (REBELION) under Grant 101104241 and in part by the United Kingdom Research and Innovation (UKRI) project “Reuse and Recycling of Lithium-Ion Batteries” (RELiB) under RELiB2 Grant FIRG005 and RELiB3 Grant FIRG057.

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2023.1179296/full#supplementary-material>

References

- Abi-Farraj, F., Pacchierotti, C., Arenz, O., Neumann, G., and Giordano, P. R. (2020). A haptic shared-control architecture for guided multi-target robotic grasping. *IEEE Trans. Haptics* 13, 270–285. doi:10.1109/TOH.2019.2913643
- Abi-Farraj, F., Pacchierotti, C., and Giordano, P. R. (2018). “User evaluation of a haptic-enabled shared-control approach for robotic telemanipulation,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 Oct. 2018 (IEEE), 1–9. doi:10.1109/IROS.2018.8594030
- Abi-Farraj, F., Pedemonte, N., and Robuffo Giordano, P. (2016). “A visual-based shared control architecture for remote telemanipulation,” in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea (South), 9–14 Oct. 2016 (IEEE), 4266–4273. doi:10.1109/IROS.2016.7759628
- Ai, N., Zheng, J., and Chen, W.-Q. (2019). U.S. end-of-life electric vehicle batteries: dynamic inventory modeling and spatial analysis for regional solutions. *Resour. Conservation Recycl.* 145, 208–219. doi:10.1016/j.resconrec.2019.01.021
- Alfaro-Algaba, M., and Ramirez, F. J. (2020). Techno-economic and environmental disassembly planning of lithium-ion electric vehicle battery packs for remanufacturing. *Resour. Conservation Recycl.* 154, 104461. doi:10.1016/j.resconrec.2019.104461
- Baazouzi, S., Rist, F. P., Weeber, M., and Birke, K. P. (2021). Optimization of disassembly strategies for electric vehicle batteries. *Batteries* 7, 74. doi:10.3390/batteries7040074
- Bernold, L. E. (2007). Control schemes for tele-robotic pipe installation. *Automation Constr.* 16, 518–524. doi:10.1016/j.autcon.2006.09.002
- Bolarinwa, J. O. (2022). *Enhancing Tele-operation-Investigating the effect of sensory feedback on performance*. Ph.D. thesis, Faculty of Engineering and Technology, University of the West of England.
- Borrás, J., Heudorfer, R., Rader, S., Kaiser, P., and Asfour, T. (2018). “The kit swiss knife gripper for disassembly tasks: a multi-functional gripper for bimanual manipulation with a single arm,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 Oct. 2018 (IEEE), 4590–4597. doi:10.1109/IROS.2018.8593567
- Choux, M., Marti Bigorra, E., and Tyapin, I. (2021). Task planner for robotic disassembly of electric vehicle battery pack. *Metals* 11, 387. doi:10.3390/met11030387
- Fan, J., Zheng, P., Li, S., and Wang, L. (2022). Two new species of *deutereulophus* schulz (hymenoptera, eulophidae) from China, with a key to Chinese species. *IEEE Trans. Automation Sci. Eng.* 1114, 1–9. doi:10.3897/zookeys.1114.86598
- Farhan, A. S., Bassiouny, A. M., Afif, Y. T., Gamil, A. A., Alsheikh, M. A., Kamal, A. M., et al. (2021). “Autonomous non-destructive assembly/disassembly of electronic components using a robotic arm,” in 2021 16th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, Egypt, 15–16 Dec. 2021 (IEEE), 1–7. doi:10.1109/ICCES54031.2021.9686083
- Gil, P., Pomares, J., Puente, S., Díaz Baca, C., Candelas Herias, F., and Medina, F. (2007). Flexible multi-sensorial system for automatic disassembly using cooperative robots. *Int. J. Comput. Integr. Manuf.* 20, 757–772. doi:10.1080/09511920601143169
- Giri, G. S., Maddahi, Y., and Zareinia, K. (2021). An application-based review of haptics technology. *Robotics* 10, 29. doi:10.3390/robotics10010029
- Gliesche, P., Krick, T., Pfingsthorn, M., Drolshagen, S., Kowalski, C., and Hein, A. (2020). Kinesthetic device vs. keyboard/mouse: a comparison in home care telemanipulation. *Front. Robotics AI* 7, 561015. doi:10.3389/frobt.2020.561015
- He, X., and Chen, Y. (2008). Six-degree-of-freedom haptic rendering in virtual teleoperation. *IEEE Trans. Instrum. Meas.* 57, 1866–1875. doi:10.1109/TIM.2008.919876
- Ju, C., and Son, H. I. (2022). Human-centered evaluation of shared teleoperation system for maintenance and repair tasks in nuclear power plants. *Int. J. Control, Automation Syst.* 20, 3418–3432. doi:10.1007/s12555-021-0770-0
- Kay, I., Farhad, S., Mahajan, A., Esmaeeli, R., and Hashemi, S. R. (2022). Robotic disassembly of electric vehicles’ battery modules for recycling. *Energies* 15, 4856. doi:10.3390/en15134856
- Lander, L., Tagnon, C., Nguyen-Tien, V., Kendrick, E., Elliott, R. J., Abbott, A. P., et al. (2023). Breaking it down: a techno-economic assessment of the impact of battery pack design on disassembly costs. *Appl. Energy* 331, 120437. doi:10.1016/j.apenergy.2022.120437
- Li, G., Caponetto, F., Del Bianco, E., Katsageorgiou, V., Sarakoglou, I., and Tsagarakis, N. G. (2020). Incomplete orientation mapping for teleoperation with one dof master-slave asymmetry. *IEEE Robotics Automation Lett.* 5, 5167–5174. doi:10.1109/LRA.2020.3006796
- Meng, K., Xu, G., Peng, X., Youcef-Toumi, K., and Li, J. (2022). Intelligent disassembly of electric-vehicle batteries: a forward-looking overview. *Resour. Conservation Recycl.* 182, 106207. doi:10.1016/j.resconrec.2022.106207
- Mizuno, N., Tazaki, Y., Hashimoto, T., and Yokokohji, Y. (2023). A comparative study of manipulator teleoperation methods for debris retrieval phase in nuclear power plant decommissioning. *Adv. Robot.* 0, 541–559. doi:10.1080/01691864.2023.2169588
- Nakanishi, J., Cory, R., Mistry, M., Peters, J., and Schaal, S. (2008). Operational space control: a theoretical and empirical comparison. *Int. J. Robotics Res.* 27, 737–757. doi:10.1177/0278364908091463
- Pehlken, A., Albach, S., and Vogt, T. (2017). Is there a resource constraint related to lithium ion batteries in cars? *Int. J. Life Cycle Assess.* 22, 40–53. doi:10.1007/s11367-015-0925-4
- Rastegarpanah, A., Gonzalez, H. C., and Stolkin, R. (2021). Semi-autonomous behaviour tree-based framework for sorting electric vehicle batteries components. *Robotics* 10, 82. doi:10.3390/robotics10020082
- Rietmann, N., Hügler, B., and Lieven, T. (2020). Forecasting the trajectory of electric vehicle sales and the consequences for worldwide co2 emissions. *J. Clean. Prod.* 261, 121038. doi:10.1016/j.jclepro.2020.121038
- Schmitt, J., Haupt, H., Kurrat, M., and Raatz, A. (2011). “Disassembly automation for lithium-ion battery systems using a flexible gripper,” in 2011 15th International Conference on Advanced Robotics (ICAR), Tallinn, Estonia, 20–23 June 2011 (IEEE), 291–297. doi:10.1109/ICAR.2011.6088599
- Shen, H., Pan, Y.-J., and Wan, L. (2021). “Teleoperated single-master-multiple-slave system for cooperative manipulations in task space,” in 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), Victoria, BC, Canada, 10–12 May 2021 (IEEE), 864–869. doi:10.1109/ICPS49255.2021.9468222
- Singh, J., Srinivasan, A. R., Neumann, G., and Kucukylmaz, A. (2020). Haptic-guided teleoperation of a 7-dof collaborative robot arm with an identical twin master. *IEEE Trans. Haptics* 13, 246–252. doi:10.1109/TOH.2020.2971485
- Tan, W. J., Chin, C. M. M., Garg, A., and Gao, L. (2021). A hybrid disassembly framework for disassembly of electric vehicle batteries. *Int. J. Energy Res.* 45, 8073–8082. doi:10.1002/er.6364
- Tang, J., Gao, Y., and Lam, T. L. (2022). *Learning to coordinate for a worker-station multi-robot system in planar coverage tasks*. arXiv doi:10.48550/arXiv.2208.02993
- Tang, Y., and Zhou, M. (2004). “Fuzzy-petri-net based disassembly planning considering human factors,” in 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583). vol. 5, (IEEE), 4195–4200. doi:10.1109/ICSMC.2004.1401189
- Thies, C., Kieckhäfer, K., Hoyer, C., and Spengler, T. S. (2018). *Economic assessment of the LithoRec process*. Cham: Springer International Publishing, 253–266. doi:10.1007/978-3-319-70572-9_15
- Tokatli, O., Das, P., Nath, R., Pangione, L., Altobelli, A., Burroughes, G., et al. (2021). Robot-assisted glovebox teleoperation for nuclear industry. *Robotics* 10, 85. doi:10.3390/robotics10030085
- Zhang, H., Yang, H., Wang, H., Wang, Z., Zhang, S., and Chen, M. (2023). “Autonomous electric vehicle battery disassembly based on neurosymbolic computing,” in *Intelligent systems and applications*. Editor K. Arai (Cham: Springer International Publishing), 443–457.
- Zhou, Y., Hu, H., Liu, Y., Lin, S.-W., and Ding, Z. (2019). A real-time and fully distributed approach to motion planning for multirobot systems. *IEEE Trans. Syst. Man, Cybern. Syst.* 49, 2636–2650. doi:10.1109/TSMC.2017.2750911
- Zorn, M., Ionescu, C., Klohs, D., Zühl, K., Kisseler, N., Daldrup, A., et al. (2022). An approach for automated disassembly of lithium-ion battery packs and high-quality recycling using computer vision, labeling, and material characterization. *Recycling* 7, 48. doi:10.3390/recycling7040048



OPEN ACCESS

EDITED BY

Alessandro Ridolfi,
University of Florence, Italy

REVIEWED BY

Alessandro Bucci,
University of Florence, Italy
Roohollah Barzamini,
Islamic Azad University Central Tehran
Branch, Iran
Maral Partovibakhsh,
Islamic Azad University System, Iran

*CORRESPONDENCE

Nargess Sadeghzadeh-Nokhodberiz,
✉ sadeghzadeh@qut.ac.ir
Allahyar Montazeri,
✉ a.montazeri@lancaster.ac.uk

RECEIVED 09 September 2023

ACCEPTED 11 December 2023

PUBLISHED 11 January 2024

CITATION

Malakouti-Khah H,
Sadeghzadeh-Nokhodberiz N and
Montazeri A (2024), Simultaneous
localization and mapping in a
multi-robot system in a dynamic
environment with unknown initial
correspondence.
Front. Robot. AI 10:1291672.
doi: 10.3389/frobt.2023.1291672

COPYRIGHT

© 2024 Malakouti-Khah,
Sadeghzadeh-Nokhodberiz and
Montazeri. This is an open-access article
distributed under the terms of the
[Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/)
(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.

Simultaneous localization and mapping in a multi-robot system in a dynamic environment with unknown initial correspondence

Hadiseh Malakouti-Khah¹,
Nargess Sadeghzadeh-Nokhodberiz^{1*} and Allahyar Montazeri^{2*}

¹Department of Control Engineering, Qom University of Technology, Qom, Iran, ²School of Engineering, Lancaster University, Lancaster, United Kingdom

A basic assumption in most approaches to simultaneous localization and mapping (SLAM) is the static nature of the environment. In recent years, some research has been devoted to the field of SLAM in dynamic environments. However, most of the studies conducted in this field have implemented SLAM by removing and filtering the moving landmarks. Moreover, the use of several robots in large, complex, and dynamic environments can significantly improve performance on the localization and mapping task, which has attracted many researchers to this problem more recently. In multi-robot SLAM, the robots can cooperate in a decentralized manner without the need for a central processing center to obtain their positions and a more precise map of the environment. In this article, a new decentralized approach is presented for multi-robot SLAM problems in dynamic environments with unknown initial correspondence. The proposed method applies a modified Fast-SLAM method, which implements SLAM in a decentralized manner by considering moving landmarks in the environment. Due to the unknown initial correspondence of the robots, a geographical approach is embedded in the proposed algorithm to align and merge their maps. Data association is also embedded in the algorithm; this is performed using the measurement predictions in the SLAM process of each robot. Finally, simulation results are provided to demonstrate the performance of the proposed method.

KEYWORDS

SLAM, multi-robot SLAM, Fast-SLAM, dynamic environments, moving landmarks, map merging

1 Introduction

Simultaneous localization and mapping (SLAM) is one of the most important developments in the field of robotics, enabling robots operating in GPS-denied environments to perceive their surroundings and localize themselves within the identified map. This is especially useful for industries such as underground mining and nuclear decommissioning, where mobile robots are required to explore and complete various tasks, such as maintenance, inspection, and transportation, in environments that are inaccessible and hazardous for humans. These missions are accomplished either using a single robot in isolation (Montazeri et al., 2021) or using a team of cooperative robots (Burrell et al., 2018). The robot or robots need to cope with the time-varying, restricted, uncertain, and

unstructured nature of the environment to achieve planning and execution of the necessary tasks. This in turn requires the design and development of advanced motion control and navigation algorithms, along with strong cognitive capabilities in order for the robot to perceive the surrounding environment effectively. The use of both single- and multi-robot platforms can be advantageous, depending on the specific application and environment.

SLAM-based navigation refers to the techniques by which the robots simultaneously localize themselves in an unknown environment where a map is required. This technique can be applied to single-robot (Debeunne and Vivet, 2020) or multi-robot (Almadhoun et al., 2019) systems, as well as in either static (Kretzschmar et al., 2010) or dynamic (Saputra et al., 2018) environments.

In a static environment, landmarks are fixed, which means that all objects in the environment are stationary and none are moving (Stachniss et al., 2016). A dynamic environment is an environment in which objects are moving, such as an environment that includes humans, robots, and other means of transportation. In a dynamic environment, the SLAM process becomes more complex due to continuous changes in the environment (such as moving objects, changing colors, and combinations of shadows), and this makes the problem more challenging, for example, by creating a mismatch between previous and new data (Li et al., 2021).

Several successful solutions have been presented for the single-robot and multi-robot SLAM problem in static environments (Saeedi et al., 2016; Lluvia et al., 2021). However, the SLAM problem in a dynamic environment is more complicated due to the requirements for simultaneous detection, classification, and tracking of moving landmarks. In recent years, single-robot and multi-robot SLAM problems in dynamic environments have attracted the attention of many researchers. In (Vidal et al., 2015; Badalkhani and Havangi, 2021), solutions for solving the single-robot and multi-robot SLAM problems in dynamic environments are presented. In (Vidal et al., 2015), a method is presented for SLAM in which only static (fixed) landmarks are considered and moving ones are omitted from the map. The paper uses an outlier filter and separates fixed and moving landmarks, which are included in a set of negligible signs; the authors only use fixed landmarks to implement their solution to the SLAM problem and do not consider moving landmarks.

In (Badalkhani and Havangi, 2021), a solution for the multi-robot SLAM problem in a dynamic environment is presented. The proposed SLAM-based method is developed using a Kalman filter (KF); it detects moving landmarks based on possible location constraints and expected landmark area, which enables detection and filtering of moving landmarks. The main goal of (Badalkhani and Havangi, 2021) is the development of a new method to identify the moving parts of the environment and remove them from the map.

There are also some pieces of research implementing the SLAM problem in a dynamic environment through object recognition or using image processing techniques. However, some of these methods, such as those presented in (Liu et al., 2019; Pancham et al., 2020; Chen et al., 2021; Theodorou et al., 2022; Guan et al., 2023), also aim to remove moving landmarks found in the dynamic environment.

As mentioned, researchers who have investigated the problem of SLAM in a dynamic environment using one or more robots have

mostly attempted to implement SLAM through the elimination of the moving landmarks, with only fixed ones being retained for the mapping procedure, which leads to the failure of the method when the environment is complex, with many moving objects, or in the absence of fixed objects.

In this paper, in contrast with previous work, a decentralized multi-robot modified Fast-SLAM-based algorithm is designed for use in a dynamic environment where all the landmarks are moving. It is worth mentioning that, in Fast-SLAM (Montemerlo et al., 2002), particle filtering (PF) (Godsill, 2019) is used to handle the non-linear kinematics of the robots via the Monte Carlo technique. Fast-SLAM is a filtering-based approach to SLAM decomposed into a robot localization problem, and a collection of landmark estimation problems conditioned on the robot pose estimate. This can be obtained using Bayes' rule combined with the statistical independence of landmark positions which leads to Rao Blackwellized particle filtering (RBPF) SLAM (Sadehghzadeh-Nokhodberiz et al., 2021).

In this study, we have simplified the multi-robot Fast-SLAM problem by considering two wheel-based robots in a dynamic environment with two moving landmarks with a known kinematic model. The proposed method can easily be extended to cases with more robots and landmarks. Each robot searches the environment and observes it with its onboard lidar sensor. Here, it is assumed that the kinematic models of the robots and landmarks are known, which makes the use of a modified Fast-SLAM method meaningful. This method is based on observer use of a particle filter and extended Kalman filter (EKF). However, several modifications of the normal Fast-SLAM method are required. The first modification is adding a prediction step to the EKF used for mapping using the kinematic model of the landmarks. Adding this step to the algorithm does not obviate the need for initialization, as the initial values of the landmarks' positions are assumed to be unknown. This prediction step is performed using the landmark's kinematic model at every sample time after the first visit to the landmark, even if the landmark has not been visited again by the lidar sensor. The second modification is that, in contrast with the normal approaches to data association with static landmarks, data association is performed based on the predicted measurement obtained from the predicted map. When data association is performed based on the predicted position of a landmark, it is less likely that a previously visited landmark will be wrongly diagnosed as a new one due to its movement. As the third modification, and due to the unknown initial correspondence, coordinate alignment and map-merging are added to the algorithm. In this step, the relative transformation matrix of the robots' inertial frames is computed when the robots meet each other, using a geographical approach as presented in (Zhou and Roumeliotis, 2006; Romero and Costa, 2010) to compute this transformation matrix, and the maps obtained in the coordinate system of each robot are then fused and merged.

In summary, the novelty of this paper can be considered to be the extension of the Fast-SLAM algorithm in several aspects, as follows:

1. Addition of a prediction step to the EKF used for mapping, based on the kinematic model of the landmarks after the first visit to each landmark and at every sample time.
2. Data association based on the predicted measurements obtained from the predicted map.

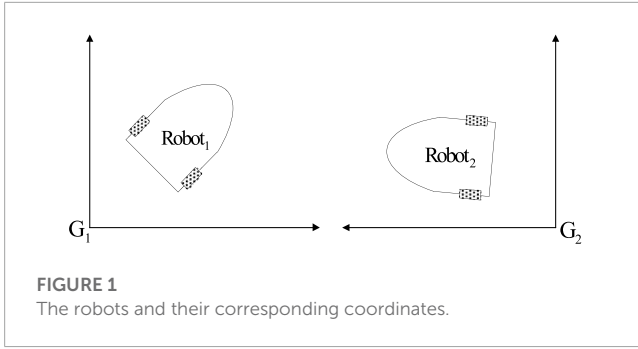


FIGURE 1
The robots and their corresponding coordinates.

3. Addition of coordinate alignment and map-merging to the algorithm when the robots meet each other.

The paper is organized as follows. In [Section 2](#), a system overview is presented, including a kinematic model of the robots and landmarks as well as the lidar measurement model. The modified Fast-SLAM algorithm for the dynamic environment is proposed in [Section 3](#). [Section 4](#) describes the coordinate alignment and map-merging problem. Simulation results are presented in [Section 5](#), and conclusions are provided in [Section 6](#).

2 System overview

In this paper, for the sake of simplicity, two mobile robots and two dynamic landmarks have been considered. As an example, differential drive mobile robots such as the Pioneer 3-DX ([Zaman et al., 2011](#)) moving in a dynamic environment were selected for the study. The robots are equipped with lidar and IMU sensors. In this section, the kinematic models of the robots and landmarks and the measurement model of the lidar are assumed to be known.

It is worth mentioning that IMUs can provide information on linear acceleration and angular velocity. However, linear speed can be obtained through the integration of linear acceleration.

2.1 Kinematic model of the robots

The kinematic model for the two differential drive Pioneer 3-DX mobile robots can be written as follows:

$$\begin{bmatrix} x_{r_i}(k+1) \\ y_{r_i}(k+1) \\ \varphi_{r_i}(k+1) \end{bmatrix} = \begin{bmatrix} x_{r_i}(k) + v(k) \cos(\varphi_{r_i}(k) + \omega(k)\Delta_t) \\ y_{r_i}(k) + v(k) \sin(\varphi_{r_i}(k) + \omega(k)\Delta_t) \\ \varphi_{r_i}(k) + \omega(k)\Delta_t \end{bmatrix} + \mathbf{v}_{r_i}(k), \quad (1)$$

where x_{r_i} and y_{r_i} are the i^{th} positions of the robot on the x and y axes and φ_{r_i} is its corresponding direction relative to the x -axis for $i = 1, 2$. The coordinate system of the first robot is G_1 and that of the second robot is G_2 , as depicted in [Figure 1](#). In [\(1\)](#), $\mathbf{v}_{r_i}(k)$ is zero-mean non-Gaussian process noise with a known probability density function (PDF). Additionally, Δ_t is the sampling time of the process and the variables $v(k)$ and $\omega(k)$ are the linear and the angular velocities of the robot, which are assumed to be known.

Eq. [1](#) can be rewritten in the general non-linear state space form as follows:

$$\mathbf{x}_{r_i}(k+1) = \mathbf{f}_{r_i}(\mathbf{x}_{r_i}(k), \mathbf{u}_{r_i}(k)) + \mathbf{v}_{r_i}(k), \quad (2)$$

where $\mathbf{x}_{r_i}(k) = [x_{r_i}(k) \ y_{r_i}(k) \ \varphi_{r_i}(k)]^T$, $\mathbf{u}_{r_i}(k) = [\omega(k) \ v(k)]^T$, and $\mathbf{f}_{r_i}(\mathbf{x}_{r_i}(k), \mathbf{u}_{r_i}(k))$ is the kinematic model presented in [\(1\)](#).

2.2 Kinematic model of the landmarks

In this study, both landmarks are considered to be moving, with the following kinematic model:

$$\begin{bmatrix} x_{L_j,r_i}(k+1) \\ y_{L_j,r_i}(k+1) \\ \varphi_{L_j,r_i}(k+1) \end{bmatrix} = \begin{bmatrix} x_{L_j,r_i}(k) + v_{L_j}(k) \cos(\varphi_{L_j,r_i}(k) + \omega_{L_j}(k)\Delta_t) \\ y_{L_j,r_i}(k) + v_{L_j}(k) \sin(\varphi_{L_j,r_i}(k) + \omega_{L_j}(k)\Delta_t) \\ \varphi_{L_j,r_i}(k) + \omega_{L_j}(k)\Delta_t \end{bmatrix} + \mathbf{v}_{L_j,r_i}(k), \quad (3)$$

where x_{L_j,r_i} and y_{L_j,r_i} are the j^{th} positions of the landmark on the x and y axes, respectively, and φ_{L_j,r_i} is its corresponding direction with respect to the x -axis for $j = 1, 2$, presented in G_i for $i = 1, 2$. Variables $v_{L_j}(k)$ and $\omega_{L_j}(k)$ are the linear velocity and angular velocity of landmark L_j , respectively; Δ_t is the sampling time of the process, and k refers to the sample number. It is assumed that $v_{L_j}(k)$ and $\omega_{L_j}(k)$ are known parameters. Moreover, $\mathbf{v}_{L_j,r_i}(k)$ is a zero-mean Gaussian noise vector: that is, $\mathbf{v}_{L_j,r_i}(k) \sim N(0, \mathbf{R}_{L_j,r_i})$, where $N(\cdot, \cdot)$ refers to the Gaussian PDF.

By writing [\(3\)](#) in the general non-linear state space model, we obtain

$$\mathbf{x}_{L_j,r_i}(k+1) = \mathbf{f}_{L_j,r_i}(\mathbf{x}_{L_j,r_i}(k), \mathbf{u}_{L_j}(k)) + \mathbf{v}_{L_j,r_i}(k), \quad (4)$$

where $\mathbf{x}_{L_j,r_i}(k) = [x_{L_j,r_i}(k) \ y_{L_j,r_i}(k) \ \varphi_{L_j,r_i}(k)]^T$, $\mathbf{u}_{L_j}(k) = [\omega_{L_j}(k) \ v_{L_j}(k)]^T$, and $\mathbf{f}_{L_j,r_i}(\mathbf{x}_{L_j,r_i}(k), \mathbf{u}_{L_j}(k))$ is the kinematic model presented in [\(4\)](#).

2.3 Lidar measurement model

Lidar sensor output includes the distance and angle of each robot relative to the observed landmarks. Each observation by each robot of the landmarks that are present can be expressed by the following observation model:

$$\begin{aligned} \mathbf{z}_{L_j,r_i}(k) &= \begin{bmatrix} r_{L_j,r_i}(k) \\ \varphi_{L_j,r_i}(k) \end{bmatrix} + \mathbf{v}_{ij}, \\ r_{L_j,r_i}(k) &= \sqrt{(x_{r_i}(k) - x_{L_j}(k))^2 + (y_{r_i}(k) - y_{L_j}(k))^2}, \\ \varphi_{L_j,r_i}(k) &= \arctan 2 \left(\frac{y_{r_i}(k) - y_{L_j}(k)}{(x_{r_i}(k) - x_{L_j}(k))} \right), \end{aligned} \quad (5)$$

where \mathbf{z}_{L_j,r_i} represents the i^{th} robot observation vector of the j^{th} landmark in the environment, r_{ij} represents the distance of the robot from the observed landmark, and φ_{ij} is the corresponding angle of

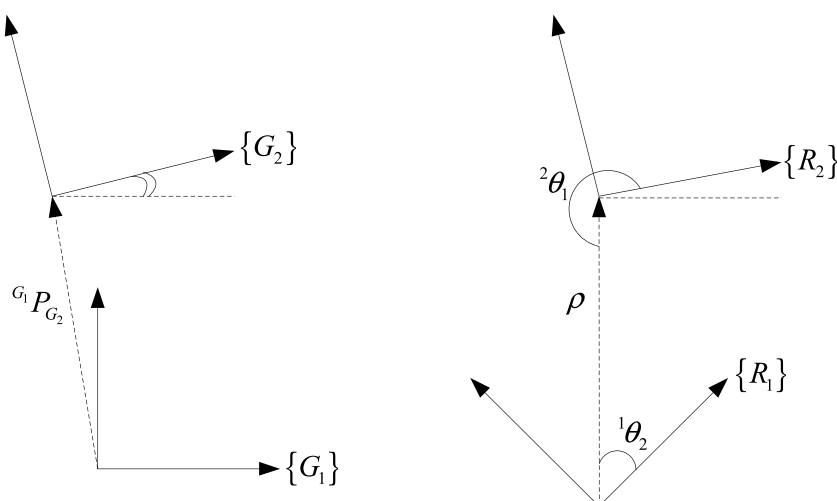


FIGURE 2

The relationships between different coordinates.

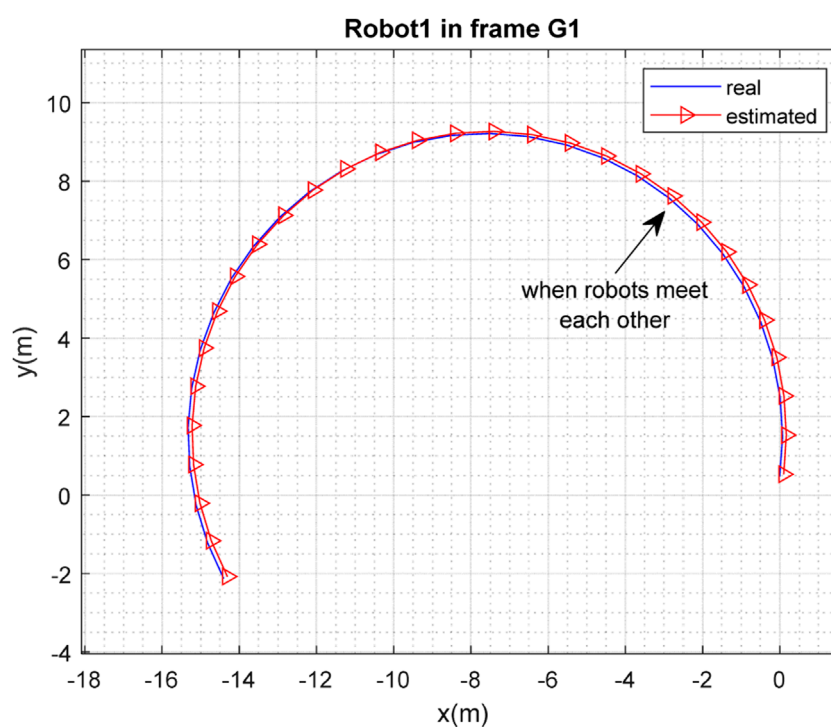


FIGURE 3

Real and estimated position of robot R_1 in the frame G_1 .

the robot relative to the landmark. Additionally, \mathbf{v}_{ij} is a zero-mean Gaussian measurement noise vector with the covariance matrix of $\mathbf{R}_{z_{ij}}$. Moreover, (5) can be rewritten in the general non-linear measurement model as follows:

$$\mathbf{z}_{L_j, r_i}(k) = \mathbf{h}(\mathbf{x}_{r_i}(k), \mathbf{x}_{L_j}(k)) + \mathbf{v}_{ij}. \quad (6)$$

3 Modified Fast-SLAM with dynamic landmarks

Robots R_1 and R_2 explore the environment, with each of them performing the SLAM process from the corresponding frames of G_1 and G_2 , respectively; and an independent map of the environment is thereby created by each of them. Information on

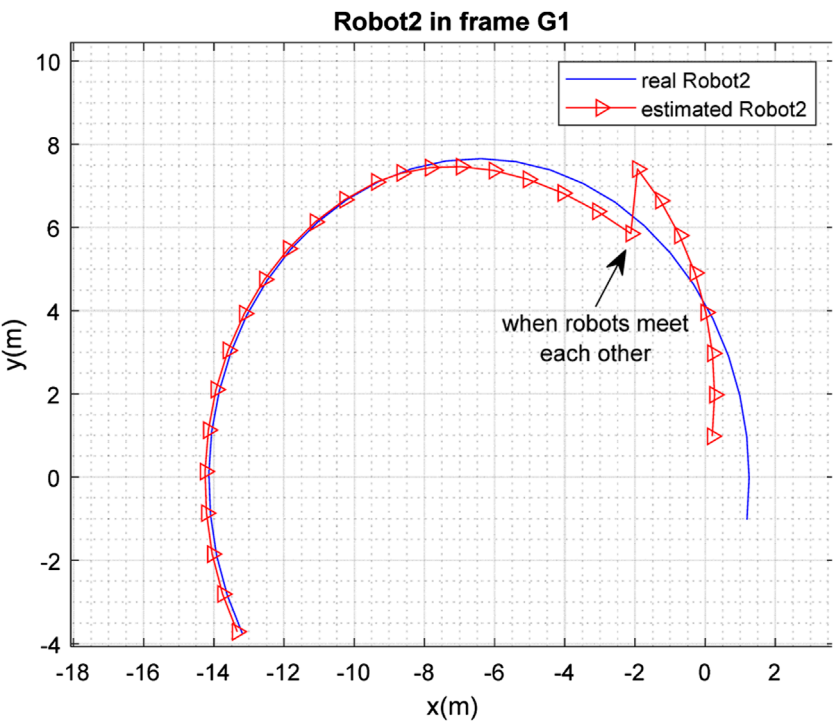


FIGURE 4
Real and estimated position of robot R_2 in the frame G_1 .

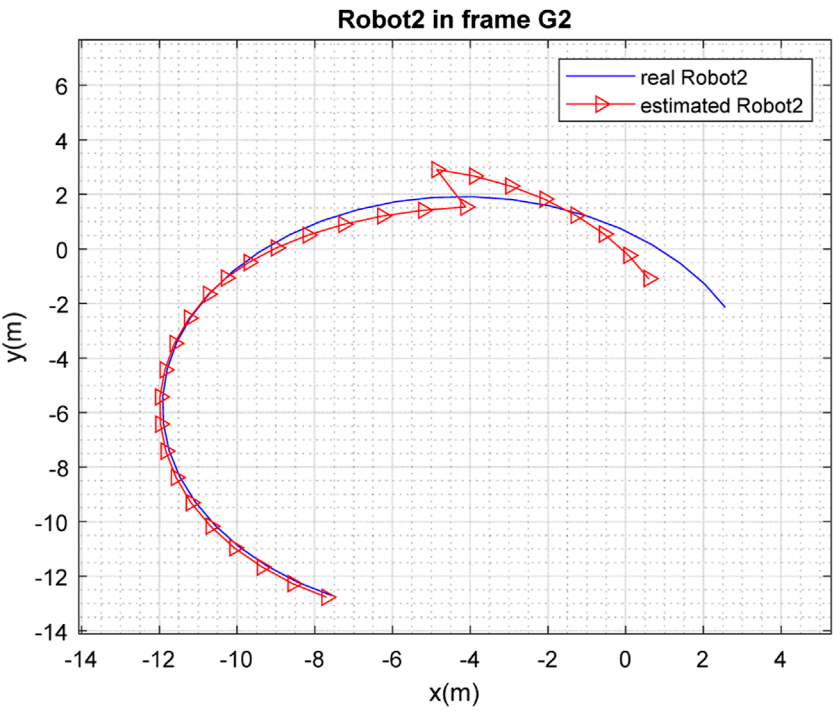


FIGURE 5
Real and estimated position of robot R_2 in the frame G_2 .

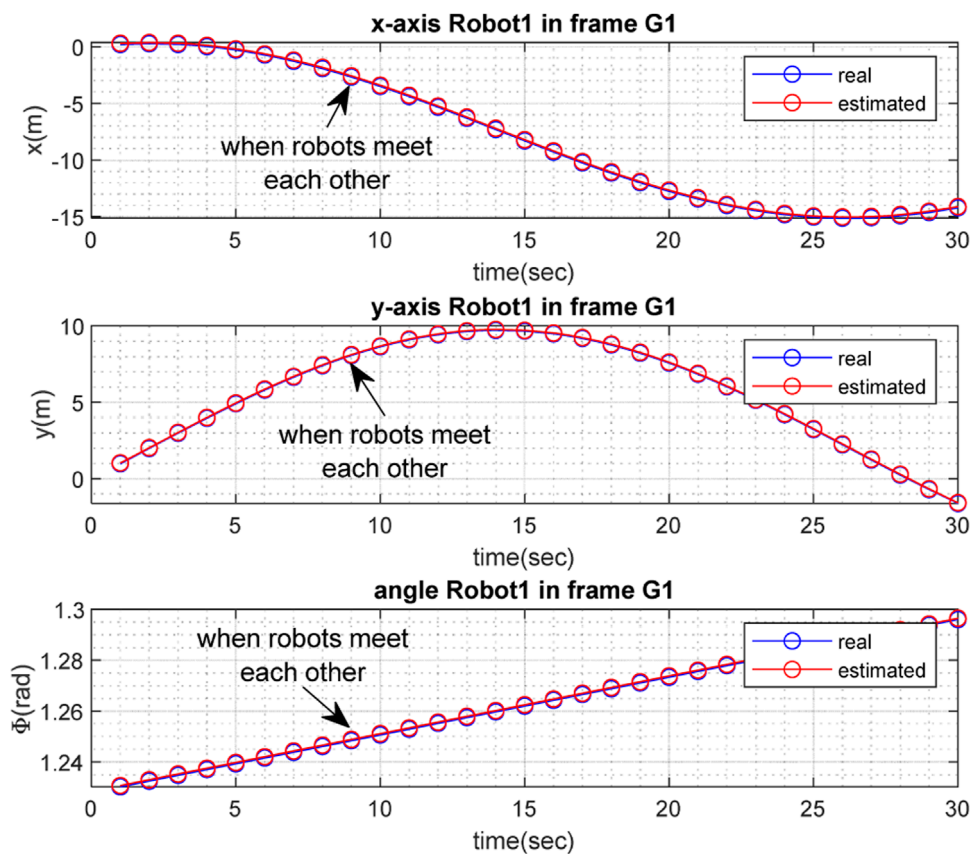


FIGURE 6

Real and estimated values of x_{r_1} , y_{r_1} , and ϕ_{r_1} in G_1 .

the distances and angles of the robots relative to the observed landmarks is provided by the lidar sensor in each robot. Each robot then uses a modified Fast-SLAM algorithm. In the proposed modified Fast-SLAM, Rao Blackwellized particle filtering (RBPF) is employed to estimate both vehicle trajectories and landmark positions, in which each landmark is estimated with EKF and PFs are employed to generate particles only used for trajectory estimation. In the remainder of this section, this process is explained, with the application of a modification in EKF due to the presence of moving landmarks with known kinematics. The general steps of Fast-SLAM consist of particle filtering with embedded extended Kalman filtering for the mapping; the data association process; and the map-merging procedure. These steps, with the necessary modifications, are explained for our problem in the following subsections.

3.1 Particle generation

An important and initial step in the PF is particle generation. To this end, particles are generated using the known PDF of $\mathbf{v}_{r_i}(k)$ through Monte Carlo simulation, where particles of $\mathbf{v}_{r_i}^l(k-1)$ for $l = 1, \dots, N$ are generated and replaced in the

motion model of the i^{th} robot (1), and therefore $x_{r_i}^l(k)$, $y_{r_i}^l(k)$, and $\phi_{r_i}^l(k)$ for $l = 1, \dots, N$ and $i = 1, 2$ are generated. It is worth mentioning that N refers to the number of particles. This process can be formulated in the following general form according to (2):

$$\mathbf{x}_{r_i}^l(k) = \mathbf{f}_{r_i}(\mathbf{x}_{r_i}^l(k-1), \mathbf{u}_{r_i}(k-1)) + \mathbf{v}_{r_i}^l(k-1). \quad (7)$$

3.2 Mapping

Each robot separately performs mapping after the particle generation process using EKF, as explained in the below subsections. After the first meeting of the robots and coordinate transformation, the maps are merged, as explained in detail in subsequent subsections.

3.2.1 EKF for mapping

- Initialization

After the particle generation process as explained in the previous subsection, when the j^{th} landmark is visited for the first time, initialization is performed using the inverse sensor model for

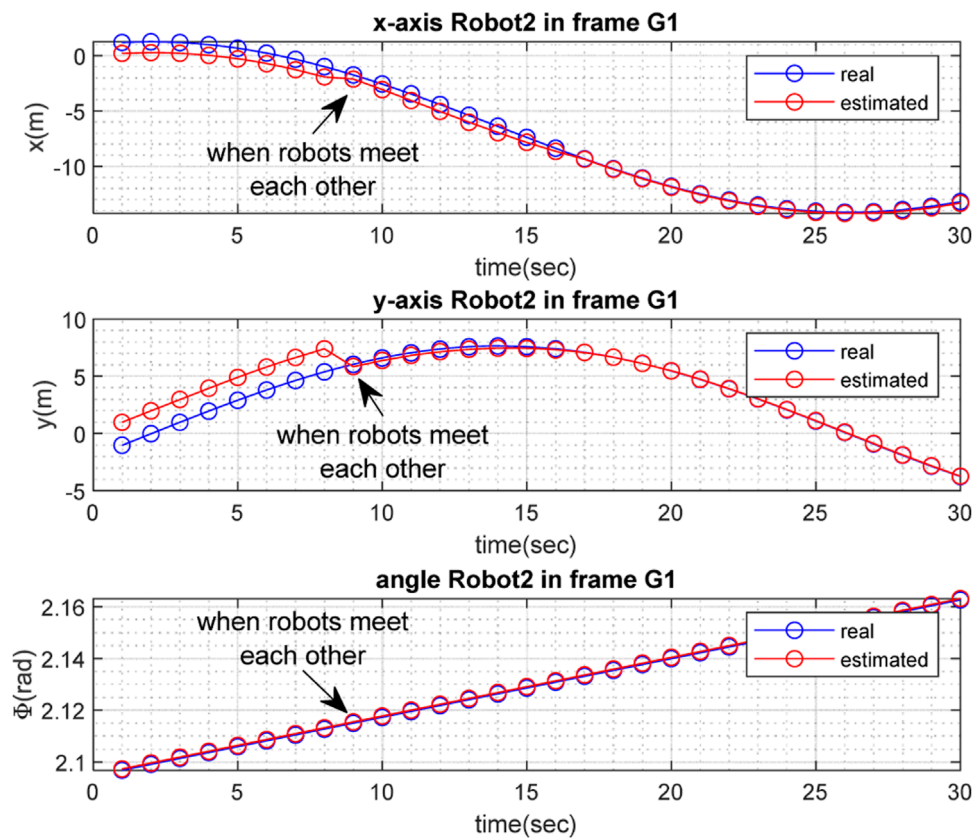


FIGURE 7

Real and estimated values of x_{r_2} , y_{r_2} , and ϕ_{r_2} in G_1 .

each particle, as follows, if it has been visited at sample instant k (neglecting the measurement noise):

$$\begin{aligned}\hat{x}_{L_j, r_i}^{l+}(k) &= x_{r_i}^l(k) + r_{L_j, r_i}(k) \cos(\varphi_{L_j, r_i}(k) + \varphi_{r_i}^l(k)), \\ \hat{y}_{L_j, r_i}^{l+}(k) &= y_{r_i}^l(k) + r_{L_j, r_i}(k) \sin(\varphi_{L_j, r_i}(k) + \varphi_{r_i}^l(k)), \\ \hat{\phi}_{L_j, r_i}^{l+}(k) &= \varphi_{L_j, r_i}(k) + \varphi_{r_i}^l(k),\end{aligned}\quad (8)$$

Where $\hat{x}_{L_j, r_i}^{l+}(k)$, $\hat{y}_{L_j, r_i}^{l+}(k)$, and $\hat{\phi}_{L_j, r_i}^{l+}(k)$ generally refer to the posterior estimates. Although the posterior estimates are obtained in the update step of the Bayesian filtering procedure, as the initial values will be used for prior estimates in the prediction step, a similar notation has been used. (8) can be rewritten in the following general form using (6):

$$\hat{\mathbf{x}}_{L_j, r_i}^{l+}(k) = \mathbf{h}^{-1}(\mathbf{z}_{L_j, r_i}(k), \mathbf{x}_{r_i}^l(k)). \quad (9)$$

Moreover, let predicted covariance matrix $\mathbf{P}_{L_j, r_i}^{l+} = K\mathbf{I}_3$, where K is a small real number and \mathbf{I}_3 is an identity matrix.

• Prediction step

After the first visit to the j^{th} landmark, prior estimates (predicted estimates) $\hat{x}_{L_j, r_i}^{l-}(k)$, $\hat{y}_{L_j, r_i}^{l-}(k)$, and $\hat{\phi}_{L_j, r_i}^{l-}(k)$ are computed

based on the landmark kinematics (3) at each time step for each particle l . Additionally, it is necessary to compute the predicted covariance matrix. This step is the first modification of our method compared with normal Fast-SLAM with static landmarks, in which no kinematics exist for the landmarks. In other words:

$$\hat{\mathbf{x}}_{L_j, r_i}^{l-}(k) = \mathbf{f}_{L_j, r_i}(\hat{\mathbf{x}}_{L_j, r_i}^{l+}(k-1), \mathbf{u}_{L_j}(k-1)). \quad (10)$$

$$\mathbf{P}_{L_j, r_i}^{l-}(k) = \mathbf{A}_{L_j, r_i}^{l+} \mathbf{P}_{L_j, r_i}^{l+}(k-1) (\mathbf{A}_{L_j, r_i}^{l+})^T + \mathbf{R}_{L_j, r_i}, \quad (11)$$

where in (11) $\mathbf{P}_{L_j, r_i}^{l-}$ is the predicted covariance matrix and $\mathbf{P}_{L_j, r_i}^{l+}$ is the updated one. Moreover:

$$\mathbf{A}_{L_j, r_i}^{l+} = \left. \frac{\partial \mathbf{f}_{L_j, r_i}(\mathbf{x}_{L_j, r_i}(k-1), \mathbf{u}_{L_j}(k-1))}{\partial \mathbf{x}_{L_j, r_i}(k-1)} \right|_{\hat{\mathbf{x}}_{L_j, r_i}^{l+}(k-1), \mathbf{u}_{L_j}(k-1)}. \quad (12)$$

• Data association

Data association is one of the important issues for SLAM in dynamic environments. For this purpose, in this paper, the measurement prediction is simply computed as follows:

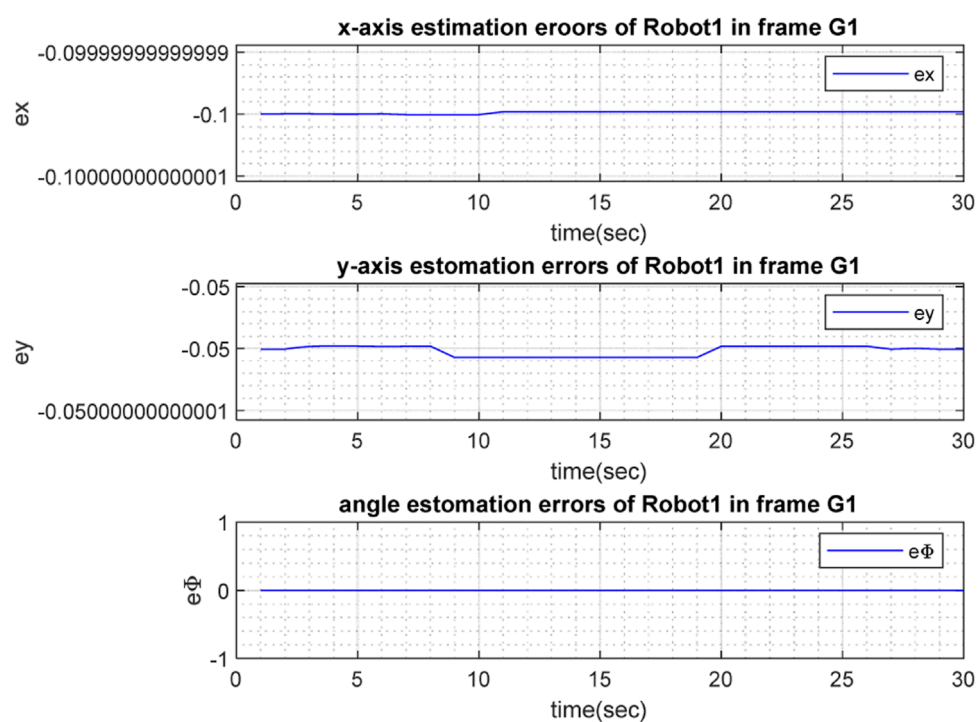


FIGURE 8
Estimation errors $e_{x_{r1}}$, $e_{y_{r1}}$, and $e_{\Phi_{r1}}$ in G_1 .

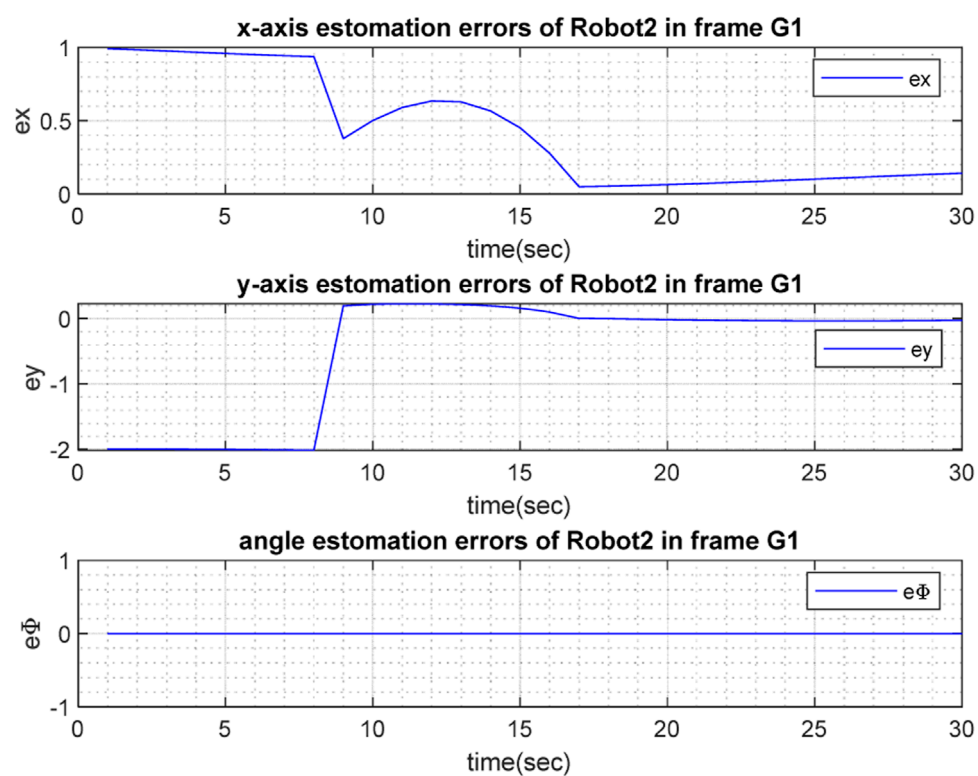


FIGURE 9
Estimation errors $e_{x_{r2}}$, $e_{y_{r2}}$, and $e_{\Phi_{r2}}$ in G_1 .

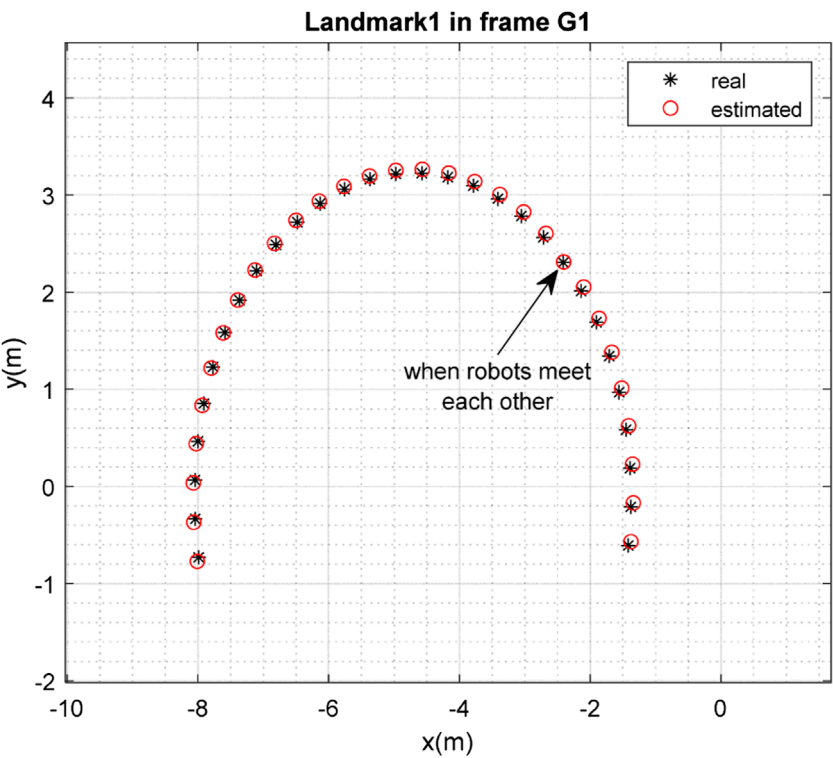


FIGURE 10
Real position of landmark L_1 and position estimated by robot R_1 in the frame G_1 .

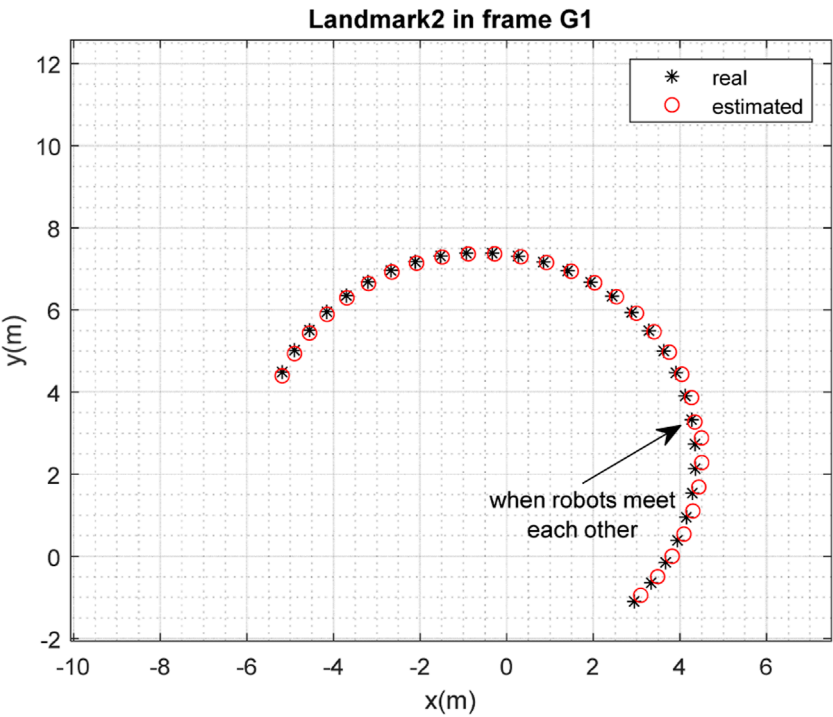
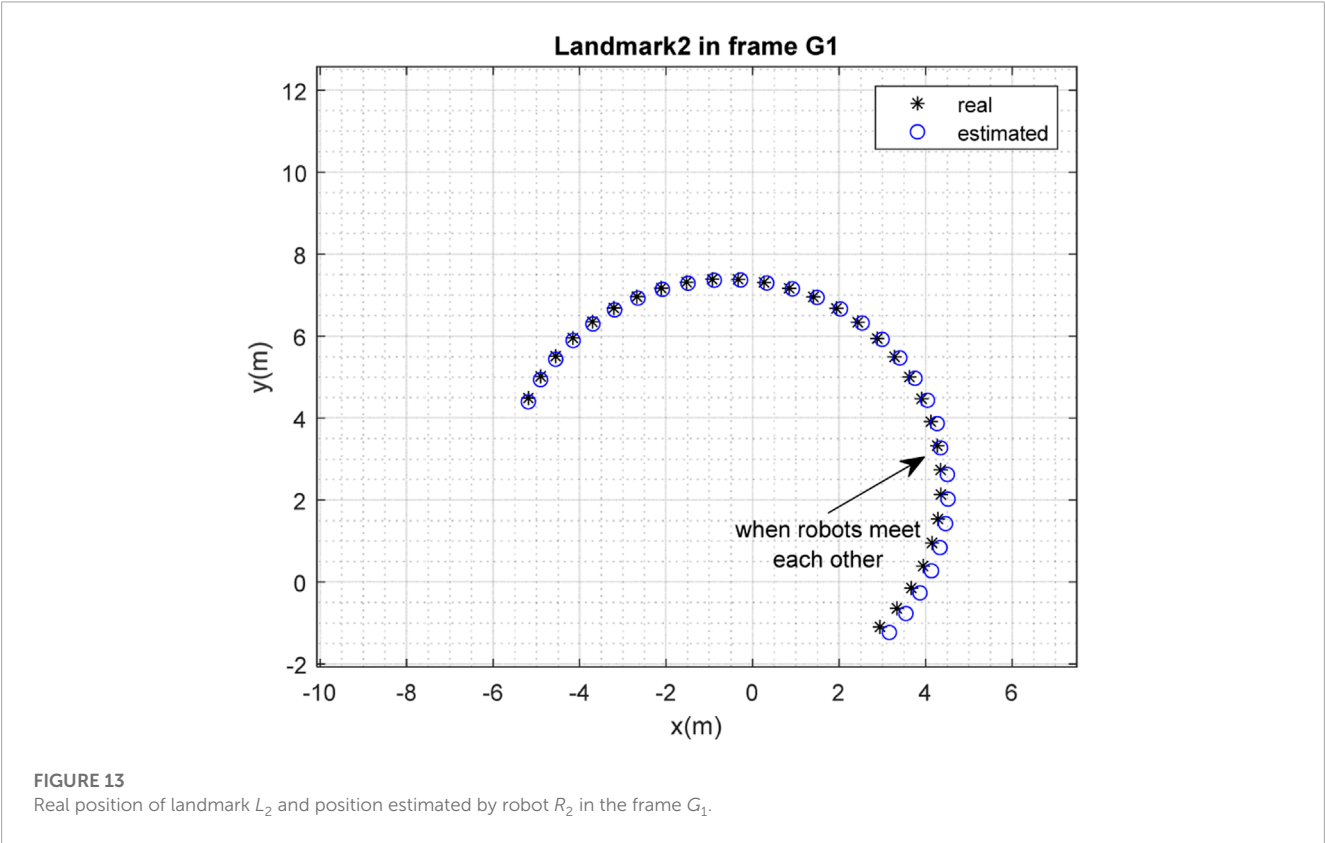
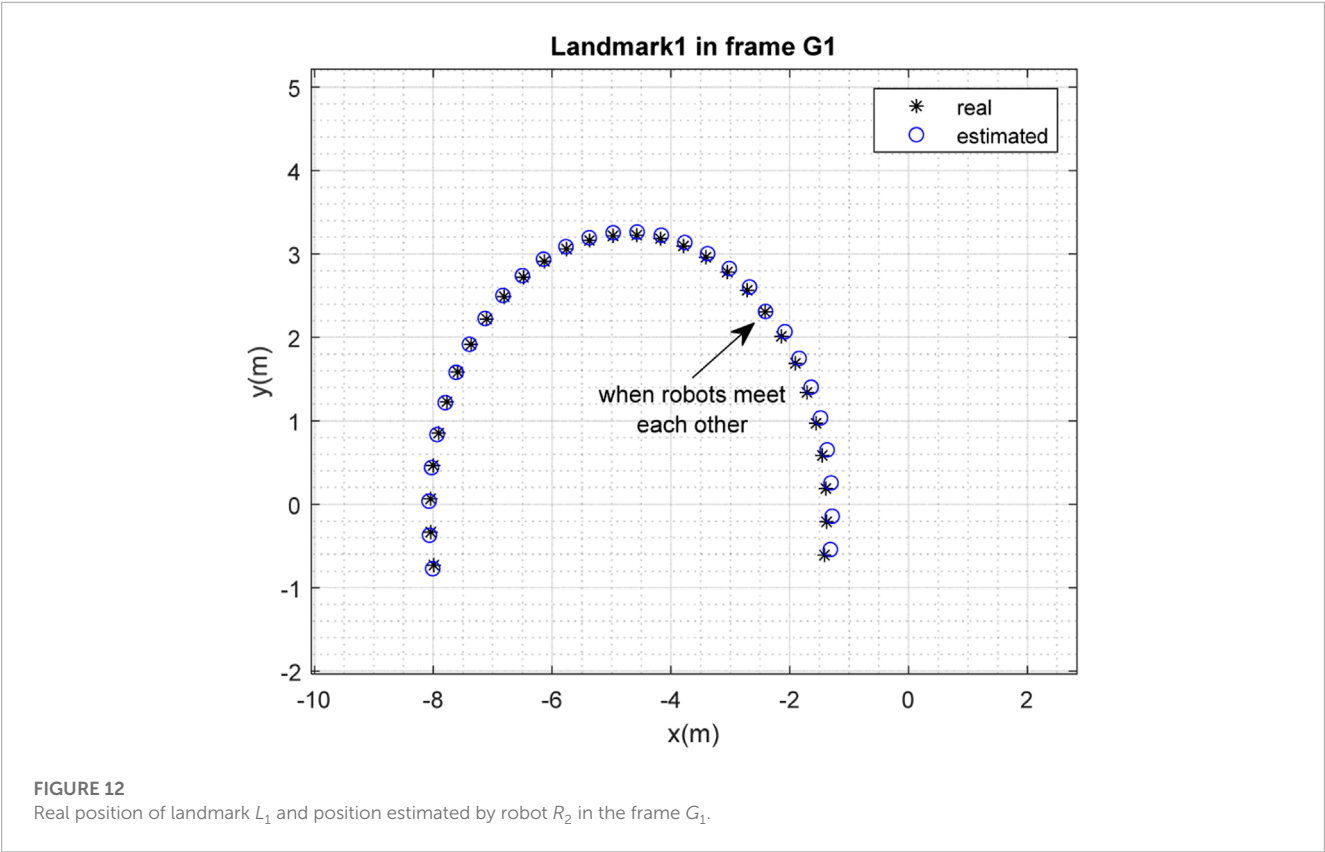


FIGURE 11
Real position of landmark L_2 and position estimated by robot R_1 in the frame G_1 .



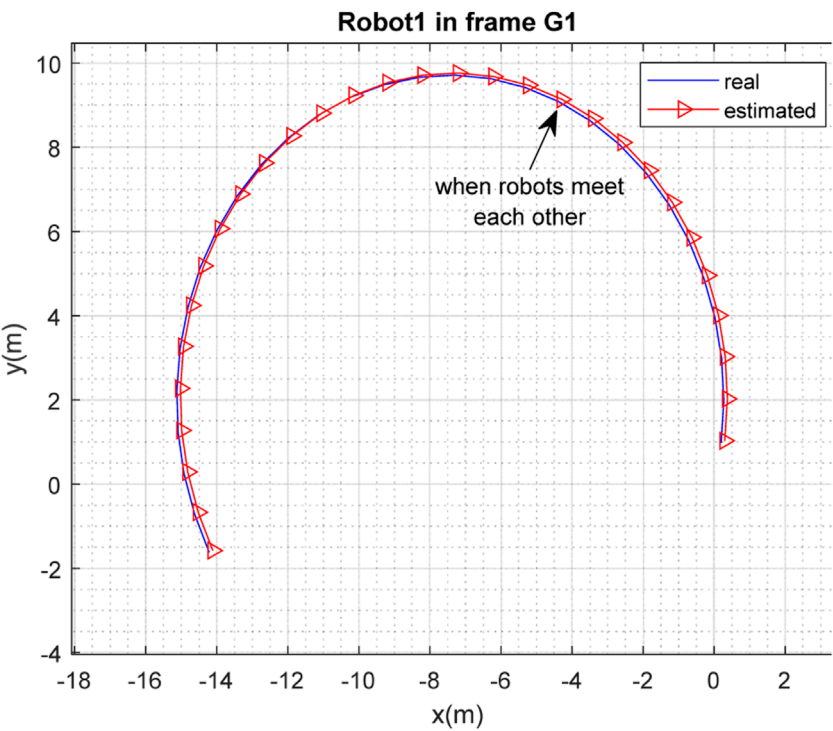


FIGURE 14
Real and estimated position of robot R_1 in the frame G_1 (with different initial conditions).

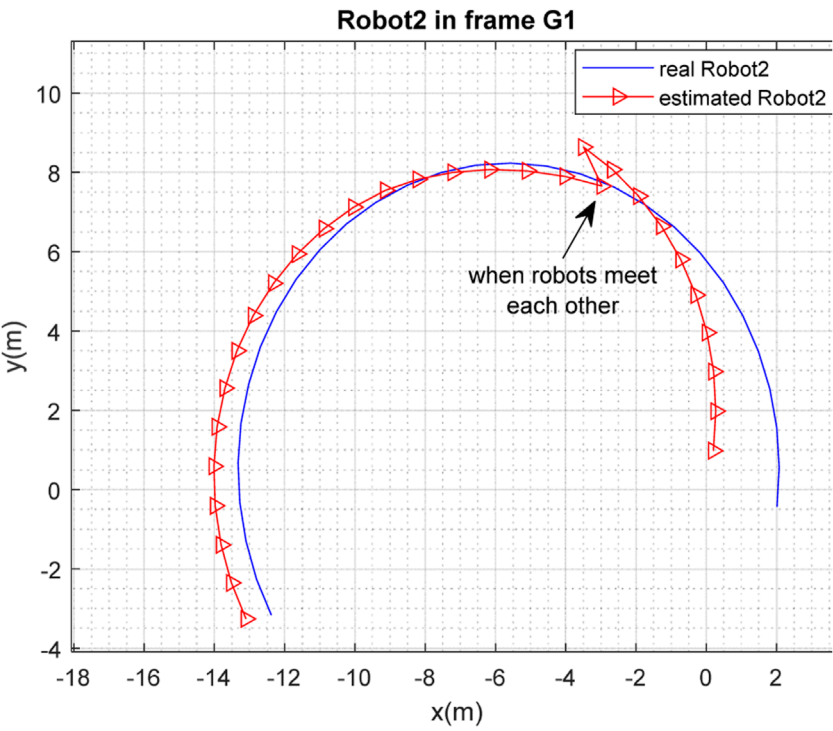


FIGURE 15
Real and estimated position of robot R_2 in the frame G_1 (with different initial conditions).

$$\begin{aligned}\hat{\mathbf{z}}_{L_j, r_i}^{i-}(k) &= \begin{bmatrix} \hat{r}_{L_j, r_i}^{i-}(k) \\ \hat{\varphi}_{L_j, r_i}^{i-}(k) \end{bmatrix}, j = 1, 2 \\ \hat{r}_{L_j, r_i}^{i-}(k) &= \sqrt{\left(\mathbf{x}_{r_i}^l(k) - \hat{\mathbf{x}}_{L_j, r_i}^{l-}(k)\right)^2 + \left(\mathbf{y}_{r_i}^l(k) - \hat{y}_{L_j, r_i}^{l-}(k)\right)^2}, \\ \hat{\varphi}_{L_j, r_i}^{i-}(k) &= \arctan 2 \left(\frac{y_{r_i}^l(k) - \hat{y}_{L_j, r_i}^{l-}(k)}{\left(\mathbf{x}_{r_i}^l(k) - \hat{\mathbf{x}}_{L_j, r_i}^{l-}(k)\right)} \right),\end{aligned}\quad (13)$$

where $\hat{r}_{L_j, r_i}^{i-}(k)$ represents the estimated distance of the i^{th} robot from the j^{th} landmark, and $\hat{\varphi}_{L_j, r_i}^{i-}(k)$ represents the estimated angle of the i^{th} robot relative to j^{th} landmark. Next, the Euclidean distance between the real observations, $\mathbf{z}_{L_j, r_i}(k)$, and the estimated ones is simply calculated:

$$r_{d_{ij}}^j = \left\| \mathbf{z}_{L_j, r_i}(k) - \hat{\mathbf{z}}_{L_j, r_i}^{l-}(k) \right\|. \quad (14)$$

For data association, the condition $r_{d_{ij}}^j < \mu$ for $j = 1, 2$ is first checked, where μ is a predefined threshold. Subsequently, if this condition is established for both observed landmarks, the smaller value is selected.

• Update step

If the landmark j is observed, this step is performed in a similar way to normal Fast-SLAM, which consists of gain computation, state update, and covariance update, as follows:

$$\begin{aligned}\mathbf{S}_{L_j, r_i}^l(k) &= \mathbf{H}_{L_j, r_i}^{l-}(k) \mathbf{P}_{L_j, r_i}^{l-}(k) \left(\mathbf{H}_{L_j, r_i}^{l-}(k) \right)^T + \mathbf{R}_{z_{ij}}, \\ \mathbf{K}_{L_j, r_i}^l(k) &= \mathbf{P}_{L_j, r_i}^{l-}(k) \left(\mathbf{H}_{L_j, r_i}^{l-}(k) \right)^T \left(\mathbf{S}_{L_j, r_i}^l(k) \right)^{-1}, \\ \hat{\mathbf{x}}_{L_j, r_i}^{l+}(k) &= \hat{\mathbf{x}}_{L_j, r_i}^{l-}(k) + \mathbf{K}_{L_j, r_i}^l(k) \left(\mathbf{z}_{L_j, r_i}(k) - \hat{\mathbf{z}}_{L_j, r_i}^{l-}(k) \right), \\ \mathbf{P}_{L_j, r_i}^{l+}(k) &= \left(\mathbf{I} - \mathbf{K}_{L_j, r_i}^l(k) \mathbf{H}_{L_j, r_i}^{l-}(k) \right) \mathbf{P}_{L_j, r_i}^{l-}(k),\end{aligned}\quad (15)$$

where \mathbf{S}_{L_j, r_i}^l is the residual covariance, \mathbf{K}_{L_j, r_i}^l is the Kalman gain, $\hat{\mathbf{x}}_{L_j, r_i}^{l+}$ is the updated map of the j^{th} landmark, and $\mathbf{P}_{L_j, r_i}^{l+}$ is the updated covariance matrix. Additionally, $\mathbf{H}_{L_j, r_i}^{l-}$ is computed as:

$$\mathbf{H}_{L_j, r_i}^{l-} = \frac{\partial \mathbf{h}(\mathbf{x}_i(k), \mathbf{x}_{L_j, r_i}(k))}{\partial \mathbf{x}_{L_j, r_i}(k-1)} \bigg|_{\hat{\mathbf{x}}_{L_j, r_i}^{l-}(k), \mathbf{x}_{r_i}^l(k)}. \quad (16)$$

For unobserved landmarks, the predicted values are replaced with the posterior ones; in other words, $\mathbf{P}_{L_j, r_i}^{l+}(k) = \mathbf{P}_{L_j, r_i}^{l-}(k)$ and $\hat{\mathbf{x}}_{L_j, r_i}^{l+}(k) = \hat{\mathbf{x}}_{L_j, r_i}^{l-}(k)$.

3.3 Localization (weight computation and resampling)

In this step, the final localization process is performed through computation of the particles' weights and a resampling process. To this end, the normalized weights are computed as follows:

$$w_{L_j, r_i}^l(k) = \frac{p(\mathbf{z}_{L_j, r_i}(k) | \hat{\mathbf{x}}_{L_j, r_i}^{l+}(k), \mathbf{x}_{r_i}^l(k))}{\sum_{i=1}^N p(\mathbf{z}_{L_j, r_i}(k) | \hat{\mathbf{x}}_{L_j, r_i}^{l+}(k), \mathbf{x}_{r_i}^l(k))}, \quad (17)$$

where $w_{L_j, r_i}^l(k)$ is the computed weight for the l^{th} particle corresponding to the j^{th} landmark and the i^{th} robot, and $p(\mathbf{z}_{L_j, r_i}(k) | \hat{\mathbf{x}}_{L_j, r_i}^{l+}(k), \mathbf{x}_{r_i}^l(k))$ refers to the conditional PDF. It is worth a reminder that, in this paper, the measurement noise vector is a Gaussian one, and therefore:

$$p(\mathbf{z}_{L_j, r_i}(k) | \hat{\mathbf{x}}_{L_j, r_i}^{l+}(k), \mathbf{x}_{r_i}^l(k)) = N(\hat{\mathbf{z}}_{L_j, r_i}^{l+}(k), \mathbf{R}_{z_{ij}}), \quad (18)$$

where $\hat{\mathbf{z}}_{L_j, r_i}^{l+}(k)$ is computed in a similar way to Eq. 13, but replacing $\hat{\mathbf{x}}_{L_j, r_i}^{l-}(k)$ and $\hat{y}_{L_j, r_i}^{l-}(k)$ with $\hat{\mathbf{x}}_{L_j, r_i}^{l+}(k)$ and $\hat{y}_{L_j, r_i}^{l+}(k)$, obtained from the update step, respectively. Next, the resampling process is performed using $w_{L_j, r_i}^l(k)$, and finally a set of N equally weighted particles is generated; these are averaged to produce the final estimate at each sample time k , $\hat{\mathbf{x}}_{L_j, r_i}(k)$ and $\hat{\mathbf{x}}_{r_i}(k)$.

It is worth mentioning that if no measurement is available, the predicted particles in (7) will be used for particle generation in the next step; additionally, they are weighted equally for the final mapping and localization, $\hat{\mathbf{x}}_{L_j, r_i}(k)$ and $\hat{\mathbf{x}}_{r_i}(k)$.

4 Coordinate alignment and map-merging in multi-robot SLAM

As mentioned earlier, two robots, R_1 and R_2 , explore the environment and perform modified Fast-SLAM as explained in the previous section. However, localization and mapping are performed in their corresponding coordinate frames, G_1 and G_2 respectively, and each robot is only aware of its own coordinate frame. In order to merge and fuse the separately provided maps, it is necessary for the robots to meet each other. Although there exist many rendezvous approaches to force the robots to meet each other, such as the one presented in (Roy and Dudek, 2001; De Hoog et al., 2010; Zaman et al., 2011; Meghjani and Dudek, 2012; Meghjani and Dudek, 2011), for the sake of simplicity, in this paper it is assumed that the robots travel in such a way that they can meet each other at least once during the mission. When the robots meet each other, it is possible to transform the coordinates. For this purpose, the proposed method of (Zhou and Roumeliotis, 2006) is employed in this paper.

To this end, when robots R_1 and R_2 meet using the measurements from the installed lidar sensors, the relative angles between the robots (${}^2\theta_1, {}^1\theta_2$) and the distance between the robots (ρ) are measured, where ${}^2\theta_1$ is the angle of the first robot as measured by the second one and ${}^1\theta_2$ is the angle of the second robot measured as by the first. Using the geometrical method presented in (Zhou and Roumeliotis, 2006), the rotation matrix, ${}^{G_1}P_{G_2}$, and the translation vector, ${}^{G_1}P_{G_2}$, for transforming a vector in frame G_2 to frame G_1 are computed. Readers are referred to (Zhou and Roumeliotis, 2006) for details of the proposed method. Figure 2 clarifies the geometrical relationships between the coordinates and the measurements. Following this, the map created by the second robot can be presented in G_1 as follows, and the two maps are merged.

$$\left[\hat{\mathbf{x}}_{L_j, r_2}^l(k) \right]_{G_1} = {}^{G_1}P_{G_2} \hat{\mathbf{x}}_{L_j, r_2}^l(k) + {}^{G_1}P_{G_2}, \quad (19)$$

where $\left[\hat{\mathbf{x}}_{L_j, r_2}^l(k) \right]_{G_1}$ refers to the coordinate of a vector in frame G_1 .

Finally, all the landmarks are presented in frame G_1 . If a landmark L_j has been visited by both robots, its corresponding

map is fused with $\hat{\mathbf{x}}_{L_j, r_1}^l(k)$ through simple averaging, and it is updated to achieve a more precise map. Moreover, similarly, the trajectory of the second robot can be expressed in G_1 as a global inertial frame.

Similarly, the map created by the first robot is presented in frame G_2 using ${}^{G_2}C_{G_1}$ and ${}^{G_2}P_{G_1}$, where $[\hat{\mathbf{x}}_{L_j, r_1}^l(k)]_{G_2}$ is obtained as follows:

$$[\hat{\mathbf{x}}_{L_j, r_1}^l(k)]_{G_2} = {}^{G_2}C_{G_1} \hat{\mathbf{x}}_{L_j, r_1}^l(k) + {}^{G_2}P_{G_1}. \quad (20)$$

Next, $[\hat{\mathbf{x}}_{L_j, r_1}^l(k)]_{G_2}$ is fused with $\hat{\mathbf{x}}_{L_j, r_2}^l(k)$ to provide more precise mapping through averaging.

To clarify the proposed method, the general proposed algorithm for multi-robot Fast-SLAM in a dynamic environment is presented and summarized in Algorithm 1.

5 Simulation results

To evaluate the performance of the proposed multi-robot SLAM algorithm in a dynamic environment, a series of simulation studies were conducted using MATLAB m-file codes; these are reported on in this section. Two Pioneer 3-DX mobile robots, each equipped with a lidar sensor, are simulated. As mentioned earlier in the article, two moving landmarks are considered and are modeled via their kinematic models.

As mentioned earlier, robots R_1 and R_2 explore the environment and observe the landmarks within it; each of them implements the SLAM process from its corresponding frame of G_1 or G_2 , respectively, and makes an independent map of the environment. The parameters in our simulation are considered as follows:

$$\begin{aligned} \omega_{L_1} &= 0.2 \text{ rad/sec}, \omega_{L_2} = 0.2 \text{ rad/sec}, v_{L_1} = 0.4 \text{ m/sec}, \\ v_{L_2} &= 0.6 \text{ m/sec}, \omega = 0.2 \text{ rad/sec}, v = 1 \text{ m/sec} \end{aligned}$$

All of the sources of noise are considered to be zero-mean Gaussian noise; the covariance matrices of the process (kinematics) noise, for both landmarks and robots, are specified as $0.001\mathbf{I}_3$, and the covariance matrices of the lidar noise are specified as $0.001\mathbf{I}_2$.

$$\text{Additionally: } {}^{G_2}T_{G_1} = \begin{bmatrix} 0.8142 & -0.5806 & 0 & 1 \\ 0.5806 & 0.8142 & 0 & -2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \Delta t = 0.65 \text{ sec}$$

$$\begin{aligned} [\mathbf{x}_{R_1}(0)]_{G_1} &= [-0.2 \ -0.5 \ 1.23]^T, [\mathbf{x}_{R_2}(0)]_{G_1} = [0 \ 0 \ 2.09]^T \\ [\mathbf{x}_{L_1}(0)]_{G_1} &= [-1.5 \ -1 \ 1.22]^T, [\mathbf{x}_{L_2}(0)]_{G_1} = [2.5 \ -1.5 \ 2.09]^T \end{aligned}$$

Once the robots meet one another, the transformation matrix between the two frames G_1 and G_2 is obtained and the maps can be fused to obtain a more precise map of the environment and achieve more precise localization performance. In the case of our simulation, the robots meet each other at 9.35 s and the estimated transformation matrix is as follows:

Step 0: Initialization

at each sample time k :

for each robot $r_i, i=1,2$:

for each particle $l=1,\dots,N$:

Receive: $\mathbf{x}_{r_i}^l(k-1)$

Step 1: particle generation: generate particles according to (7).

Step 2: Mapping:

Receive: $\mathbf{P}_{L_j, r_i}^{l+}(k-1), \hat{\mathbf{x}}_{L_j, r_i}^{l+}(k-1)$.

for each $L_j, j=1,2$:

• **if** no measurement is available but the

landmark has been visited **do**:

1. predict according to (10), (11) and (12)

2. let $\mathbf{P}_{L_j, r_i}^{l+}(k) = \mathbf{P}_{L_j, r_i}^{l-}(k)$ and $\hat{\mathbf{x}}_{L_j, r_i}^{l+}(k) = \hat{\mathbf{x}}_{L_j, r_i}^{l-}(k)$.

• **endif**

• **if** measurement $\mathbf{z}_{L_j, r_i}(k)$ is available **then**:

• do data association according to (14):

• **if** the landmark L_j is visited for the first

time **do**

1. initialization according to (8) and (9).

• **elseif** the landmark has been visited

previously **do**

1. predict according to (10), (11) and (12).

2. update according (15) and (16).

• **endif**

• **endif**

Step 3: Localization:

• **if** measurement $\mathbf{z}_{L_j, r_i}(k)$ is available **do**:

1. generate particle weights according to (17)

and (18).

2. perform resampling and generate equally

weighted estimates, resulting in:

$$\left\{ \frac{1}{N}, \hat{\mathbf{x}}_{L_j, r_i}^{l+}(k), \mathbf{x}_{r_i}^l(k) \right\}_{l=1}^N,$$

• **endif**

• **if** no measurement is available **do**

$$\left\{ \frac{1}{N}, \hat{\mathbf{x}}_{L_j, r_i}^{l+}(k), \mathbf{x}_{r_i}^l(k) \right\}_{l=1}^N,$$

• **endif**

Step 4: Coordinate alignment and map merging:

• **if** the robot visits another robot **do**:

1. compute the rotation matrix, ${}^{G_1}C_{G_2}$, and the translation vector, ${}^{G_1}P_{G_2}$.

2. transform maps in G_2 to G_1 according to (19).

3. transform maps in G_1 to G_2 according to (20).

4. fuse maps as explained in Section 4.

• **endif**

Return: $\hat{\mathbf{x}}_{L_j, r_i}(k)$ and $\hat{\mathbf{x}}_{r_i}(k)$ through averaging.

Algorithm 1. The general multi-robot Fast-SLAM-based algorithm with moving landmarks.

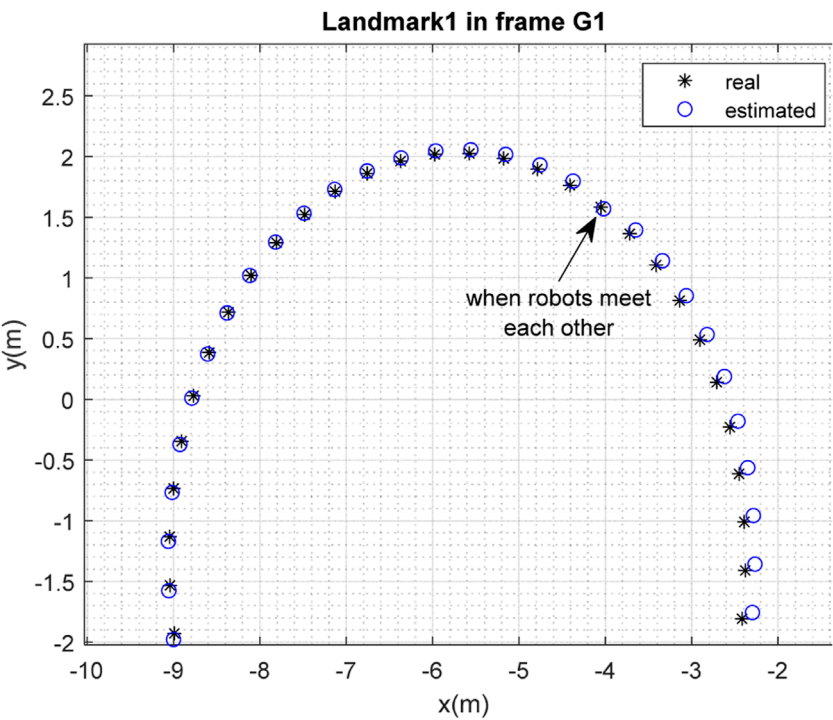


FIGURE 16
Real position of landmark L_1 and position estimated by robot R_2 in the frame G_1 (with different initial conditions).

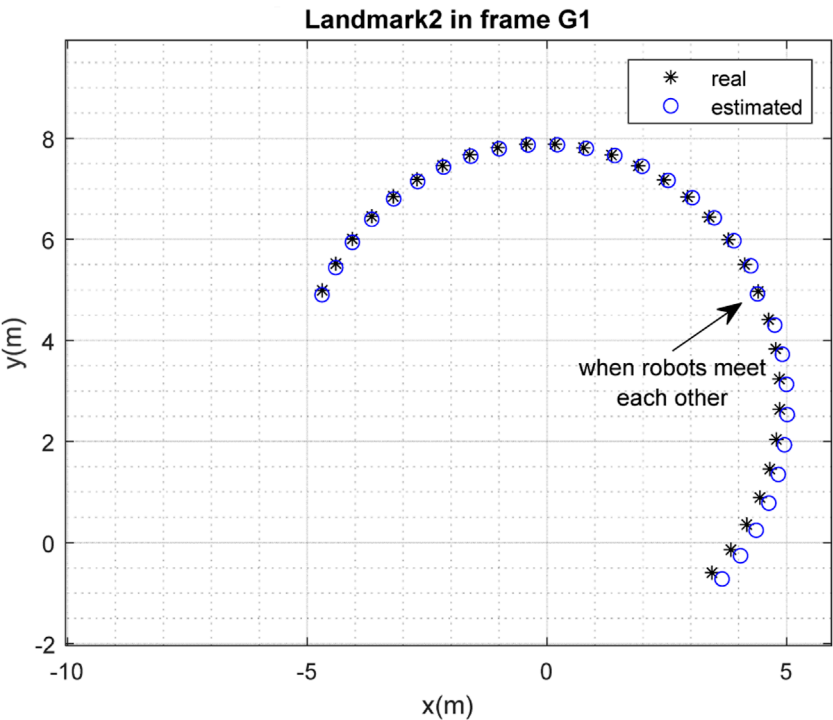


FIGURE 17
Real position of landmark L_2 and position estimated by robot R_2 in the frame G_1 (with different initial conditions).

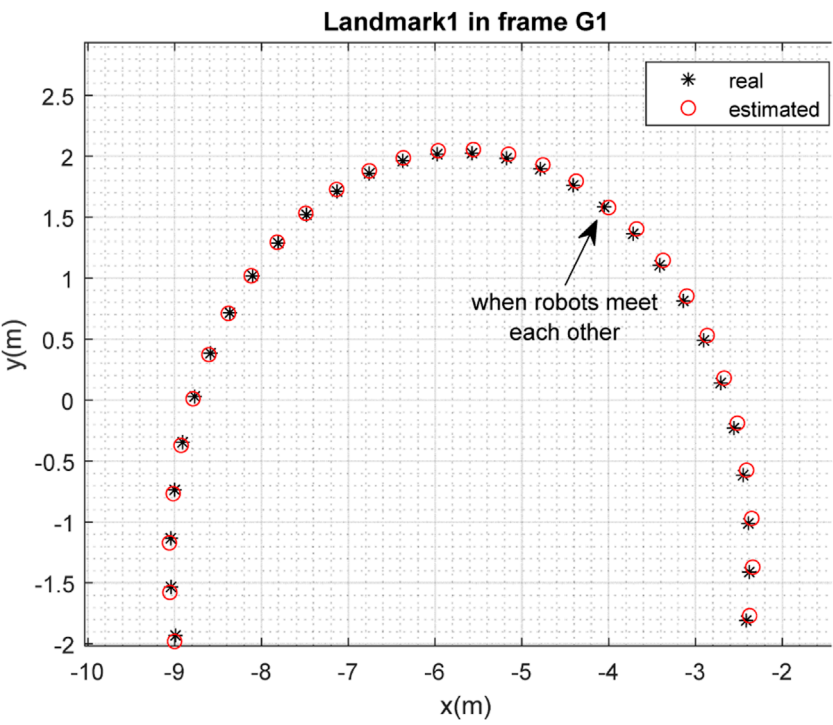


FIGURE 18
Real position of landmark L_1 and position estimated by robot R_1 in the frame G_1 (with different initial conditions).

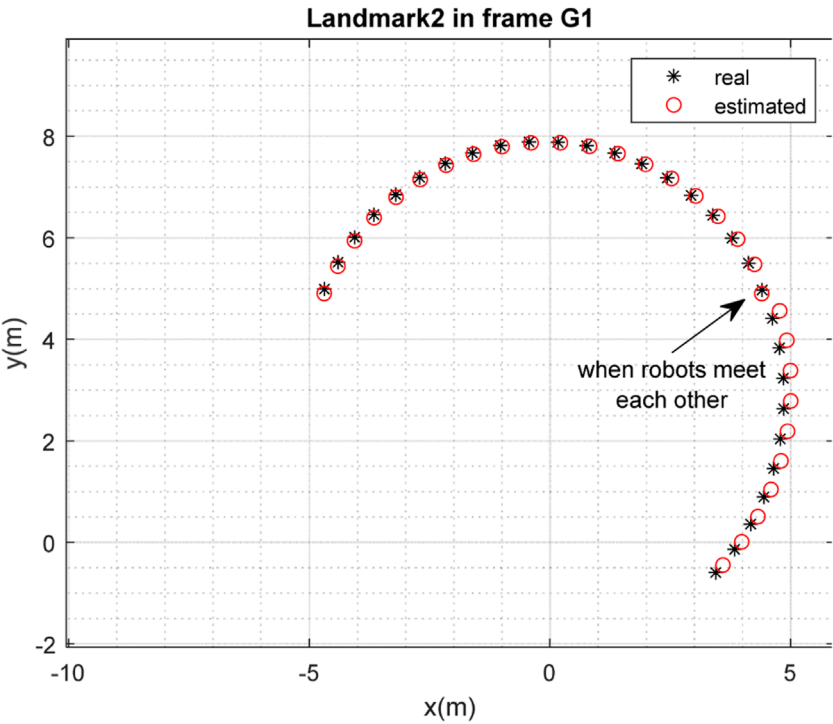


FIGURE 19
Real position of landmark L_2 and position estimated by robot R_1 in the frame G_1 (with different initial conditions).

$$G_2 T_{G_1}(\text{estimated}) = \begin{bmatrix} 0.8048 & -0.5935 & 0 & 1.0985 \\ 0.5935 & 0.8048 & 0 & -1.7098 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figures 3, 4 depict the results of the localization of robots R_1 and R_2 , respectively, in 2D space, expressed in the G_1 coordinate frame, which is selected as the global frame for this paper. Figure 5 depicts the estimated and real paths of robot R_2 in frame G_2 to provide a better illustration of how robot R_2 is performing localization in its own frame G_2 .

The real and estimated values of x_{r_i} , y_{r_i} , and ϕ_{r_i} , expressed in frame G_1 with respect to time, are depicted in Figures 6, 7 for $i = 1$ (robot R_1) and $i = 2$ (robot R_2), respectively. To show the precision of the estimation, Figures 8, 9 depict the corresponding errors: that is, the error between the real and estimated values ($e_{x_{r_i}}$, $e_{y_{r_i}}$, and $e_{\phi_{r_i}}$) in G_1 for $i = 1, 2$.

The results of the mapping in 2D space are shown in Figures 10, 11 for landmarks L_1 and L_2 obtained from R_1 , respectively, expressed in frame G_1 ; the results obtained from R_2 in frame G_1 are shown in Figures 12, 13. It is worth mentioning again that when the robots meet each other, the maps obtained from R_1 and R_2 are merged. It is clear from the figures that the localization and mapping results are significantly improved after the map-merging process.

To examine the efficiency of the method for the case in which some of the conditions are altered, the initial conditions were selected as follows:

$$\begin{aligned} [\mathbf{x}_{R_1}(0)]_{G_1} &= [0 \ 0 \ 1.23]^T, [\mathbf{x}_{R_2}(0)]_{G_1} = [1 \ 0 \ 2.09]^T \\ [\mathbf{x}_{L_1}(0)]_{G_1} &= [-2.5 \ -2.2 \ 1.22]^T, [\mathbf{x}_{L_2}(0)]_{G_1} = [3 \ -1 \ 2.09]^T \end{aligned}$$

The results of localization of the robots in frame G_1 are depicted in Figures 14, 15, and the results of the mapping are depicted in Figures 16–19. It is clear from the figures that the algorithm is successful in simultaneous localization and mapping under changing initial conditions.

6 Conclusion

In this paper, an efficient algorithm has been presented for a multi-robot SLAM problem with unknown initial correspondence in a dynamic environment, using a modified Fast-SLAM method. In our scenario, each robot independently searches the environment, observes the moving landmarks in the environment using a lidar sensor, and implements the SLAM algorithm. In order to distinguish the moving landmarks, kinematic models are considered for the landmarks, which led to a modification of the normal Fast-SLAM method in the form of the addition of a prediction phase to the method; additionally, data association was performed according to the predicted measurements obtained from this prediction step. Although the kinematic models of the landmarks are known within each robot's coordinate system, after the first meeting of the robots, an initialization is embedded in the algorithm to obtain the current positions of the landmarks, as they are unknown to the robots.

Since the initial correspondence of the robots is unknown (or, in other words, each robot performs the mapping from the perspective of its own coordinate frame), a map-merging procedure was embedded in the proposed algorithm to fuse the independent maps of the robots when the robots meet each other. This map-merging is only possible when the relative transformation matrix of the robots' inertial frames is computed, which occurs when the robots meet each other. For this purpose, a geographical approach to compute this transformation matrix was embedded in the proposed algorithm.

The performance of the proposed method was evaluated through simulations in MATLAB. It can be concluded from the simulation results that although each robot was able to solve the SLAM problem with an acceptable level of performance, the accuracy of SLAM was significantly improved when the robots met each other and map-merging was performed.

Although the proposed method showed very good performance in simulations in the MATLAB environment, it will perhaps encounter some difficulties in a real environment, such as model mismatches in relation to both the landmarks and the robots due to noise, and the occurrence of drift, for example, in the inertial sensor measurements. Additionally, designing a rendezvous method for the robots to force them to visit each other can represent another issue in a real environment.

Generalization of the proposed method for the general case of a multi-robot and multi-landmark system, with more than two robots and two moving landmarks, is proposed for future work. Embedding some approach to object detection, instead of using the kinematic models of the landmarks, would also add significant value to this research.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

Author contributions

HK: Investigation, Methodology, Software, Writing—original draft. NS-N: Conceptualization, Methodology, Project administration, Supervision, Validation, Writing—original draft. AM: Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Writing—review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/R02572X/1.

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 author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

References

- Almadhoun, R., Taha, T., Seneviratne, L., and Zweiri, Y. (2019). A survey on multi-robot coverage path planning for model reconstruction and mapping. *SN Appl. Sci.* 1, 847–924. doi:10.1007/s42452-019-0872-y
- Badalkhani, S., and Havangi, R. (2021). Effects of moving landmark's speed on multi-robot simultaneous localization and mapping in dynamic environments. *Iran. J. Electr. Electron. Eng.* 17 (1), 1740. doi:10.22068/IJEEE.17.1.1740
- Burrell, T., West, C., Monk, S. D., Montazeri, A., and Taylor, C. J. (2018). "Towards a cooperative robotic system for autonomous pipe cutting in nuclear decommissioning," in UKACC 12th International Conference on Control (CONTROL), Sheffield, UK, 5-7 Sept. 2018 (IEEE), 283–288.
- Chen, B., Peng, G., He, D., Zhou, C., and Hu, B. (2021). "Visual SLAM based on dynamic object detection," in 2021 33rd Chinese Control and Decision Conference (CCDC), Germany, 22-24 May 2021 (IEEE), 5966–5971.
- Debeunne, C., and Vivet, D. (2020). A review of visual-LiDAR fusion based simultaneous localization and mapping. *Sensors* 20 (7), 2068. doi:10.3390/s20072068
- De Hoog, J., Cameron, S., and Visser, A. (2010). *Selection of rendezvous points for Multi-robot exploration in dynamic environments*.
- Godsill, S. (2019). "Particle filtering: the first 25 years and beyond," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), China, 12-17 May 2019 (IEEE), 7760–7764.
- Guan, H., Qian, C., Wu, T., Hu, X., Duan, F., and Ye, X. (2023). A dynamic scene vision SLAM method incorporating object detection and object characterization. *Sustainability* 15 (4), 3048. doi:10.3390/su15043048
- Kretschmar, H., Grisetti, G., and Stachniss, C. (2010). Lifelong map learning for graph-based SLAM in static environments. *KI-Künstliche Intell.* 24, 199–206. doi:10.1007/s13218-010-0034-2
- Li, A., Wang, J., Xu, M., and Chen, Z. (2021). DP-SLAM: a visual SLAM with moving probability towards dynamic environments. *Inf. Sci.* 556, 128–142. doi:10.1016/j.ins.2020.12.019
- Liu, H., Liu, G., Tian, G., Xin, S., and Ji, Z. (2019). "Visual SLAM based on dynamic object removal," in IEEE International Conference on Robotics and Biomimetics (ROBIO), New York, 5-9 Dec. 2022 (IEEE), 596–601.
- Lluvia, I., Lazkano, E., and Ansuategi, A. (2021). Active mapping and robot exploration: a survey. *Sensors* 21 (7), 2445. doi:10.3390/s21072445
- Meghjani, M., and Dudek, G. (2011). "Combining multi-robot exploration and rendezvous," in *Canadian conference on computer and robot vision* (USA: IEEE), 80–85.
- Meghjani, M., and Dudek, G. (2012). "Multi-robot exploration and rendezvous on graphs," in IEEE/RSJ International Conference on Intelligent Robots and Systems, USA, 23-27 Oct. 2022 (IEEE), 5270–5276.
- Montazeri, A., Can, A., and Imran, I. H. (2021). *Unmanned aerial systems: autonomy, cognition, and control*. Unmanned aerial systems. USA: Academic Press, 47–80.
- Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). FastSLAM: a factored solution to the simultaneous localization and mapping problem. *Aaai/iaai* 593598. doi:10.5555/777092.777184
- Pancham, A., Withey, D., and Bright, G. (2020). Amore: CNN-based moving object detection and removal towards SLAM in dynamic environments. *South Afr. J. Industrial Eng.* 31 (4), 46–58. doi:10.7166/31-4-2180
- Romero, V., and Costa, O. (2010). "Map merging strategies for multi-robot fastSLAM: a comparative survey," in *Latin American robotics symposium and intelligent robotics meeting* (America: IEEE), 61–66.
- Roy, N., and Dudek, G. (2001). Collaborative robot exploration and rendezvous: algorithms, performance bounds and observations. *Aut. Robots* 11, 117–136. doi:10.1023/a:1011219024159
- Sadeghzadeh-Nokhodberiz, N., Can, A., Stolkin, R., and Montazeri, A. (2021). Dynamics-based modified fast simultaneous localization and mapping for unmanned aerial vehicles with joint inertial sensor bias and drift estimation. *IEEE Access* 9, 120247–120260. doi:10.1109/access.2021.3106864
- Saeedi, S., Trentini, M., Seto, M., and Li, H. (2016). Multiple-robot simultaneous localization and mapping: a review. *J. Field Robotics* 33 (1), 3–46. doi:10.1002/rob.21620
- Saputra, M. R. U., Markham, A., and Trigoni, N. (2018). Visual SLAM and structure from motion in dynamic environments: a survey. *ACM Comput. Surv. (CSUR)* 51 (2), 1–36. doi:10.1145/3177853
- Stachniss, C., Leonard, J. J., and Thrun, S. (2016). *Simultaneous localization and mapping*. Germany: Springer Handbook of Robotics, 1153–1176.
- Theodorou, C., Velisavljevic, V., and Dyo, V. (2022). Visual SLAM for dynamic environments based on object detection and optical flow for dynamic object removal. *Sensors* 22 (19), 7553. doi:10.3390/s22197553
- Vidal, F. S., Barcelos, A. d. O. P., and Rosa, P. F. F. (2015). "SLAM solution based on particle filter with outliers filtering in dynamic environments," in IEEE 24th International Symposium on Industrial Electronics (ISIE, USA, 3-5 June 2015 (IEEE), 644–649.
- Zaman, S., Slany, W., and Steinbauer, G. (2011). "ROS-based mapping, localization and autonomous navigation using a Pioneer 3-DX robot and their relevant issues," in *Saudi international electronics, communications and photonics conference* (Germany: SIEPC), 1–5.
- Zhou, X. S., and Roumeliotis, S. I. (2006). *Multi-robot SLAM with unknown initial correspondence: the robot rendezvous case* in IEEE/RSJ international conference on intelligent robots and systems, USA, October 23–27, 2022 (IEEE), 1785–1792.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.



OPEN ACCESS

EDITED BY

Shaoming He,
Beijing Institute of Technology, China

REVIEWED BY

Hyondong Oh,
Ulsan National Institute of Science and
Technology, Republic of Korea
Önder Tutsoy,
Adana Science and Technology University,
Türkiye
Defeng Wu,
Jimei University, China

*CORRESPONDENCE

Allahyar Montazeri,
✉ a.montazeri@lancaster.ac.uk

RECEIVED 11 November 2023

ACCEPTED 22 February 2024

PUBLISHED 12 March 2024

CITATION

Mansfield D and Montazeri A (2024), A survey
on autonomous environmental monitoring
approaches: towards unifying active sensing
and reinforcement learning.
Front. Robot. AI 11:1336612.
doi: 10.3389/frobt.2024.1336612

COPYRIGHT

© 2024 Mansfield and Montazeri. This is an
open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). 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 autonomous environmental monitoring approaches: towards unifying active sensing and reinforcement learning

David Mansfield and Allahyar Montazeri*

Lancaster University, School of Engineering, Lancaster, United Kingdom

The environmental pollution caused by various sources has escalated the climate crisis making the need to establish reliable, intelligent, and persistent environmental monitoring solutions more crucial than ever. Mobile sensing systems are a popular platform due to their cost-effectiveness and adaptability. However, in practice, operation environments demand highly intelligent and robust systems that can cope with an environment's changing dynamics. To achieve this reinforcement learning has become a popular tool as it facilitates the training of intelligent and robust sensing agents that can handle unknown and extreme conditions. In this paper, a framework that formulates active sensing as a reinforcement learning problem is proposed. This framework allows unification with multiple essential environmental monitoring tasks and algorithms such as coverage, patrolling, source seeking, exploration and search and rescue. The unified framework represents a step towards bridging the divide between theoretical advancements in reinforcement learning and real-world applications in environmental monitoring. A critical review of the literature in this field is carried out and it is found that despite the potential of reinforcement learning for environmental active sensing applications there is still a lack of practical implementation and most work remains in the simulation phase. It is also noted that despite the consensus that, multi-agent systems are crucial to fully realize the potential of active sensing there is a lack of research in this area.

KEYWORDS

reinforcement learning, environmental monitoring, active sensing, deep learning, robotics, multi-agent

1 Introduction

The fields of Artificial Intelligence (AI) and robotics are accelerating the world toward its next technological revolution. Advances in both of these fields bring about new and exciting technologies that can be used to help tackle many of the challenges we face in life on planet Earth. One of the most pressing issues is climate change for which environmental monitoring (EM) plays a vital role in understanding and mitigating the impact of both natural and human activity that contribute to this growing issue. Before we can implement *in situ* solutions, we first need a deep understanding of each specific issue which in many cases requires scientists to collect much more empirical

data at each site of interest, as in reality, no two sites are exactly the same. For many natural processes, over which we do not have direct control, forecasting is an invaluable practice. However, each case is unique and at many of the sites where forecasting would be beneficial researchers do not have enough consistent knowledge or data to make accurate predictions. A persistent EM system is paramount to the development of accurate foresight in these situations. However, traditional monitoring methods are often expensive, slow, dangerous or inefficient when compared to the full potential of an intelligent autonomous system.

Simply, EM is the task of making observations of an environmental phenomenon. In the literature EM is a blanket term which refers to many different applications such as pollution monitoring [Alvear et al. \(2017\)](#), boundary tracking [Jin and Bertozzi \(2007\)](#), search and rescue [Waharte and Trigoni \(2010\)](#), volcanology [James et al. \(2020\)](#), subterranean mapping [Neumann et al. \(2014\)](#) and many more similar applications. This paper also touches on other fields such as agriculture [Duckett et al. \(2018\)](#) and nuclear decommissioning [Martin et al. \(2016\)](#), wherein EM technology plays a vital role in various aspects of their operation. In all of these applications environmental phenomena often occur over large physical spaces, vary on massive scales and are largely unpredictable meaning they require persistent or periodic monitoring. Due to these complex and changing environments, it is necessary to design robots that can adapt their behaviour in real-time to fit the evolving environment according to their sensed data; this process is called active sensing [Yang et al. \(2016\)](#). In this paper, we discuss the emerging sub-category of active sensing concerned with monitoring environmental phenomena which will be referred to henceforth as *active environmental monitoring* (active EM).

Active EM entails complex agent behaviours to achieve diverse goals, necessitating real-time adaptability and decision-making capabilities. This behaviour requirement lends itself naturally to machine learning since hard-coding these required behaviours can be complex [Kober et al. \(2013\)](#). Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) is a branch of machine learning that involves training agents to make optimal decisions by interacting with the environment. This process includes trial-and-error interactions aimed at maximizing rewards or achieving specific goals, leading to the development of an optimal policy for decision-making across various scenarios [Mnih et al. \(2015\)](#). For example, a system that has to operate in outdoor environments may be subject to multiple lighting levels, extreme weather conditions, and large areas of operation. RL has been demonstrated as an effective solution to allow robots to navigate under such varying conditions [Maciel-Pearson et al. \(2019\)](#). The increasing popularity of both RL and EM are demonstrated in [Figure 1](#), evidencing the growing importance of research and development in both fields. However, not all the literature that involves EM systems is released under the term 'environmental monitoring' and standardizing this will help research accelerate.

In this review paper, a comprehensive overview of the application of RL to EM problems is provided. Upon reviewing the literature it was found that there are numerous trends in RL state representation across different EM applications. For example, it is common to represent the environment as a grid of coloured "tiles" that encode information about a spatial area [Padrao et al. \(2022\)](#); [Luis et al. \(2020\)](#). Accordingly, an alternative classification

of RL algorithms in terms of continuous or discrete state and action spaces is included. This allows one to pick an algorithm based on the constraints of common EM problems. It is also shown by means of a unified formulation that the nature of RL and active sensing problems are complementary. In the proposed framework, both problems are described by a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) which facilitates both single and multi-agent systems. Despite the synergy between these two problems and the growing body of research in both fields, it is found that there is a lack of practical implementation, environment-realistic simulation environments and research into multi-agent reinforcement learning (MARL) approaches. This paper provides researchers with a condensed view of common approaches to the application of RL for active EM and unifies them under a suitable framework.

2 Reinforcement learning for active environmental monitoring

2.1 Previous surveys

There are a number of previous surveys covering the application of RL to robotics through which their compatibility is well discussed. [Kober et al. \(2013\)](#) is a 2014 survey on the application of RL to robotic platforms. It gives a comprehensive explanation of how RL can be used in robotic research and highlights the challenges in the field. [Azar et al. \(2021\)](#) looks at the application of DRL to Unmanned Aerial Vehicles (UAVs). The authors note some open EM applications and their associated problems where RL and DRL can be Utilized. [Lapeyrolerie et al. \(2022\)](#) discusses how RL is suited to conservation applications and provides a good background that helps justify further research and the joining of these two fields. [Canese et al. \(2021\)](#) reviews the current challenges and applications of Multi-Agent Reinforcement Learning (MARL), again noting how developments in these systems would be well suited to EM problems. Also in the domain of MARL, [Zhu et al. \(2022\)](#) discusses communication in MARL systems and [Oroojlooy and Hajinezhad \(2023\)](#) reviews cooperative MARL, offering a novel classification system for how these systems are designed and trained. While much of the previous work highlights the potential of robotic platforms trained with RL for application to EM, they do not discuss the topic in great detail or highlight the current specific challenges that EM brings.

Complementary to the previous surveys, this paper brings the following contributions: RL for robotic platforms is framed as an active EM problem, allowing for a clear picture of how RL can benefit such systems. This is done with a classification of RL by mission requirements rather than by algorithmic properties which is more suited for the research into active EM. It is also shown that active EM problems can be framed as a general active sensing problem and they are unified with RL problems under a single framework. In fact, the idea of *active environmental monitoring* generalizes the current approaches under a single framework by which state-of-the-art implementations are presented. A general RL-based active EM system is visualized in [Figure 2](#). Finally, the challenges of practical implementation and open questions in the context of active EM problems are discussed.

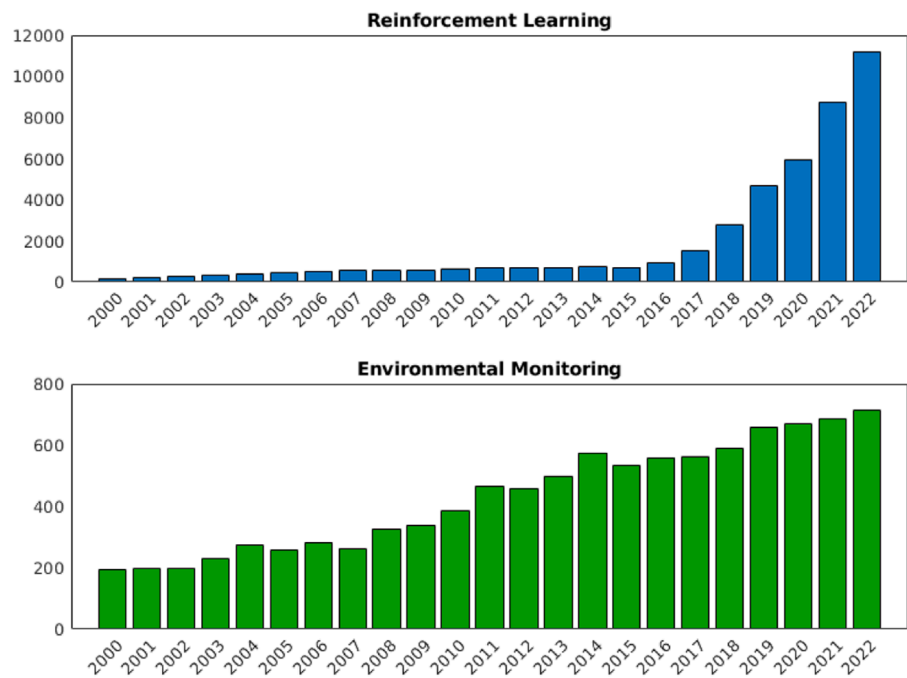


FIGURE 1
Number of publications with 'RL' and 'EM' in the title. Scraped from Google Scholar [Strobel, \(2018\)](#).

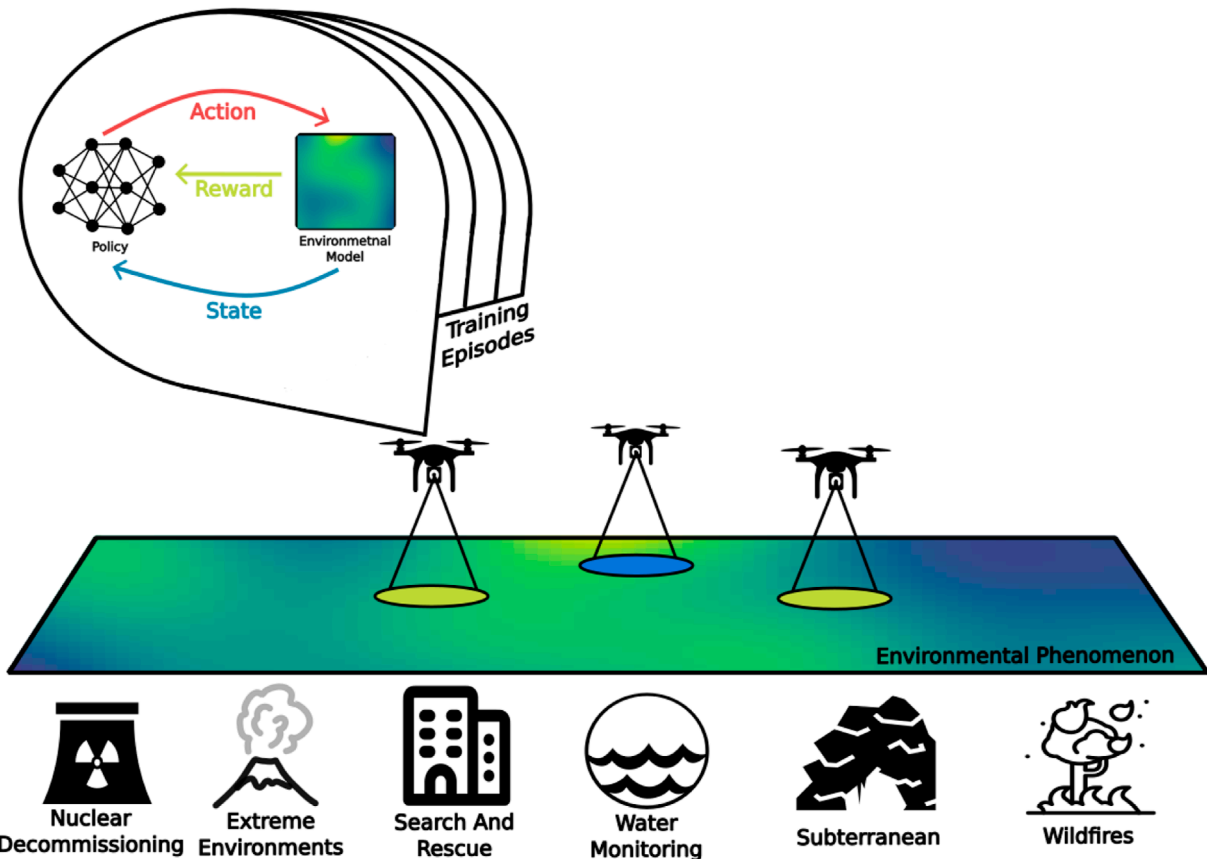
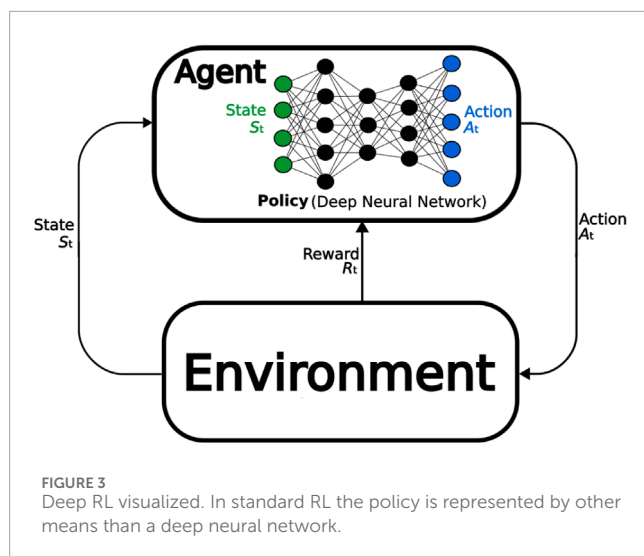


FIGURE 2
A general environmental monitoring problem, showing a model-based deep reinforcement learning trained UAV system.



2.2 Reinforcement learning terminologies

Reinforcement Learning (RL) is the process of training an *agent* to learn a *policy* that picks *actions* based on an *observation* of the *environment*. Sequences of individual actions should accumulate to a desired overall behaviour. The environment is represented and described by a *state*. Effective state-actions pairs are reinforced by a *reward*. The goal of the agent is to maximize the long-term (collective) reward by updating its policy. The goal of the policy is essentially to map state-action pairs to rewards, allowing us to choose the most effective action for a given state. RL can use various function approximates to represent the policy, including linear models or decision trees. This paper is also concerned with DRL. The defining difference between RL and DRL is that the policy in DRL is represented by a neural network. This is illustrated in Figure 3.

Different environments and different learning agents permit different actions. All of the actions that are valid for a given environment make up the *action space*. In a given state, the agent can choose any action in the action space. Actions are chosen either according to a policy or an exploration strategy. Action spaces can be continuous or discrete. Continuous action spaces can be used for tasks such as sending commands directly to robot actuators, whereas discrete action spaces are generally more abstract and involve more high-level commands such as, ‘move forward’ or ‘turn around’. Having an overly complex action space can lead to difficulties in training and the learning algorithm may not converge to an optimal policy. On the other hand, action spaces that are too simple may result in sub-optimal and limited behaviour.

In general RL and DRL algorithms are divided into model-based and model-free approaches. Model-based approaches have direct access to a learned or given model of the environment whereas model-free agents do not. Model-based techniques give an agent more information on the next optimal action to take, meaning that agents can quickly learn policies and make projections of future rewards. However, this is only true if the model of the environment is accurate. In lots of cases, the environments are complex and predicting them accurately is a difficult task in

and of itself. That being said, model-based algorithms that have access to a strong environmental model can give state-of-the-art results and show a substantial improvement in sample efficiency Silver et al. (2017); Kaiser et al. (2020). However, model mismatch can have some detrimental consequences. For instance, the agent may learn to exploit the bias or inaccuracies in the model causing poor performance in the real-world environment. In cases where a reliable model is not present then model-free algorithms perform better Lapeyrolerie et al. (2022). RL algorithms can also be classified as on-policy where actions are chosen from the same policy used during evaluation Schulman et al. (2017) or off-policy algorithms where agents sample their actions from a separate policy Jang et al. (2019); Lillicrap et al. (2019). Algorithms can also be value-based or policy gradient-based. Value-based algorithms use a value function to quantify the ‘goodness’ or value of taking a certain action in a certain state. The most well-known family of value-based algorithms is the Q-learning approach Watkins and Dayan (1992). Instead, in policy-gradient methods, the agents perform gradient ascent on the expected reward function to find the optimal policy. There is a class of RL algorithms called Actor-Critic methods that combine different aspects of both value-based and policy-gradient-based RL algorithms. This class of algorithms consist of two main components: the actor, which is responsible for taking actions and learning a policy that maximizes reward, and the critic, which is responsible for estimating the value of taking an action in a given state. This helps the actor to make more informed decisions. Actor-critic methods have been used to produce some very promising results Mnih et al. (2016); Haarnoja et al. (2018).

Generally speaking, various RL and DRL algorithms are reviewed and discussed in depth in the literature and an interested reader is referred to these papers for further discussions Lapeyrolerie et al. (2022); Kober et al. (2013); AlMahamid and Grolinger (2021). However, in the context of EM, the active sensing objectives are different for each application and are usually partially constrained according to the operating environment. Therefore, it would be more beneficial to classify the algorithms for EM applications based on their environment and action spaces. For example, we may need to train an agent to scan a given area. Depending on the mission objectives, we can decide to either abstract this area into smaller sub-areas allowing us to use a limited and discrete state and action space, or we can choose to predict a continuous model of the region of interest for which we need a continuous learning environment. A classification of standard and state-of-the-art RL algorithms used for EM applications is illustrated in Figure 4.

2.3 Reinforcement learning for single-agent environmental monitoring systems

Hard-coding complex and adaptive behaviours for robotic systems can often be a very difficult task, this is compounded by the fact that robots also often have non-linear dynamics and operate in complex environments whilst under considerable amounts of uncertainties and constraints. Unlike traditional programming techniques, RL allows engineers to take a more end-to-end approach when developing control and navigation algorithms. In

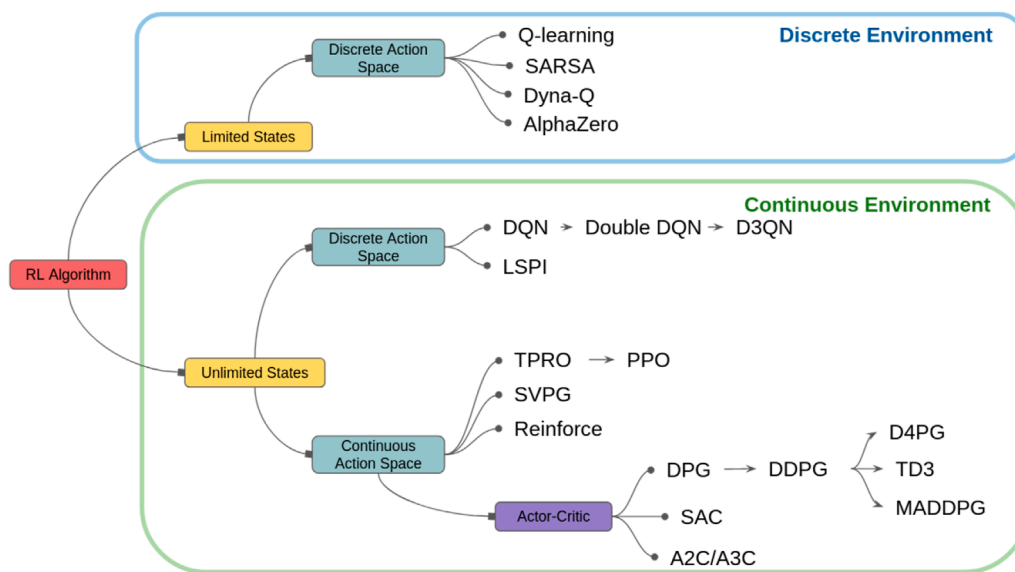


FIGURE 4
RL Classification by environment type AlMahamid and Grolinger (2021).

this approach, the desired behaviour of a learning agent is described by designing a suitable reward function. The design of this reward function, a process which is called reward shaping, is much more high-level than programming the behaviours directly.

Reward functions should describe the goals of a system rather than specific behaviour. Under a well-designed reward function and a suitable RL algorithm, an agent can learn to behave optimally. A general reward function might take the form:

$$R = R_{goal} + R_{penalty}, \quad (1)$$

where R_{goal} is a positive scalar that is given to an agent for actions that are productive to the overall goal of the system and $R_{penalty}$ is a negative scalar that penalizes the robot for undesirable actions that are counterproductive to the goal. For example, Consider designing a reward function to teach a robot to follow a line on the x-axis to reach a goal.

$$R_{goal} = \frac{1}{D} + \alpha v_x, \quad (2)$$

where D is the distance to the goal, v_x is the agent's velocity in the x-direction and α is a constant controlling the contribution of velocity to the reward. Here R_{goal} rewards the robot for moving towards the goal at a higher velocity. To accompany this we might have:

$$R_{penalty} = -|y|^2 \quad (3)$$

where y is the robot's y coordinate. Here, $R_{penalty}$ increases with the square of the distance from the line, and thus encourages the robot to stay on the x-axis by administering a negative reward (penalty). Since the overall goal is for the agent to maximize the cumulative rewards we can incorporate the concept of return G which is the sum of the discounted rewards from the current time step

$$G = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \quad (4)$$

The discount factor γ influences the agent's behaviour by balancing immediate rewards with long-term objectives. As the agent takes actions it receives rewards; the value of these rewards is responsible for shaping the policy or value function. It is possible for agents to find a way to exploit the reward function, where the policy may converge to undesirable behaviour that never the less returns high rewards. Furthermore, reward shaping should pay close attention to both environmental and system-specific constraints such as actuator saturation or limited power supplies. For instance, small rewards or penalties might be given to encourage the agent to also minimize actuator effort. R_{goal} also poses a divide by zero error when the agent has reached the goal. It is common in this case to give the agent a large conditional reward for completing the mission.

In EM applications not only the complexities of the robot itself, but also the operating environments are intrinsically challenging. Furthermore, the missions that the robots are trying to accomplish can be complex. For instance, if the robot is expected to find the shortest path, to cover an entire target area, or detect a specific target in a search and rescue (SAR) operation. While some robots may be expected to operate in the same region for every mission, observing the same lake on a daily basis, for example In such scenarios, while the area may remain the same and known to the robot, the conditions can vary dramatically across multiple different time scales. In other applications, such as fire-fighting or emergency SAR the agents may have to adapt to completely new and *a priori* unpredictable environments in every mission. This means that effective active EM systems should be able to adapt online to a wide range of conditions and environments. Using RL techniques is one way by which we can generalise the performance of an agent and extend the functionality to multiple different environments.

It should also be noted that any real active EM system, that is put into a real use case scenario, should be designed with great care towards the energy constraints. This is because the time scale of these missions can be long undefined prior to the mission start

and safe refueling locations may be few and far between. For EM systems to be effective and worthwhile they need to bring long-term autonomous persistent monitoring to the region of interest Egerstedt et al. (2018). With current battery technology, the vast majority of systems will have to stop and recharge. This is a much bigger issue for some platforms than others, for example, small autonomous multi-rotor UAVs have a very limited flight time Azar et al. (2021). This constraint adds additional layers of complexity for traditional optimization that can be quite simply added to a reward function in RL, making it easier to consider from the beginning of the design process, perhaps at the cost of additional training time.

One EM example where RL and DRL have proved to be effective is the patrolling problem Luis et al. (2020). In this problem, the agent must cover an area effectively under time and energy constraints. Furthermore, this problem can be inhomogeneous (when some areas of the region of interest may be of more importance than others). Here the agent must learn to cover the target area completely while more regularly visiting some areas than others. In real situations, these areas of high importance will also change themselves meaning that no area can remain forgotten for too long as it may have become important as time passes. One can see the complexity of the problems that active EM can pose and how RL offers a high-level and intuitive design that leads to the desired behaviour through trial and error.

One final note on the application of RL to EM is that in environmental sciences it is standard practice to use simulations for predictions and research. As a result, there are many sophisticated and standard simulations which can generate data and model environmental processes. These simulations could be used to train RL and DRL agents for active EM. This potentially means developers of RL systems do not have to build their own simulations which, due to time and budget restrictions, could potentially over-simplify the environmental dynamics negatively impacting the performance of the RL agent. And so there is a potential area of technical contribution that can be made by porting these simulations to popular development platforms in robotics like Gazebo which is used commonly in conjunction with the Robot Operating System (ROS).

2.4 Reinforcement learning for multi-agent environmental monitoring systems

So far we have discussed RL in the context of single-agent systems. However, multi-agent robotic systems or robotic swarms are ideal solutions to many of the challenges imposed by active EM. Multi-agent systems are teams of autonomous agents that can cooperate on their movements, sensing and computations to achieve a common goal. Using multiple agents for active EM improves performance, time and energy efficiency and allows the system to cover wider areas. It also can add redundancy as there are multiple robots performing the same task. Designing multi-agent systems, with traditional methods, can be challenging as not only does each individual robot have to be able to perform in a challenging environment, but also the collective behaviour must also be able to

adapt in real-time to *a priori* unknown disturbances and avoid inter-agent collisions and ultimately adaptive behaviour is preferable in changing environments Busoniu et al. (2008).

In the same way that RL and DRL can be used to encode complex behaviour onto single agents, multi-agent reinforcement learning (MARL) can be used to encode the behaviours at the level of the swarm Kouzeghar et al. (2023). In MARL, the agents coexist and interact with the same shared environment. In such scenarios, they must consider how to maximize not only their individual reward but also a collective reward. When you have multiple agents in a shared environment their goals might be shared or opposed. In systems where agents share common goals the system is known as cooperative while in systems where agents have opposing interests, the system is called competitive. In systems with multiple optimization objectives, there may be a mix of competitive and cooperative behaviours and the reward function should reflect these goals. Consider the example of a team of factory robots tasked with cooperatively lifting a block towards a goal. Each agent will receive an individual reward for its own actions such as approaching and touching the block. However, we must now also consider the *joint reward*, which is a reward that all agents receive for actions that accomplish the overall system goal: to lift the block. In this case, the joint reward will be given when the block moves towards the goal location. While agents may learn to cooperate organically, joint rewards can also be given to directly encourage cooperation and shorten training time. In this example, this might be a joint reward for synchronized lifting actions.

Like the systems themselves, MARL training can either be centralized or decentralized. In centralized training approaches agents share experience from previous training episodes. This gives agents the opportunity to learn from data they would not have access to during execution Lin et al. (2023). In this case, the agents will have a richer idea of the operation environment and can learn from other agent's policies while in a fully decentralized training, agents still only learn from their own experience. It is worth noting that some MARL systems are simply composed of multiple single agents learning in the same environment but not sharing any training data.

2.5 Limitations of reinforcement learning

As discussed RL and DRL can be used to train robust and intelligent agents for a variety of important applications. However, these algorithms are not without their limitations. One of the biggest limiting factors for RL is that it is very sample-inefficient. Training agents to operate, even in the simplest system, can take hundreds of thousands of iterations Li et al. (2023). This is especially limiting in the field of robotics where agents will be deployed on physical systems. In many cases, it is near impossible to train agents in the practical domain. To avoid this issue it is common practice to develop simulations that mimic the real operation environment and train the agent there. But here lies another limitation of RL, model mismatch. This is the difference between the simulation and the true environment, these differences are always present and since one cannot simplify the real world, simulations must capture all of its most important elements. It is conceivable that agents trained in simulation will converge to an optimal policy but not behave optimally on the physical system due to this mismatch. This is

especially important for EM and in much of the literature addressed in this, paper simulation environments are not comprehensive. More dynamic solutions are available for robots, for example, in Feiyu et al. (2024) gazebo is used to train agents or in Liu and Bai (2021) where a Geographic Information System was used to generate terrain data. An important step for EM is to bridge the gap between existing engineering and environmental simulations and tools allowing researchers to utilize the advantages of both platforms. Despite these limitations practical results have still been achieved with constrained simulations but more accurate training environments often mean better and more reliable results.

The increasing complexity of simulation naturally brings an increase in training time with each iteration taking longer to process. A common critique is that development of RL can be slow since some agents require days to train. Complex systems also require powerful computational resources. RL algorithms are also known to be very sensitive to training parameter values such as learning rate and discount factor and attempts at training may lead to no solutions. Tuning these parameters relate to the balance of exploration and exploitation. This refers to whether an agent should choose to try a new action in a given state (explore) or take an action which it knows will bring some reward (exploit). Exploring may shed light on new, more desirable behaviour, but too much, may lead to intractable convergence times or no convergence at all. How to handle this trade-off is still an open question in RL. An incorrect balance of exploration and exploitation can lead to the singularity problem which is when the agent's policy or value function becomes unbounded or diverges during training.

For some applications of RL, one may encounter the issue of sparse rewards. The sparse reward problem refers to systems in which an agent may only receive rewards in certain scenarios. Like in collision avoidance, for example, if the agent does not meet obstacles often then it may not learn how to avoid them Hu et al. (2023). Sparse rewards might also be inherent problems of the operation environment themselves, such as in an underwater environment. Despite these limitations, RL and DRL are still promising solutions to real and complex problems, but researchers should be informed of these issues during development.

2.6 A general framework for active environmental monitoring problems

The active EM problem can be formulated as a Partially Observable Markov Decision Process (POMDP). A POMDP is a mathematical framework used to model decision-making problems, though unlike a normal Markov decision process (MDP) a POMDP does not assume the agent has access to the true underlying states. This is a useful practical assumption for real systems as EM robots have limited information about the underlying target phenomena and even their own states in practice. Since agents cannot be certain of the state of the underlying function, they maintain what is called, a *belief state*. The agent takes actions based on the current belief state. Each action moves the agent to a new state and the agent receives a reward based on this outcome. The goal of an active sensing agent solving a POMDP is to take actions that provide the most information and maximize the cumulative reward over time. One can change how much an agent prioritizes immediate

reward vs. cumulative reward by leveraging a discount factor. It is common for RL problems to be modelled as an MDP or POMDP in the literature. Nevertheless, in this work, we propose a more general framework that also includes active sensing techniques as a special case. Table 1 shows how the components of a POMDP are comparable in RL and active sensing. To generalize further, we formulate a decentralized POMDP (Dec-POMDP) to include multi-agent systems reviewed in Section 2.4 as they become more and more popular. It is also common to formulate multi-agent systems as a Markov game as discussed in Littman (1994). The Dec-POMDP is a tuple that consists of:

- N is the number of agents (for a single agent system $N = 1$).
- \mathcal{S} is the set of possible states s_t of the environment.
- \mathcal{A} is the set of joint actions such that $\mathcal{A}^m = \{a_1^m, \dots, a_t^m\}$ is the set of actions of agent m .
- \mathcal{Z} is the set of joint observations such that $\mathcal{Z}^m = \{z_1^m, \dots, z_t^m\}$ is the set of observations of agent m .
- $\mathcal{T}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition probability function.
- $\mathcal{R}: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the joint reward function.
- $\gamma_t \in (0, 1]$ is a discount factor at time t .

Since it is assumed that an agent does not have access to ground truths about its state in a Dec-POMDP problem, a joint belief state is maintained which is the probability of being in a state given the history of all joint actions and observations:

$$\mathcal{B}_t(s_t) = P(s_t | z_{1:t}, a_{1:t}), \quad (5)$$

The joint action space \mathcal{A} is described by all the actions an agent can take, the observation space \mathcal{Z} is made up of all the possible observations an agent can make, and the transition function describes the probability of transitioning from the state s_t to state s_{t+1} when joint action a_t is taken. The reward function is a type of information that measures the 'goodness' of performing a joint action a_t in a given state s_t , the discount factor is used to decide how much to consider future rewards compared to immediate reward when taking the next action. It is worth noting that in model-based active sensing and RL, the agent can make a prediction of a future state based on its current belief and predicted observation \hat{z}_{t+1}

$$\mathcal{B}_{t+1}(s_{t+1}) = P(s_{t+1} | \hat{z}_{t+1}, a_{t+1}, \mathcal{B}_t(s_t)). \quad (6)$$

The goal of solving the Dec-POMDP is to find an optimal joint policy π^* that maximizes the total reward. A single joint multi-agent policy could be given as:

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^H \gamma_t \mathcal{R}(s_t, a_t) \right], \quad (7)$$

where H is the time horizon and the expectation is taken with respect to the probability of making a joint observation given the current belief state and joint action $P(z_{t+1} | (\mathcal{B}_t(s_t), a_t))$.

3 Methodologies in active environmental monitoring

In EM there are a number of common methodologies used to achieve specific outcomes or incorporate essential required

TABLE 1 Table summarizing the how components of the POMDP compare between active sensing formulations and RL formulations.

POMDP	Reinforcement learning	Active sensing	Summary
State \mathcal{S}	Represents the environment's true information which cannot be directly observed by the agent	Represents the underlying characteristic of the environment or phenomena that the agent is observing	Both maintain a belief state $\mathcal{B}_t(st)$. RL focuses on representing unobservable true state, while active sensing aims to understand observed characteristics to reduce uncertainty
Action \mathcal{A}	Actions are taken by the agent to influence the environment based on the current belief state $\mathcal{B}_t(st)$	Actions are the agent's decisions on where to sample and collect data. Actions are typically selected to maximize information gained and/or reduce uncertainty in the belief state $\mathcal{B}_t(st)$	RL actions influence environment directly, while active sensing actions aim to maximize information gain through data collection
Observation \mathcal{Z}	Observations in RL provide partial information about the hidden state	Observations represent the data or measurements collected by the sensors	Typically directly comparable used to infer the underlying state \mathcal{S}
Transition Function $\mathcal{T}:\mathcal{S} \times \mathcal{A} \times \mathcal{S}$	This is an internal model of the environment describing the probability of how the environmental states evolve from the current state to the next state under the influence of a specific action \mathbf{a}_t	A model of the environment describing how it's characteristics change when agents take a specific action \mathbf{a}_t . Typically models the dynamics of what is being sensed	In RL, transition function models state evolution; in active sensing, it models environmental characteristics change
Reward Function $\mathcal{R}:\mathcal{S} \times \mathcal{A}$	A single scalar feedback value that determines how effective a given action was at the current time step. It is a high-level description of the agent's desired outcomes	The value of information gain. Assigned based on how well the latest sensing action reduces uncertainty in the belief state	RL reward is more literal, it is intrinsic to learning and can incorporate the active sensing reward
Discount Factor γ_t	A scalar factor that determines how much weight the agent is giving to the future long-term rewards compared to the immediate reward	A factor that balances between exploration of new unobserved regions and exploitation of data in the previously sensed regions with high level of information	RL balances future rewards against immediate, active sensing balances exploration against exploitation of existing information

behaviour for an autonomous system which are addressed in this section. Specifically, we look at coverage, patrolling, path planning, collision avoidance, exploration, search and rescue, source seeking, source estimation and boundary tracking. In practice, these behaviours are often closely linked and may be used together on a single system. For example, a search and rescue system may need to use a coverage protocol to ensure no targets are missed within a search area. Thanks to the flexibility of RL it has been widely applied to each of these behaviours and the literature in this section is organized according to which methodology it is applied to.

The application of RL to these systems requires many design considerations. The learning agent needs to have access to information that allows it to learn how its actions influence the environment and how it can take subsequent actions to maximize reward. This requires careful design of a state-space and training environment that provides the needed information and is a reliable model of the true environment representing its features and constraints alike. The state space is often used as the input for the RL agent and thus the policy should be designed accordingly. This is also true of the action space which should reflect the environment and the demands of the physical system, a ground robot cannot change its altitude, for example. The state space is often used as the input

for the RL agent and the action space is the output. This means that a compatible RL algorithm and policy should be chosen. There is a significant body of research in RL that combines different learning algorithms with different policies. In a recent example, the authors replaced the standard Q-table in Q-learning with a neural network that used a radial basis function to represent the policy Wang Y. et al. (2024). While the state space may contain information not directly accessible in the real environment, the observation space should only contain information that is feasible for the agent to access. For instance, in source estimation, the agent would not be able to observe the true state of the field, only discrete sensor readings, but in simulation training this ground truth is known. For some ground truths, that are only available in the simulation environment, can be strategically used to reward the agent during training. The reward function itself is designed in a way that reflects the overall goal of the system and considers the constraints of the operating environment. It is common to use different rewards under different conditions like using one reward to encourage exploration of new areas when the local proximity of the robot is free of obstacles but in the presence of obstacles reward the agent for successfully avoiding them. This strategy can also help elevate the problem of sparse rewards present in some applications and environments.

3.1 State representation in active environmental monitoring

A large portion of the literature in RL is based on model-free approaches as they do not have the issue of model mismatch and are generally quicker to train. However, it is often the case in EM that the goal of the system is to produce a model of the environment, and hence, there is a higher proportion of model-based RL approaches in EM literature. The model state S can include the environmental phenomena we are observing, the dynamics of the environment, the dynamic state of the robot and other agents in the team. The way that the model is represented depends on the specific nature of the system, the monitoring task and the RL algorithm.

In reality, environmental phenomena are continuous but attempting to model them exactly can make training difficult and slow. As a result, it is often the case that the state space is discretized. In EM, this is often done via tile coding [Padrao et al. \(2022\)](#) in which the state space or environment is split into a 2D grid of tiles. In systems where there is a need to highlight specific information or features of the environment, one can use a relevance map. A relevance map is made by encoding these tiles with information that helps an agent make decisions. Tile encoding and image inputs are also viable for 3D state representations in active EM. For example, [Liu and Bai \(2021\)](#) uses DRL for coverage path planning and represents the 3D path of the agent as a 2D grey-scale image. This is achieved by what they term ‘the elevation compression method’ where the grey-scale value of each cell represents the altitude. Using feature extraction the CNN can successfully extract a 3D path from the grey-scale values. These types of state space representations are abstractions of the environment and are sometimes represented as an image, which is a concise way of presenting such information. Abstracting the environment to these discrete representations is good for reducing complexity but can limit performance to be optimal with respect to the abstracted environmental representation rather than the actual real-world environment.

As mentioned before, some research in RL has been applied to the patrolling problem. The patrolling problem in EM involves the agent revisiting the same locations periodically. Furthermore, in the real world, the application of patrolling is often inhomogeneous, meaning that some areas with a higher importance must be visited more frequently than others and the importance of the areas itself may also be variable over time. The environment in such problems can be represented as a relevance map. [Luis et al. \(2020\)](#) studied the case of monitoring the Ypacaraí Lake which is an essential source of drinking water and a popular tourist destination in Paraguay. The Ypacaraí Lake is a commonplace case study for the conservation of water resources. The Ypacaraí Lake is unfortunately suffering from an abnormal eutrophication process due to increased agriculture in the surrounding areas. Autonomous surface vehicles (ASVs) have been deployed and multiple different representations of the environment were considered. First, a model-free approach is used where the environment state S is just the position of the agents, then a binary relevance map that splits the lake into cells and colours visited cells as black and un-visited as white, which models a homogeneous patrol. Finally, a relevance map that colours cells a shade of grey which depends on the time since the last visit and their level of importance. The latter relevance map-based method leads to

the best results. The state is passed to a convolutional neural network for DRL.

Relevance maps have also been used for patrolling with UAVs [Piciarelli and Foresti \(2019\)](#). In this work, the relevance map is also used to solve an inhomogeneous patrolling problem and the colouring grid represents the importance of areas. The relevance map is combined with a binary mask, which tells us which states have been visited, before being passed to a Convolutional Neural Network (CNN), which is used to extract features from images. UAVs that utilize tile-coded maps have also been useful in SAR applications [Lu et al. \(2022\)](#). The target area is represented as a grid of sub-areas. Each sub-area or ‘pixel’ can have 1 of 3 values: imaged, being imaged or un-imaged. The goal is to train a policy to image all areas. It is also worth noting that in many UAV applications, the navigation module of an autonomous system will have to consider no-fly zones. These are areas where it may not be legal or safe to fly a small UAV. [Theile et al. \(2020\)](#) develop a state representation as a 3-channel map where the channels describe no-fly zones, target zones and take-off and landing zones. RL has also been discussed for use in agricultural applications [Binas et al. \(2019\)](#). A similar environmental representation is used for agricultural applications in MARL [Faryadi and Mohammadpour Velni \(2021\)](#). Here the world is split into a ‘grid world’ where each grid point is a possible state. The goal of the study is to find free states, target states and states blocked with obstacles.

Not all model-based approaches use relevance maps. For example, [Chen J. et al. \(2020\)](#); [Rückin et al. \(2022\)](#); [Viseras and Garcia \(2019\)](#) use Gaussian Processes to represent the environmental phenomena. A Gaussian process (GP) is a probabilistic model often used in EM to represent spatial and spatio-temporal fields. GP regression allows agents to develop continuous predictions of a field and give variance values at each prediction point based on a set of discrete observations. Learning a GP as a model of the environment is different from a GP being used as a function approximate for the RL policy (in place of a neural network in DRL, for instance) and the reader should take care not to confuse the two applications [Gadd et al. \(2020\)](#). GPs have also been implemented in RL to improve exploration during training [Zhao et al. \(2024\)](#).

It should be clarified that not all systems in the literature that use some kind of representation of the environment are model-based. The classification depends strictly on how the RL algorithm learns. For example, in [Placed and Castellanos \(2020\)](#) DRL is used to make decisions in active Simultaneous Localization and Mapping (SLAM). The distinction here is that even though we have a DRL agent guiding the robot's position for the SLAM process the inputs to the DNN are limited to the most recent laser scans, not the map produced by SLAM. Here the RL agent does not have any information retention about the model and cannot predict future observations based on previous ones.

3.2 Actions, observations and rewards in active environmental monitoring

As already discussed, it can be very difficult to hard code complex behaviour onto robotic systems and RL provides us with strategies for doing so. In EM and robotics, the nuances of the

desired behaviour depend on the specifics of each problem but there are overall mission goals common across the literature. This section reviews the different methodologies used to achieve those mission goals. As seen in the previous section, there are multiple ways to represent the environment/agent state \mathcal{S} , nonetheless, how a system can act to achieve its goals depends on its observations $\{\mathcal{Z}\}$, action space $\{\mathcal{A}\}$ and reward function \mathcal{R} . The main factor, driving the resultant behaviour of a system is the reward function. Ultimately, the principal aim of reward shaping is to guide training to be efficient and effective in producing the desired performance and optimal policy. Typically, rewards can be sparse or dense. A sparse reward function means the agent receives a reward only occasionally which can lead to long training times because agents need more experience to understand the implications of specific actions. Sparse rewards can be the result of an intentional design choice or of a sparse environment, like in deep-sea applications. Dense rewards are given more frequently and can produce more informative feedback on each action. While a well-designed dense reward function can speed up the training convergence and lead to good performance, it is worth noting that they can also impose the designer's preconceived notions of optimal behaviour on the agent rather than allowing the agent to learn for itself. It is also important to find a good balance between an action space which is simple enough to allow the training to converge in a reasonable time frame and also complex enough to allow optimal behaviour.

3.2.1 Coverage and patrolling

It is often the case in EM where agents will need to perform full area coverage. This is where a single agent or a team of m agents are tasked with covering an area completely under time or energy constraints. The coverage problem appears in a plethora of EM applications and is often implemented as a search step prior to other mission goals like source seeking or SAR which are described later. For example, in [Wu et al. \(2024\)](#) a coverage algorithm for multi-agent maritime SAR is proposed in which DRL is used to train agents to navigate to a grid location that has the highest probability of containing a target. This probability is based on novel drift simulations at a given time which predicts target trajectories and search boundaries that allow an agent to find targets thus facilitating their rescue. In real-world scenarios, the desired area to cover may not be defined *a priori*, like in the case of monitoring a wildfire, which can spread rapidly over time. Coverage also encapsulates patrolling which has been partly discussed in [Section 3.1](#). A summary of RL-based coverage and the relevant literature is provided in [Table 2](#). A large proportion of the areas we want to study with EM are boundless or undefined meaning that complete and exact coverage becomes difficult. Most readers are probably familiar with the travelling salesman problem where an agent, the salesman, must find the shortest path that visits every node once and then returns to the origin. Once the salesman has visited every node it can be said that the salesman has achieved coverage. The travelling salesman and coverage problems are generally NP-hard. That means as the problem scales up, the time needed to find an exact solution becomes infeasible [Ai et al. \(2021\)](#). This does not even account for the complex, dynamic, and partially observable nature of the environmental phenomenon aimed for monitoring. This is where RL is employed to learn the optimal coverage behaviour through interaction with the

TABLE 2 Characteristics of reinforcement learning based coverage and patrol.

Characteristic	Description
Objective	Maximize area coverage and importance (patrolling)
State Representation	Environment states as grid cells, importance information
Action Space	Movement actions and sensing actions
Reward Function	Rewards for total coverage, visiting new areas, patrolling importance
Observation	Environmental sensing target
Challenges	Sparse rewards, large spaces and inhomogeneous patrolling
Citations	Pham et al. (2018) ; Faryadi and Mohammadpour Velni. (2021) ; Theile et al. (2020) ; Luis et al. (2020) ; Lu et al. (2022) ; Kouzehgar et al. (2020) ; Luis et al. (2021) ; Ai et al. (2021)

environment. [Luis et al. \(2021\)](#) compared the use of DRL techniques to evolutionary algorithms for the application of patrolling the Ypacaraí lake. It was found that DRL performs better when there is a higher number of solutions which is expected in EM applications. Evolutionary algorithms are also well suited to active EM for similar reasons to RL and the combination of both is trending research [Zhang et al. \(2023\)](#).

In [Lu et al. \(2022\)](#), they point out the limitation of the field of view for visual sensors used in SAR UAVs. In the case of a camera, at lower altitudes less of the target area will be in the frame and there is a need for an efficient coverage algorithm to make sure every section of the target area is scanned. In this work, the scanning mechanism is included in the discrete action space $\{\mathcal{A}\}$ (scan left or scan right) along with the movement actions (fly straight, turn left, turn right). A DRL agent is trained using Deep Q Learning (DQN) to fully cover the target area while minimizing the number of actions taken. To this end, movement actions are given a small penalty (negative reward) \mathcal{R} as they only assist in the scanning of the target area but do not achieve it directly. There is also a penalty for when the UAV is not inside the target area. A reward is given proportional to the number of cells in a relevance map that have been scanned or are currently being scanned. A large reward is given if complete coverage occurs. Although this method was shown to improve the efficiency of UAV SAR coverage the system is too simplified to allow for true optimal behavior.

The coverage problem is also directly suited to multi-agent systems and it is easy to see how multiple agents with the possibility to collaborate can cover a given area more efficiently than a single-acting agent. [Pham et al. \(2018\)](#) proposed using MARL to design a distributed algorithm for coverage of an unknown field while minimizing overlapping scanned areas with other agents. The system is formulated as a Markov game, which is an extension of an MDP that includes a joint state space \mathcal{S} and action space for multiple

agents interacting with the same environment to optimize a joint policy. In this work, individual agents have a discrete action space \mathcal{A} that allows them to pick one of 6 actions: north, east, south, west, up or down. The set of actions agents take is called the joint action $\{\mathcal{A}^m\}$. Agents must reach a consensus on the next joint action. To ensure coverage while minimizing overlapping, agents get an individual reward \mathcal{R}^m proportional to the number of cells they covered minus the number of overlapping cells. There is also a global reward applied to all agents, equally used to help train the converge problem faster.

Kouzehgar et al. (2020) proposed MARL for a structured approach to area coverage for a team of ocean buoys. They modify the Multi-Agent Deep Deterministic policy-gradient (MADDPG) reward function to intrinsically bring about the collective behaviour of the swarm. The reward \mathcal{R}^m depends on the state \mathcal{S} of the swarm and thus, the reward itself has a collective behaviour. A positive reward is given based on the overlap between an 'agent coverage polygon' (a circle around the agent) and the 'region polygon' (the target area) and the total overlap which represents the total area covered. While the agents still receive some independent reward based on their actions the nature of this design is to reward the swarm as a whole for desirable *collective* behaviour rather than to encourage individual behaviour within the swarm.

3.2.2 Path planning and collision avoidance

Path planning and collision avoidance are often solved under the same problem framework. This is because, in practical situations, obstacles are often the reason that advanced path-planning techniques are required. In EM, especially SAR or indoor applications, a robust method for detecting obstacles and planning trajectories that allow the robot to navigate around them safely is crucial. However, many standard methods of collision avoidance are designed specifically for static obstacles Choi et al. (2021). RL provides a good option as it can be trained on many different environments without having to hard code behaviour for each one making it easy to generalize to unseen environments. A summary of RL-based path planning and collision avoidance algorithms and the relevant literature is provided in Table 3.

On the other hand, for UAVs that operate at a high altitude, such as for terrain mapping or pollution monitoring, there are not likely to be many obstacles that an agent will encounter, especially in single robot systems where there are no inter-agent collisions. It is, however, especially important for UAVs to have efficient and reliable path-planning capabilities due to their restricted flight times. Due to the nature of environmental processes, autonomous UAVs need to be able to adapt in real time to the information they collect. This is known as informative path planning (IPP) and is a crucial step in active EM. Rückin et al. (2022) combined DRL with a Monte Carlo tree search to reduce the computational burden of the predictive planning. This is also useful in development during the simulation stage as comprehensive active sensing simulations can be expensive. They also address the issue that, in the current literature, the action space is typically 2D $\mathcal{A} \subseteq \mathbb{R}^2$. This is because 3D action spaces $\mathcal{A} \subseteq \mathbb{R}^3$ are very large, making it difficult for training to converge. Nevertheless, this simplification does not make use of an aerial vehicle's principal virtue which is their ability to move in three dimensions. To fully unlock the potential of autonomous aerial systems for EM, more work like this must be done to include 3D action spaces. The action space $\mathcal{A} \subseteq \mathbb{R}^3$ considered is a discrete

TABLE 3 Characteristics of reinforcement learning based path planning and collision avoidance.

Characteristic	Description
Objective	Find safe, efficient paths and avoid collisions
State Representation	Environment states, possibly occupancy grid
Action Space	Movement actions or possible trajectories, sensing actions
Reward Function	Penalty for collisions, rewards for avoidance and goal reaching
Observation	Sensor data for obstacle detection (e.g., vision or LiDAR)
Challenges	High-dimensional states, real-time constraints and dynamic obstacles
Citations	Larsen et al. (2021); Popovic et al. (2020); Rückin et al. (2022); Chen et al. (2017); Yanes Luis et al. (2022); Choi et al. (2021); Woo and Kim. (2020)

set of possible measurement positions. The reward received by the RL agent depends on the reduction in predictive uncertainty of the environmental model and the cost of the action taken. The cost of the action is the flight time between measurement locations. The environmental model is a Gaussian Process which is updated using a Kalman filter Popovic et al. (2020). A Monte Carlo tree search is used to traverse the policy, which is represented as a convolutional neural network to decide the most information-rich actions. This is stopped when a budget or maximal search depth is reached. This tree search speeds up training and performance by removing the need to predict the informativeness of every possible action allowing for 3D action spaces. The authors showed their method performs better than multiple benchmarks.

Larsen et al. (2021) compared some state-of-the-art RL algorithms for navigating unknown environments on an ASV. The desired behaviour was efficient path planning and collision avoidance to reach a position goal. The algorithms tested were Proximal Policy Optimization PPO, Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3) and Soft Actor-Critic (SAC). The performance for each algorithm was measured not only by the total reward but also by mission progress, collisions, time taken and cross-track error. The algorithms were all implemented using Python and StableBaselines which are discussed in Supplementary Material. The simulated operation environments were randomly generated in calm ocean settings with moving obstacles being introduced later, in more challenging environments. Agents were tasked with reaching the goal without colliding along the way. The reward function \mathcal{R} punishes agents for colliding or being too close to obstacles and rewards them for progress towards the goal. After training, each agent was tested in environments of growing complexity. It was found that PPO produced agents that performed consistently across all environments in cases where the other agents did not. A repeat of these tests was run on agents

trained with a simplified reward function that was more sparse. In this case, it was found that a sparse reward function stunted the performance in every case. While PPO still shows superior performance its generalisation capability is not nearly as good as the case under a denser reward. This work sets a good example for comparing RL algorithms in EM and demonstrates that more standardization across the literature would afford a more empirical consensus on the performance of state-of-the-art algorithms in different applications.

As mentioned UAVs are an ideal choice for EM since they can fly, but much of the research for autonomous UAV navigation focuses on 2D space since 3D spaces are more complex. In Wang J. et al. (2024) a 3D path-planning algorithm with collision avoidance is proposed. Collision avoidance is inspired by the International Regulations for Preventing Collisions at Sea (COLREGS). It uses four distinct collision avoidance behaviours, which are part of the discrete action space, to avoid collisions with an obstacle. The UAV will undergo collision avoidance behaviour when an obstacle is within a certain distance of the UAV. Using a spherical Artificial Potential Field (APF) that has 3 zones: safety zone, collision avoidance zone and mandatory collision avoidance zone, the UAV can choose what action to take. The authors, note that RL can be difficult to apply to collision avoidance due to sparse rewards, and solves this by using the different zones of the APF to design a dynamic and conditional reward that rewards an agent for approaching its 3D goal way-point when no obstacles are present in the collision zone. When an obstacle is detected the agent is rewarded for avoiding collisions by using the correct collision avoidance behaviours derived from COLREGS.

COLREGS is designed for maritime missions and applying it to robotic applications in bridges the gap to real-world implementation. COLREGS is also leveraged in Li et al. (2021) with DRL to perform collision avoidance for an ASV. The APF is also used to generate a dynamic reward function that eliminates the problem of sparse rewards. In this solution, DQN is implemented with a continuous action space that represents the heading for the agent. This system, which is designed to adhere to the mandatory COLREGS results, is only tested in numerical simulations and so does not take into account the complexity of the real world and the highly dynamic and variable ocean environment. Another example of using RL to train agents to follow COLREGS is given in Wen et al. (2023). Here, a multi-agent team of ASVs are trained to avoid collisions with other agents in the system, environmental obstacles and other ships. An action evaluation network is pre-trained using a large data set of real, pre-recorded COLREGS avoidance trajectories. This is combined with an action selection network for cooperative path planning. The action selection network is trained on individual agents using DRL. The reward function rewards agents for successfully avoiding obstacles in such a way that aligns with COLREGS.

As mentioned, multi-agent systems hold massive potential and are the key to the future success of autonomous active sensing in EM. Cooperation for multiple agents adds complexity to both path planning and collision avoidance. One may choose to use RL in such systems as it allows agents to learn coordinated policies that may be complicated and time-consuming to hard code. Yanes Luis et al. (2022) uses DQN with prioritised experience replay to develop a multi-agent framework for IPP. The proposed method is tested and

TABLE 4 Characteristics of reinforcement learning based exploration.

Characteristic	Description
Objective	Discover unknown environment
State Representation	Environmental features, grid cells or frontiers
Action Space	Movement actions, sensing actions
Reward Function	Rewards for successful exploration, penalty for revisiting areas
Observation	Sensor data for map building or data collection
Challenges	Balancing exploration and exploitation, stopping conditions
Citations	Niroui et al. (2019); Hu et al. (2020); Chen et al. (Chen et al., 2020a; Chen et al., 2020b); Maciel-Pearson et al. (2019)

applied to the Peralta et al. (2020). They also use their framework to address the credit assignment problem Mao et al. (2020). This is a critical challenge in RL and refers to the difficulty of determining which actions or decisions taken by an agent in the past, contributes to a particular reward or outcome.

3.2.3 Autonomous exploration and SAR

Exploration is different to the coverage problem as it is concerned with autonomously gathering information about its environment to discover unknown areas often with the aim of achieving other tasks. Coverage on the other hand aims to visit every point of a given area. These two are often combined if the aim is to cover an *unknown* area Chen et al. (2021) but sometimes other goals are also combined with exploration. Autonomous environment exploration is required for applications like SAR Zuluaga et al. (2018). SAR is one of the most popular applications of RL in environmental monitoring. Providing effective solutions has great value in reducing human fatalities following disasters where traditional resources can become stretched. Research has been conducted into both indoor GPS-denied SAR and outdoor SAR with and without GPS localization. Most of these applications note the need for sophisticated intelligence in these agents as they often need to navigate previously unknown, dynamic, and cluttered environments and identify victims within them. A summary of RL-based exploration and SAR along with the relevant literature is provided in Tables 4 and 5 respectively.

It can be argued that due to the unpredictability of SAR environments, such systems should not solely rely on a good GPS signal and redundancy is required in their localization system. GPS-denied localization is often achieved through SLAM, especially in indoor environments. This uses a vision sensor like a camera or LiDAR to iteratively map the surroundings and localize the robot within them. Active SLAM refers to systems that use SLAM data to autonomously navigate their environment to facilitate more accurate measurements of the environment Placed et al. (2023). Active SLAM can also fit under the umbrella of active sensing

TABLE 5 Characteristics of reinforcement learning based search and rescue.

Characteristic	Description
Objective	Locate and assist or recover targets
State Representation	Environmental features, grid cells, target locations
Action Space	Movement actions, alarm/communicate, sensing actions
Reward Function	Rewards for finding targets, exploration, safe navigation
Observation	Sensor data for navigation or target detection (e.g., vision or LiDAR)
Challenges	Handling dynamic environments, long horizons, target detection
Citations	Zuluaga et al. (2018); Sampedro Pérez et al. (2019); Peake et al. (2020); Kulkarni et al. (2020); Ai et al. (2021)

and the Dec-POMDP discussed in [section 2.6](#). SLAM can be combined with RL agents to achieve active SLAM in cluttered environments useful for SAR applications. [Placed and Castellanos \(2020\)](#) use a trained DDQN and Dueling double deep Q Learning (D3QN) policy as the decision-making module in active SLAM. SAR missions would also typically involve some methods for detecting victims. In [Sampedro Pérez et al. \(2019\)](#) 2D LiDAR SLAM is used for localization while a DDPG agent is used to perform image-based visual servoing. This is where image features are used as control signals to direct the sensing agent towards a target. Both high-quality simulation and experimental results demonstrate that RL can be used in this system to produce effective solutions for image-based visual servoing in cluttered environments. This paper also demonstrates that RL systems can be robust to model mismatch between simulation and reality. The agents were trained largely through simulations but still performed well in real-world environments.

During SAR exploration it is important to make sure that all of the region of interest is searched. One popular companion of SLAM for indoor exploration is called ‘frontier exploration’. A frontier is defined as the border between mapped and unmapped areas that are not known to be an obstacle wall or boundary. The hypothesis is that agents can achieve complete exploration by visiting frontiers until there are none left. The standard method of choosing which frontiers to visit is either breadth-first (nearest frontiers) or depth-first (most distant frontiers). [Niroui et al. \(2019\)](#) proposes using an Asynchronous Advantage Actor-Critic (A3C) DRL agent as an alternative method of frontier selection to improve the operation time of a frontier exploration-based system. The agent aims to maximize the information gained along a robot’s path of exploration. Compared with several traditional exploration techniques implemented on a physical robot, the authors prove the superiority of their method in exploring the target area in a smaller total distance.

Cluttered environments also exist in outdoor applications and agents must also deal with potentially difficult weather conditions. An important consideration for the actual deployment of these algorithms is that different environments may demand different physical UAVs with varying payload capacities and different sensors. Robustness to such potential hardware changes is one of the key requirements in developing these algorithms. [Maciel-Pearson et al. \(2019\)](#) proposed an algorithm that can be used on any UAV with any camera for autonomous exploration of outdoor environments. They tested their algorithm in a simulated forest under different conditions. The DRL agent is trained directly to explore a region of interest and is penalized for collisions, revisiting areas, and leaving the search area. A double state input \mathcal{S} is utilized to learn collision avoidance from raw sensor data \mathcal{Z} , and complete exploration from navigational history and an environmental model. Outdoor environments are often difficult to deal with due to size. [Peake et al. \(2020\)](#) developed a SAR UAV to search a large area in a wilderness by splitting the area into unique regions on a quadrant tree. Each quadrant is assigned a probability distribution indicating that it currently contains a victim. A DDQN-trained policy is then used to select which of these quadrants should be visited next and generate an optimal flight trajectory across segments \mathcal{A} that maximizes information gain along the way. A second DRL agent trained using Advantage Actor-Critic (A2C) is used to allow the agent to explore the target quadrant more freely to find a potential victim via a continuous action space \mathcal{A}_{cont} . This combination of discrete and continuous action spaces is a clever way of negating the drawbacks of each option. The DDQN agent interacts with a large area. A common and effective strategy for exploring large areas while saving resources and ensuring convergence of the training is to discretize the area. This can lead to a loss of optimality for the true continuous environment. However, within the quadrant, which is a small cell of the full environment, a continuous action space is used which allows more dynamic movement and better exploration of the original continuous environment, thus reducing some disadvantages of a discretized environment.

Like all applications in EM, using multiple cooperative agents can improve exploration performance. The benefits of cooperation are especially true for SAR as these applications are time-critical. A common technique for allocating search areas is to split the target region into Voronoi partitions which can be either dynamic or static. [Hu et al. \(2020\)](#) combines dynamic Voronoi partitions for exploration with a DRL collision avoidance algorithm to allow agents to safely explore their partitions. The assignment of Voronoi partitions is an effective way to stop agents from colliding with each other, however, navigating in a cluttered environment with obstacles demands full online collision avoidance. DRL has proved to be an effective solution for using raw sensor data for collision avoidance [Long et al. \(2018\)](#); [Chen et al. \(2017\)](#).

As described in [Section 2.6](#), active sensing tasks mostly follow a generalized design philosophy of adaptive information gathering. Sensing decisions are made based on collected data and/or a belief model of the environment or target phenomenon [Chen F. et al. \(2020\)](#). [Viseras and Garcia \(2019\)](#) proposed an algorithm based on multi-agent DRL to make an algorithmic framework that can be applied to different information-gathering tasks without changing the underlying algorithm. To incorporate the needs of different applications both free and model-based approaches are

TABLE 6 Characteristics of reinforcement learning based source seeking and estimation.

Characteristic	Description
Objective	Locate the source of phenomenon
State Representation	Environmental features, source model grid cells
Action Space	Movement actions, sensing actions
Reward Function	Rewards for locating source
Observation	Intensity/Concentration values of environmental target
Challenges	Noisy and sparse information, dynamic sources, initial search phase
Citations	Duisterhof et al. (2021); Li et al. (2023); Viseras and Garcia. (2019)

proposed. The base algorithm, named Deep-IG, is a multi-agent extension of the A3C algorithm. The shared reward function is dual objective, simultaneously minimizing both the total mission time and the normalized root mean squared error in predicting the time-invariant physical process. Agents are also penalized for inter-agent collisions. The action space \mathcal{A} , is made up of discrete high-level robot movement commands. Deep-IG is shown to work across different physical platforms without changing algorithmic parameters. Experimental data is collected with 4 quadrotors for a terrain mapping application and ground robots are used for magnetic field mapping. Generalized reliable algorithms are valuable for EM applications as every deployment situation is different. Efforts such as this pave the way for widespread deployment as the bespoke algorithms and hardware designs for an intractable amount of potential monitoring applications are unrealistic, especially nowadays that climate change demands a quick answer.

3.2.4 Source seeking and boundary tracking

Source seeking is useful in many areas of EM, whether it is finding the location of a gas or radiation source, finding a wildfire in a forest, or finding a victim in a SAR scenario. In practice, the source concentrations can be very low and the data collected can be highly noisy, especially when far away from the source. In such cases, RL agents can become robust to high-noise environments with a sufficient number of training episodes. In general, concentration can be considered to be exponentially decaying as the distance from the sources increases. This means it is common for reward functions \mathcal{R} in source-seeking applications to be directly related to the observed concentration values \mathcal{Z} . Some systems are also concerned with estimating the properties of the target phenomenon whose source is being located, this is termed source term estimation (STE). Table 6 contains a summary of RL-based source-seeking algorithms and the relevant literature.

Source seeking via DRL has been demonstrated on a nano quadrotor with very highly constrained computational resources. Using a cheap commercially available and common processor, Duisterhof et al. (2021) demonstrated a source-seeking DRL agent

that is robust to high noise and shows consistent performance across several real and simulated environments. A neural network with only two hidden layers was trained to locate a source from concentration measurements. Systems such as these demonstrate a true potential for RL in EM. The main contribution of this work is how affordable and disposable the nano quadrotors are and their ability to operate in environments that may pose a risk to human life. This system uses a discrete action space \mathcal{A} , made up of high-level commands and a decoupled flight controller capable of translating these into actuator commands.

Li et al. (2023) propose an active RL algorithm. Active RL means there is information from the current state of the sensing process guiding the decisions and hence training of the agent. Here, the information metric is based on the maximum entropy sampling principle. This method combines both model-free and model-based approaches in RL into one algorithm based on recent promising results from similar approaches Nagabandi et al. (2018); Ostrovski et al. (2021). A logarithmic reward function is used to account for the exponential nature of concentration decay over physical space. In this system, they used an RL agent to optimize the dual objectives of the source seeking and source estimation; the agent must locate the source and maintain knowledge of the spatial concentration distribution. To achieve this, keeping a balance between the exploration of new areas and the exploitation of the previously collected information is necessary. The trade-off between exploration and exploitation is intrinsic to RL and active sensing alike. In RL specifically, this is manipulated by changing between future and immediate rewards using a discount factor γ defined for each agent. This trade-off is itself also a common optimization objective in time-constraint systems, where the agent becomes more exploitative as the mission progresses. The contribution of this work is the use of active exploration within RL which is shown to increase sample efficiency.

Gas source localization applications are often time sensitive as dangerous gasses have the potential to do harm quickly and good STE is required to coordinate an appropriate response. RL approaches for source seeking and estimation tend to direct agents by means of a discrete grid which can be limiting when applied to practical scenarios. Park et al. (2022) leverages DRL to find the optimal search path using a continuous action space. While the proposed action space does contain a continuous heading allowing for more diverse movement the limitations of a discrete action space are not totally elevated since the robot moves a fixed distance per step. For instance Zhang et al. (2023) combines a continuous heading with a continuous speed command. To allow for quick estimation of the gas distribution a particle filter is used and a Gaussian mixture model is used to extract features that make for suitable inputs to an instance of DDPG. They implement a gated recurrent unit (GRU) into their DNN that allows for memory of previous states and better STE. They verify the effectiveness of this choice by comparing the learning performance with and without the GRU.

Another area of research in which active EM can be applied is concerned with controlling environmental conditions in sensitive environments like artificial habitats or laboratories. While much of this research is concerned with novel control algorithms some have proposed using mobile sensors to complement this process don (Bing et al., 2019); Jin et al. (2018); Xiang et al. (2013). Another

example of where environmental conditions must be closely monitored is in a space habitat like that of the International Space Station. In [Guo et al. \(2021\)](#) they propose using moving sensors for temperature monitoring and fire detection in a space habitat called the Active Environmental Monitoring and Anomaly Search System. They implement a dynamic value iteration policy (DVI) to solve the problem which is modelled as a MDP. The performance of the DVI policy is measured against numerous other benchmarks. The DVI policy approaches the “jump policy” in anomaly detection time. This is a promising result since the implemented jump policy is not physically viable as it does not consider the continuous space a sensor would have to move through and assumes that it can “jump” to the next measurement position. The system is also extended to a distributed multi-agent system by implementing multiple signally trained agents. The authors point out that in confined and sensitive atmospheres it is important to consider how much benefit can be extracted by adding more sensors so that the optimal trade-off between anomaly detection time and number of sensors can be found. However, this has limited implications as MARL and cooperatively trained agents are not considered.

An interesting idea for using source seeking in an SAR application is presented in [Kulkarni et al. \(2020\)](#). A Q-learning RL agent is trained to locate victims via radio frequency signals of a device possessed by the victim. This, for example, could be a smartphone or a smartwatch. Using a discrete action space \mathcal{A} , the objective is to find the source or the victim, in the shortest possible time. The reward function \mathcal{R} , is based on the received signal strength (RSS) \mathcal{Z} , which has also been demonstrated for localisation in GPS-denied environments. The system is tested using ray tracing software to provide a better and more realistic simulation than simple numerical models. This helps the RL agent to perform upon deployment as it was trained in an environment more similar to the true environment.

In wildfire monitoring systems, manned aerial missions are often used to track the boundary and spread of the fire. This helps make decisions on where to use fire suppressants and where to remove fuel for fire containment. It can also be used to motivate and prepare evacuations if the fire is heading towards settlements. [Viseras et al. \(2021\)](#) proposed single and multi-agent DRL systems to locate and track the boundary of a wildfire for both multi-rotor and fixed-wing type UAVs. Two MARL approaches, i.e., multiple single-trained agents (MSTA) and a value decomposition network VDN, are proposed. In both cases, the agents are trained by a joint reward function \mathcal{R} . They found that both proposed algorithms outperform the benchmarks. More specifically, VDN tends to perform better towards the end of an episode when the fire behaviour is more complex and coordination between UAVs is more important, whereas MSTA provides a better performance early on when the main goal is finding the fire. They also found MSTA to be more stable and scales better when there are more than three agents. [Table 7](#) contains a summary of how RL boundary tracking algorithms are posed.

Boundary tracking can also be applied to UAV systems that are equipped with fire fighting capabilities, with the aim of containing the spread of a fire by applying suppressants at the boundary of the fire. This is the goal of the work in [Ostrovski et al. \(2021\)](#).

TABLE 7 Characteristics of reinforcement learning based boundary tracking.

Characteristic	Description
Objective	Track boundaries of phenomenon
State Representation	Environmental features, object boundaries
Action Space	Movement actions, sensing actions
Reward Function	Rewards for accurate boundary tracking, penalties for errors
Observation	Sensor data for measuring phenomenon (e.g., vision sensor or intensity/concentration measurements)
Challenges	Noisy or ambiguous boundaries, initial search phase
Citations	Julian and Kochenderfer, (2019); Viseras et al. (2021)

Here, a multi-agent fire-containing system of UAVs is trained using multi-agent DQN (MADQN) in which each agent acts independently but can learn from a pool of shared agent experience which helps to accelerate training. The authors encourage the cooperation of agents by removing any extra effect if agents apply fire suppressants to the same location. This research includes both sensing and communication constraints and a bespoke forest fire simulation. Since agents are limited to communicating with only their neighbours the system is decentralized and scalable which is very valuable for EM applications.

In the mentioned works that discuss boundary tracking, however, the simulations are entirely numerical and thus make a large number of assumptions about the operation environment. While the results in these papers are promising and further demonstrate the benefit of applying RL to active EM systems there is a large step between simulations like this and practical implementation. This is a trend across all the EM applications discussed but boundary tracking has a significant lack of practical results in the RL sphere.

3.3 Open challenges

Many of the system behaviours covered in this work are themselves well explored in the literature and the consensus is that autonomous systems are a big part of the future of EM. However, there is a shortage of research applying these behaviours to specific environmental domains, especially within the field of RL. Most of the existing efforts are simulation-based. These simulations are often greatly simplified versions of the true environments or focus on one specific element like a temperature field or wind model. There is potential for a large body of interesting research where robotic and RL-based EM systems are developed by making use of existing ecological simulations. Simulation is a large part of ecological research and collaboration between the two disciplines

can help accelerate the implementation of the important systems discussed in this paper. Simulations that utilize these advanced environmental models would be much closer to the true operation environment reducing model mismatch and accelerating work towards the practical domain. RL offers an easy first step to this unification as training is done offline and data can be collected beforehand. One example of early efforts in this space is given in [Liu and Bai \(2021\)](#) where a geographic information system (GIS) is used as the DRL training environment to provide the terrain data for a simulated UAV. This allows them to consider areas that may be obstructed from certain perspectives which are not considered in 2D terrain-based simulations. Besides, in EM there is a lack of practical testing and much more effort is required in the practical domain to start to build the road to actual implementations of these systems. Practical results are crucial for development and at the current stage, any practical testing offers immense value.

One of the biggest criticisms of RL is its sample inefficiency. As RL is better understood and solutions improve sample-efficiency will get better but effort is needed to address this directly. One promising area is the combination of model-based and model-free approaches [Pong et al. \(2020\)](#); [Chebotar et al. \(2017\)](#). This has been applied to EM applications in [Li et al. \(2023\)](#) to improve STE. These approaches aim to extract the benefits of both model-free RL that does not suffer from model mismatch and the learning efficiency of model-based approaches. Another way to improve sample efficiency is using long-term prediction [Tutsoy \(2022\)](#); [Doerr et al. \(2017\)](#). Long-term prediction involves estimating the future states of the environment, the rewards associated with those states, and how the agent's actions will influence the system over multiple time steps. Accurate long-term prediction allows agents to take more optimal actions earlier in the training process. It is especially useful when the consequences of actions are not immediately apparent.

Another open research area is the use of multi-agent reinforcement learning scenarios. The benefits of multi-agent systems are clear, especially for tasks like EM which can have such a large and diverse scope. It is apparent from the literature covered in this paper that most research in the RL for EM work is focused on single-agent implementations and the existing MARL approaches are rudimentary compared to the demands for a full operational system. There is an *extreme* lack of practical results for MARL despite promising simulation results and the clear potential of these systems. In light of these challenges, we offer a summary of some of the potential research areas that would help accelerate the development of systems capable of full-time deployment.

- Simulations that include more complete environmental models for specific environmental domains.
- MARL-based EM system research.
- Both single and multi-agent practical implementations at real sites of interest.

4 Real-world deployment challenges

There is a lack of practical application of these methodologies in the field of active EM. It is crucial that more practical testing is carried out and documented in the literature for fully autonomous

EM systems to be realized. As stated, the environments are often difficult to operate in and each specific environment will bring its own challenges. Furthermore, some of these systems will have to operate in multiple different environments. Thus the characteristics of these environments must be closely studied. In the literature, it is common to refer to environments as being either indoor or outdoor. Indoor environments are generally smaller, more cluttered and GPS-denied making navigation and localization challenging. On the other hand, outdoor environments are often much bigger, less sparsely populated, GPS enabled and have large variations due to changing light and weather conditions. That being said this does not hold for all outdoor environments, for example, active EM systems designed for operation in a forest will have to deal with unreliable GPS and a high density of potential obstacles which is more consistent with indoor environments. A simplified comparison of some environments of interest is given in [Table 8](#). One benefit of RL is that it can be trained under these considerations by utilizing multiple training environments and iterations of similar environments. It can learn to perform well under large variations and respond well under unknown conditions.

Another limiting factor for the practical application of active EM systems is the energy constraints of the platform. For example, UAVs have a very short flight time. Active EM systems will inevitably have to stop and refuel regularly. This requires safe and accessible locations to install charging points or places for the system to remain idle and charge passively via a utility such as on-board solar panels. In some environments, safe charging locations may be hard to find. For example, it is the case in volcanic monitoring that these charging stations will often have to be far away from the survey region as the areas close to the volcano are usually inaccessible and dangerous. This again highlights the need for such systems to have an optimal energy performance but also sheds light on the infrastructure needed for long-term practical application [Egerstedt et al. \(2018\)](#); [Notomista and Egerstedt \(2021\)](#). These environments may also pose challenges for maintenance and retrieval. In some environments, such as nuclear decommissioning, where the robot is in an environment that is potentially lethal to humans, retrieving the robot may be impossible. This means that the robots need to be very reliable and robust to the challenges of their environment. It also demands that they can communicate their findings without physical collection. For certain extreme environments like deep-sea or subterranean environments, wireless communication is a big challenge.

It is often the case that AUVs often periodically resurface to both recharge, communicate their data and connect with other agents. In underwater multi-agent systems like this, RL could be used to predict the position of other agents. This is a valuable corrective action as When the team is submerged it is unable to communicate and apply corrections until the communication is available again. Or RL has been used for processing acoustic signals which are a method of underwater communication [Rahmati et al. \(2019\)](#). MARL can also be used to teach agents how to cooperate without or with very limited communication [Zhang and Lesser \(2013\)](#). Since in RL, the training is done before deployment, MARL systems can take advantage of training agents based on other agents' training data that would not be available during run time. Having a good simulation environment can teach agents to cooperate without any communication when they are deployed.

TABLE 8 Comparison of environments where active EM may be applied.

	Rural	Urban	Lake/River	Deep-sea	Sub-terranean	Nuclear	Farming	Forest
GPS	Good	Medium	Good	None	None	None	Good	Bad
Scale	Large	Large	Predefined	Large	Medium	Small	Large	Large
Pre-defined Area	No	Maybe	Yes	No	Maybe	Yes	Yes	No
Weather Sensitivity	High	Medium	Medium	Medium	Low	None	High	Medium
Robot-Type	UAV, AGV	UAV	UAV, ASV, AUV	AUV	UAV, AGV	UAV, AGV	UAV, AGV	UAV, AGV
Commu-nication	Good	Good	Good	Bad	Bad	Good	Good	Medium
Obstacles	Few	Lots	Few	Few	Medium	Lots	Medium	Lots
Light	Variable	Variable	Variable	Dark	Dark	Light	Variable	Variable (Darker)
Charging Potential	Medium	Good	Good	Medium	Bad	Bad	Good	Bad

Underwater environments pose lots of unique challenges that may have simple solutions in other domains [Shruthi and Kavitha \(2022\)](#). Further to communication restrictions, the environment is also GPS-denied and non-variant, making localization extremely challenging. The uniform nature of the underwater environment also makes rewards very sparse and thus the learning efficiency and rates are often unfavourable. One option is to use a human trainer. This can either be using human-based examples to produce a reward or to have a human providing a reward based on good or bad behaviour. [Zhang et al. \(2020\)](#) proposed a DQN-based system that learns a path following control policy from a human-given reward and a pre-defined environmental reward. The use of human reward is helpful here as the underwater operation environment means that the environmental rewards can be highly sparse. This can ultimately lead to the system not converging to an optimal policy.

Certain survey areas of interest may be predefined before the mission starts, the size and shape of a lake, for example, while others may change constantly. For example, if we consider the problem of monitoring the plume of a volcano, the position of the plume with respect to the source, will change largely based on the strength and direction of the wind on a given day. It is a waste of time and energy resources to survey areas where the plume does not exist. This changing area of interest is common to many EM problems. The extreme case of this is when the survey area is completely unknown like in an urban SAR response to a natural disaster. Some environments where active EM is required are very isotropic and lack any clear landmarks, appearing very uniform to the agent. This is especially true for underwater, deep-sea applications. This is a big challenge for successful localization and navigation especially when GPS is also not an option. It is also a challenge for RL agents as the nature of the environment means that the rewards will be very sparse.

It is also the case that applying these systems in certain environments can be potentially dangerous to humans. For example, using UAVs in urban areas runs the risk of a malfunction causing injury to pedestrians or property damage. To remove or minimize

this risk in busy cities there should be no-fly zones. It may also be the case that these will change, based on the time of the day or current events. Some social challenges come along with some environments, especially highly populated, urban areas. For example, if autonomous ground vehicles were deployed for air-quality monitoring in cities, there is a potential for theft and vandalism. Thus, not only do they have to be safe for operation around the general public but they must have strong security features and be physically robust. Citizens may also dislike the idea of autonomous systems being used due to a lack of trust in the technology. Finding ways to introduce these systems in ways that are least offensive to the public is a real concern [Çetin et al. \(2022\)](#).

5 Conclusion

In this work, we have reviewed the application of RL and DRL to robotic active sensing in EM. The escalating climate crisis has highlighted the need for comprehensive EM solutions. Challenging, unknown and dynamic environments are ubiquitous to EM applications and intelligent and adaptive solutions are needed for autonomous and persistent monitoring that we can rely on. We have discussed RL as a method of encoding complex behaviours onto single and multiple agents for active EM. To provide a unifying framework we have utilized Dec-POMDPs to frame these systems as an active sensing pursuit and to unify them with traditional active sensing approaches. We discuss methods of representing environments as in EM there is more of a demand for model learning than in other areas of RL. The main body of this review is separated into common mission goals and agent behaviours that are useful in EM. We conclude this review by discussing and comparing the spectrum of the most widely used technical development platforms available for researchers in the hopes of streamlining further development and collaborating on a framework for standardization.

Author contributions

DM: Conceptualization, Methodology, Visualization, Investigation, Resources, Writing–original draft. AM: Conceptualization, Methodology, Visualization, Funding acquisition, Supervision, Validation, Writing–review and editing.

Funding

The author(s) declare that financial support was received for the research, authorship, and/or publication of this article. EPSRC Doctoral Training Fund to support cross-disciplinary research in FST.

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.

References

- Ai, B., Jia, M., Xu, H., Xu, J., Wen, Z., Li, B., et al. (2021). Coverage path planning for maritime search and rescue using reinforcement learning. *Ocean. Eng.* 241, 110098. doi:10.1016/j.oceaneng.2021.110098
- AlMahamid, F., and Grolinger, K. (2021). “Reinforcement learning algorithms: an overview and classification,” in 2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), ON, Canada, September, 2021. doi:10.1109/ccece53047.2021.9569056
- Alvear, O., Zema, N., Natalizio, E., and Calafate, C. (2017). Using uav-based systems to monitor air pollution in areas with poor accessibility. *J. Adv. Transp.* 2017, 1–14. doi:10.1155/2017/8204353
- Azar, A. T., Koubaa, A., Ali Mohamed, N., Ibrahim, H. A., Ibrahim, Z. F., Kazim, M., et al. (2021). Drone deep reinforcement learning: a review. *Electronics* 10, 999. doi:10.3390/electronics10090999
- Binas, J., Luginbuehl, L., and Bengio, Y. (2019). “Reinforcement learning for sustainable agriculture,” in *ICML 2019 workshop climate change: how can AI help*.
- Bing, D., Vishnu, P., Fan, F., and Zheng, O. N. (2019). A review of smart building sensing system for better indoor environment control. *Energy Build.* 199, 29–46. doi:10.1016/j.enbuild.2019.06.025
- Busoniu, L., Babuska, R., and De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.* 38, 156–172. doi:10.1109/TSMCC.2007.913919
- Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., et al. (2021). Multi-agent reinforcement learning: a review of challenges and applications. *Appl. Sci.* 11, 4948. doi:10.3390/app11114948
- Çetin, E., Cano, A., Deransy, R., Tres, S., and Barrado, C. (2022). Implementing mitigations for improving societal acceptance of urban air mobility. *Drones* 6, 28. doi:10.3390/drones6020028
- Chebota, Y., Hausman, K., Zhang, M., Sukhatme, G., Schaal, S., and Levine, S. (2017). Combining model-based and model-free updates for trajectory-centric reinforcement learning. <https://arxiv.org/abs/1703.03078>.
- Chen, F., Martin, J. D., Huang, Y., Wang, J., and Englot, B. (2020a). Autonomous exploration under uncertainty via deep reinforcement learning on graphs. <https://arxiv.org/abs/2007.12640>.
- Chen, J., Shu, T., Li, T., and de Silva, C. W. (2020b). Deep reinforced learning tree for spatiotemporal monitoring with mobile robotic wireless sensor networks. *IEEE Trans. Syst. Man, Cybern. Syst.* 50, 4197–4211. doi:10.1109/TSMC.2019.2920390
- Chen, Y. F., Liu, M., Everett, M., and How, J. P. (2017). “Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, May, 2017, 285–292. doi:10.1109/ICRA.2017.7989037
- Chen, Z., Peng, Z., Jiao, L., and Gui, Y. (2021). “Efficient multi-robot coverage of an unknown environment,” in 2021 40th Chinese Control Conference (CCC), Shanghai, China, July, 2021, 5166–5171. doi:10.23919/CCC52363.2021.9549546
- Choi, J., Lee, G., and Lee, C. (2021). Reinforcement learning-based dynamic obstacle avoidance and integration of path planning. *Intell. Serv. Robot.* 14, 663–677. doi:10.1007/s11370-021-00387-2
- Doerr, A., Daniel, C., Nguyen-Tuong, D., Marco, A., Schaal, S., Marc, T., et al. (2017). Optimizing long-term predictions for model-based policy search 78, 227–238.
- Duckett, T., Pearson, S., Blackmore, S., Grieve, B., Chen, W.-H., Cielniak, G., et al. (2018). Agricultural robotics: the future of robotic agriculture. <https://arxiv.org/abs/1806.06762>.
- Duisterhof, B. P., Krishnan, S., Cruz, J. J., Banbury, C. R., Fu, W., Faust, A., et al. (2021). Learning to seek: autonomous source seeking with deep reinforcement learning onboard a nano drone microcontroller. <https://arxiv.org/abs/1909.11236>.
- Egerstedt, M., Pauli, J. N., Notomista, G., and Hutchinson, S. (2018). Robot ecology: constraint-based control design for long duration autonomy. *Annu. Rev. Control* 46, 1–7. doi:10.1016/j.arcontrol.2018.09.006
- Faryadi, S., and Mohammadpour Velni, J. (2021). A reinforcement learning-based approach for modeling and coverage of an unknown field using a team of autonomous ground vehicles. *Int. J. Intelligent Syst.* 36, 1069–1084. doi:10.1002/int.22331
- Feiyu, Z., Dayan, L., Zhengxu, W., Jianlin, M., and Niya, W. (2024). Autonomous localized path planning algorithm for uavs based on td3 strategy. *Sci. Rep.* 14, 763. doi:10.1038/s41598-024-51349-4
- Gadd, C., Heinonen, M., Lähdesmäki, H., and Kaski, S. (2020). Sample-efficient reinforcement learning using deep Gaussian processes
- Guo, Y., Xu, Z., and Saleh, J. H. (2021). “Active sensing for space habitat environmental monitoring and anomaly detection,” in 2021 IEEE Aerospace Conference (50100), Big Sky, MT, USA, March, 2021. doi:10.1109/AERO50100.2021.9438412
- Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. <https://arxiv.org/abs/1801.01290>.
- Hu, J., Niu, H., Carrasco, J., Lennox, B., and Arvin, F. (2020). Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning. *IEEE Trans. Veh. Technol.* 69, 14413–14423. doi:10.1109/TVT.2020.3034800
- Hu, K., Li, H., Zhuang, J., Hao, Z., and Fan, Z. (2023). Efficient focus autoencoders for fast autonomous flight in intricate wild scenarios. *Drones* 7, 609. doi:10.3390/drones7100609
- James, M., Carr, B., D’Arcy, F., Diefenbach, A., Dietterich, H., Fornaciai, A., et al. (2020). Volcanological applications of unoccupied aircraft systems (uas): developments, strategies, and future challenges. *Volcanica* 3, 67–114. doi:10.30909/vol.03.01.67114

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frobt.2024.1336612/full#supplementary-material>

- Jang, B., Kim, M., Harerimana, G., and Kim, J. W. (2019). Q-learning algorithms: a comprehensive classification and applications. *IEEE Access* 7, 133653–133667. doi:10.1109/ACCESS.2019.2941229
- Jin, M., Liu, S., Schiavon, S., and Spanos, C. (2018). Automated mobile sensing: towards high-granularity agile indoor environmental quality monitoring. *Buill. Environ.* 127, 268–276. doi:10.1016/j.buillenv.2017.11.003
- Jin, Z., and Bertozzi, A. L. (2007). "Environmental boundary tracking and estimation using multiple autonomous vehicles," in 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, December, 2007, 4918–4923. doi:10.1109/CDC.2007.4434857
- Julian, K. D., and Kochenderfer, M. J. (2019). Distributed wildfire surveillance with autonomous aircraft using deep reinforcement learning. *J. Guid. Control, Dyn.* 42, 1768–1778. doi:10.2514/1.6004106
- Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., et al. (2020). Model-based reinforcement learning for atari. <https://arxiv.org/abs/1903.00374>.
- Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: a survey. *Int. J. Robotics Res.* 32, 1238–1274. doi:10.1177/0278364913495721
- Kouzeghar, M., Song, Y., Meghjani, M., and Bouffanais, R. (2023). Multi-target pursuit by a decentralized heterogeneous uav swarm using deep multi-agent reinforcement learning
- Kouzeghar, M., Meghjani, M., and Bouffanais, R. (2020). "Multi-agent reinforcement learning for dynamic ocean monitoring by a swarm of buoys," in *Global Oceans 2020* (Singapore: U.S. Gulf Coast), 1–8. doi:10.1109/IEEECONF38699.2020.9389128
- Kulkarni, S., Chaphekar, V., Uddin Chowdhury, M. M., Erden, F., and Guvenc, I. (2020). Uav aided search and rescue operation using reinforcement learning. *2020 S. 2*, 1–8. doi:10.1109/SoutheastCon44009.2020.9368285
- Lapeyrolerie, M., Chapman, M. S., Norman, K. E. A., and Boettiger, C. (2022). Deep reinforcement learning for conservation decisions. *Methods Ecol. Evol.* 13, 2649–2662. doi:10.1111/2041-210X.13954
- Larsen, T. N., Teigen, H. d., Laache, T., Varagnolo, D., and Rasheed, A. (2021). Comparing deep reinforcement learning algorithms' ability to safely navigate challenging waters. *Front. Robotics AI* 8, 738113. doi:10.3389/frobt.2021.738113
- Li, L., Wu, D., Huang, Y., and Yuan, Z.-M. (2021). A path planning strategy unified with a colregs collision avoidance function based on deep reinforcement learning and artificial potential field. *Appl. Ocean Res.* 113, 102759. doi:10.1016/j.apor.2021.102759
- Li, Z., Chen, W.-H., Yang, J., and Yan, Y. (2023). Aid-rl: active information-directed reinforcement learning for autonomous source seeking and estimation. *Neurocomputing* 544, 126281. doi:10.1016/j.neucom.2023.126281
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2019). Continuous control with deep reinforcement learning. <https://arxiv.org/abs/1509.02971>.
- Lin, C., Han, G., Tao, Q., Liu, L., Bilal Hussain Shah, S., Zhang, T., et al. (2023). Underwater equipotential line tracking based on self-attention embedded multiagent reinforcement learning toward auv-based its. *IEEE Trans. Intelligent Transp. Syst.* 24, 8580–8591. doi:10.1109/TITS.2022.3202225
- Littman, M. L. (1994). "Markov games as a framework for multi-agent reinforcement learning," in *Machine learning proceedings 1994*, Editors W. W. Cohen, and H. Hirsh (San Francisco, CA, USA: Morgan Kaufmann).
- Liu, S., and Bai, Y. (2021). Uav intelligent coverage navigation based on drl in complex geometrical environments. *Int. J. Comput. Intell. Syst.* 14, 177. doi:10.1007/s44196-021-00031-y
- Long, P., Fan, T., Liao, X., Liu, W., Zhang, H., and Pan, J. (2018). "Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning," in 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, May, 2018, 6252–6259. doi:10.1109/ICRA.2018.8461113
- Lu, H., Yang, Y., Tao, R., and Chen, Y. (2022). "Coverage path planning for sar-uav in search area coverage tasks based on deep reinforcement learning," in 2022 IEEE International Conference on Unmanned Systems (ICUS), Guangzhou, China, October, 2022, 248–253. doi:10.1109/ICUS55513.2022.9987002
- Luis, S. Y., Peralta Samaniego, F., Reina, D. G., and Toral Marín, S. (2021). "A sample-efficiency comparison between evolutionary algorithms and deep reinforcement learning for path planning in an environmental patrolling mission," in 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, June, 2021, 71–78. doi:10.1109/CEC45853.2021.9504864
- Luis, S. Y., Reina, D. G., and Marín, S. L. T. (2020). A deep reinforcement learning approach for the patrolling problem of water resources through autonomous surface vehicles: the ypacarai lake case. *IEEE Access* 8, 204076–204093. doi:10.1109/ACCESS.2020.3036938
- Maciel-Pearson, B. G., Marchegiani, L., Akcay, S., Atapour-Abarghouei, A., Garforth, J., and Breckon, T. P. (2019). Online deep reinforcement learning for autonomous uav navigation and exploration of outdoor environments. <https://arxiv.org/abs/1912.05684>.
- Mao, H., Gong, Z., and Xiao, Z. (2020). Reward design in cooperative multi-agent reinforcement learning for packet routing. <https://arxiv.org/abs/2003.03433>.
- Martin, P., Kwong, S., Smith, N., Yamashiki, Y., Payton, O., Russell-Pavier, F., et al. (2016). 3d unmanned aerial vehicle radiation mapping for assessing contaminant distribution and mobility. *Int. J. Appl. Earth Observation Geoinformation* 52, 12–19. doi:10.1016/j.jag.2016.05.007
- Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., et al. (2016). Asynchronous methods for deep reinforcement learning. <https://arxiv.org/abs/1602.01783>.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *nature* 518, 529–533. doi:10.1038/nature14236
- Nagabandi, A., Kahn, G., Fearing, R. S., and Levine, S. (2018). "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, May, 2018, 7559–7566. doi:10.1109/ICRA.2018.8463189
- Neumann, T., Ferrein, A., Kallweit, S., and Scholl, I. (2014). *Towards a mobile mapping robot for underground mines*.
- Niroui, F., Zhang, K., Kashino, Z., and Nejat, G. (2019). Deep reinforcement learning robot for search and rescue applications: exploration in unknown cluttered environments. *IEEE Robotics Automation Lett.* 4, 610–617. doi:10.1109/LRA.2019.2891991
- Notomista, G., and Egerstedt, M. (2021). Persistification of robotic tasks. *IEEE Trans. Control Syst. Technol.* 29, 756–767. doi:10.1109/TCST.2020.2978913
- Oroojlooy, A., and Hajinezhad, D. (2023). A review of cooperative multi-agent deep reinforcement learning. *Appl. Intell.* 53, 13677–13722. doi:10.1007/s10489-022-04105-y
- Ostrovski, G., Castro, P. S., and Dabney, W. (2021). The difficulty of passive learning in deep reinforcement learning. *Adv. Neural Inf. Process. Syst.* 28, 23283–23295. Cited by: 9. doi:10.48550/arXiv.2110.14020
- Padrao, P., Fuentes, J., Bobadilla, L., and Smith, R. N. (2022). Estimating spatio-temporal fields through reinforcement learning. *Front. Robotics AI* 9, 878246. doi:10.3389/frobt.2022.878246
- Park, M., Ladosz, P., and Oh, H. (2022). Source term estimation using deep reinforcement learning with Gaussian mixture model feature extraction for mobile sensors. *IEEE Robotics Automation Lett.* 7, 8323–8330. doi:10.1109/LRA.2022.3184787
- Peake, A., McCalmon, J., Zhang, Y., Raiford, B., and Alqahtani, S. (2020). "Wilderness search and rescue missions using deep reinforcement learning," in 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Abu Dhabi, United Arab Emirates, November, 2020, 102–107. doi:10.1109/SSRR50563.2020.9292613
- Peralta, F., Arzamendia, M., Gregor, D., Reina, D. G., and Toral, S. (2020). A comparison of local path planning techniques of autonomous surface vehicles for monitoring applications: the ypacarai lake case-study. *Sensors* 20, 1488. doi:10.3390/s20051488
- Pham, H. X., La, H. M., Feil-Seifer, D., and Nefian, A. (2018). Cooperative and distributed reinforcement learning of drones for field coverage. <https://arxiv.org/abs/1803.07250>.
- Piciarelli, C., and Foresti, G. L. (2019). "Drone patrolling with reinforcement learning," in Proceedings of the 13th International Conference on Distributed Smart Cameras, Trento, Italy, September, 2019. doi:10.1145/3349801.3349805
- Placed, J. A., and Castellanos, J. A. (2020). A deep reinforcement learning approach for active slam. *Appl. Sci.* 10, 8386. doi:10.3390/app10238386
- Placed, J. A., Strader, J., Carrillo, H., Atanasov, N., Indelman, V., Carlone, L., et al. (2023). A survey on active simultaneous localization and mapping: state of the art and new frontiers. <https://arxiv.org/abs/2207.00254>.
- Pong, V., Gu, S., Dalal, M., and Levine, S. (2020). Temporal difference models: model-free deep rl for model-based control. <https://arxiv.org/abs/1802.09081>.
- Popovic, M., Vidal-Calleja, T., Hitz, G., Chung, J. J., Sa, I., Siegwart, R., et al. (2020). An informative path planning framework for uav-based terrain monitoring. <https://arxiv.org/abs/1809.03870>.
- Rahmati, M., Nadeem, M., Sadhu, V., and Pompili, D. (2019). "UW-MARL: multi-agent reinforcement learning for underwater adaptive sampling using autonomous vehicles," in 19: Proceedings of the 14th International Conference on Underwater Networks and Systems, Atlanta, GA, USA, October, 2019. doi:10.1145/3366486.3366533
- Rückin, J., Jin, L., and Popović, M. (2022). "Adaptive informative path planning using deep reinforcement learning for uav-based active sensing," in 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, May, 2022.
- Sampedro Pérez, C., Rodríguez Ramos, A., Bavle, H., Carrio, A., de la Puente, P., and Campoy, P. (2019). A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. *J. Intelligent Robotic Syst.* 95, 1–27. doi:10.1007/s10846-018-0898-1

- Schulman, J., Levine, S., Moritz, P., Jordan, M. I., and Abbeel, P. (2017). Trust region policy optimization. <https://arxiv.org/abs/1502.05477>.
- Shruthi, K. R., and Kavitha, C. (2022). "Reinforcement learning-based approach for establishing energy-efficient routes in underwater sensor networks," in 2022 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, July, 2022, 1–6. doi:10.1109/CONECCT55679.2022.9865724
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., et al. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. <https://arxiv.org/abs/1712.01815>.
- Strobel, V. (2018). Pold87/academic-keyword-occurrence: first release. <https://zenodo.org/records/1218409>.
- Theile, M., Bayerlein, H., Nai, R., Gesbert, D., and Caccamo, M. (2020). "Uav coverage path planning under varying power constraints using deep reinforcement learning," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, October, 2020, 1444–1449. doi:10.1109/IROS45743.2020.9340934
- Tutsoy, O. (2022). Pharmacological, non-pharmacological policies and mutation: an artificial intelligence based multi-dimensional policy making algorithm for controlling the casualties of the pandemic diseases. *IEEE Trans. Pattern Analysis Mach. Intell.* 44, 9477–9488. doi:10.1109/TPAMI.2021.3127674
- Viseras, A., and Garcia, R. (2019). Deepig: multi-robot information gathering with deep reinforcement learning. *IEEE Robotics Automation Lett.* 4, 3059–3066. doi:10.1109/LRA.2019.2924839
- Viseras, A., Meissner, M., and Marchal, J. (2021). Wildfire front monitoring with multiple uavs using deep q-learning. *IEEE Access*, 1. doi:10.1109/ACCESS.2021.3055651
- Waharte, S., and Trigoni, N. (2010). "Supporting search and rescue operations with uavs," in 2010 International Conference on Emerging Security Technologies, Canterbury, UK, September, 2010, 142–147. doi:10.1109/EST.2010.31
- Wang, J., Zhao, Z., Qu, J., and Chen, X. (2024a). Appa-3d: an autonomous 3d path planning algorithm for uavs in unknown complex environments. *Sci. Rep.* 14, 1231. doi:10.1038/s41598-024-51286-2
- Wang, Y., Lu, C., Wu, P., and Zhang, X. (2024b). Path planning for unmanned surface vehicle based on improved q-learning algorithm. *Ocean. Eng.* 292, 116510. doi:10.1016/j.oceaneng.2023.116510
- Watkins, C. J., and Dayan, P. (1992). Q-learning. *Mach. Learn.* 8, 279–292. doi:10.1023/a:1022676722315
- Wen, N., Long, Y., Zhang, R., Liu, G., Wan, W., and Jiao, D. (2023). Colregs-based path planning for usvs using the deep reinforcement learning strategy. *J. Mar. Sci. Eng.* 11, 2334. doi:10.3390/jmse11122334
- Woo, J., and Kim, N. (2020). Collision avoidance for an unmanned surface vehicle using deep reinforcement learning. *Ocean. Eng.* 199, 107001. doi:10.1016/j.oceaneng.2020.107001
- Wu, J., Cheng, L., Chu, S., and Song, Y. (2024). An autonomous coverage path planning algorithm for maritime search and rescue of persons-in-water based on deep reinforcement learning. *Ocean. Eng.* 291, 116403. doi:10.1016/j.oceaneng.2023.116403
- Xiang, Y., Piedrahita, R., Dick, R. P., Hannigan, M., Lv, Q., and Shang, L. (2013). "A hybrid sensor system for indoor air quality monitoring," in 2013 IEEE International Conference on Distributed Computing in Sensor Systems, Cambridge, MA, USA, May, 2013. doi:10.1109/DCOSS.2013.48
- Yanes Luis, S., Perales, M., Gutiérrez, D., and Toral, S. (2022). *Deep reinforcement learning applied to multi-agent informative path planning in environmental missions*. Berlin, Germany: Springer. doi:10.1007/978-3-031-26564-8_2
- Yang, S. C.-H., Wolpert, D. M., and Lengyel, M. (2016). Theoretical perspectives on active sensing. *Curr. Opin. Behav. Sci.* 11, 100–108. doi:10.1016/j.cobeha.2016.06.009
- Zhang, C., and Lesser, V. (2013). "Coordinating multi-agent reinforcement learning with limited communication," in Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems (Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems), AAMAS '13, St. Paul MN USA, May, 2013, 1101–1108.
- Zhang, J., Liu, Y., and Zhou, W. (2023). Adaptive sampling path planning for a 3d marine observation platform based on evolutionary deep reinforcement learning. *J. Mar. Sci. Eng.* 11, 2313. doi:10.3390/jmse11122313
- Zhang, Q., Lin, J., Sha, Q., He, B., and Li, G. (2020). Deep interactive reinforcement learning for path following of autonomous underwater vehicle. <https://arxiv.org/abs/2001.03359>.
- Zhao, D., Huanshi, X., and Xun, Z. (2024). Active exploration deep reinforcement learning for continuous action space with forward prediction. *Int. J. Comput. Intell. Syst.* 17, 6. doi:10.1007/s44196-023-00389-1
- Zhu, C., Dastani, M., and Wang, S. (2022). A survey of multi-agent reinforcement learning with communication. <https://arxiv.org/abs/2203.08975>.
- Zuluaga, J. G. C., Leidig, J. P., Trefftz, C., and Wolffe, G. (2018). "Deep reinforcement learning for autonomous search and rescue," in NAECON 2018 - IEEE National Aerospace and Electronics Conference, Dayton, OH, USA, July, 2018, 521–524. doi:10.1109/NAECON.2018.8556642



OPEN ACCESS

EDITED BY

Holger Voos,
University of Luxembourg, Luxembourg

REVIEWED BY

Önder Tutsoy,
Adana Science and Technology University,
Türkiye
Xuejing Lan,
Guangzhou University, China

*CORRESPONDENCE

Nargess Sadeghzadeh-Nokhodberiz,
✉ sadeghzadeh@qut.ac.ir
Allahyar Montazeri,
✉ a.montazeri@lancaster.ac.uk

RECEIVED 13 January 2024

ACCEPTED 23 May 2024

PUBLISHED 15 July 2024

CITATION

Sadeghzadeh-Nokhodberiz N, Sadeghi MR,
Barzamini R and Montazeri A (2024),
Distributed safe formation tracking control of
multiquadcopter systems using barrier
Lyapunov function.
Front. Robot. AI 11:1370104.
doi: 10.3389/frobt.2024.1370104

COPYRIGHT

© 2024 Sadeghzadeh-Nokhodberiz, Sadeghi,
Barzamini and Montazeri. This is an
open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). 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.

Distributed safe formation tracking control of multiquadcopter systems using barrier Lyapunov function

Nargess Sadeghzadeh-Nokhodberiz^{1*},
Mohammad Reza Sadeghi¹, Rohollah Barzamini² and
Allahyar Montazeri^{3*}

¹Department of Control Engineering, Qom University of Technology, Qom, Iran, ²Department of Electrical Engineering, Islamic Azad University Tehran Central Branch, Tehran, Iran, ³School of Engineering, Lancaster University, Lancaster, United Kingdom

Coordinating the movements of a robotic fleet using consensus-based techniques is an important problem in achieving the desired goal of a specific task. Although most available techniques developed for consensus-based control ignore the collision of robots in the transient phase, they are either computationally expensive or cannot be applied in environments with dynamic obstacles. Therefore, we propose a new distributed collision-free formation tracking control scheme for multiquadcopter systems by exploiting the properties of the barrier Lyapunov function (BLF). Accordingly, the problem is formulated in a backstepping setting, and a distributed control law that guarantees collision-free formation tracking of the quads is derived. In other words, the problems of both tracking and interagent collision avoidance with a predefined accuracy are formulated using the proposed BLF for position subsystems, and the controllers are designed through augmentation of a quadratic Lyapunov function. Owing to the underactuated nature of the quadcopter system, virtual control inputs are considered for the translational (x and y axes) subsystems that are then used to generate the desired values for the roll and pitch angles for the attitude control subsystem. This provides a hierarchical controller structure for each quadcopter. The attitude controller is designed for each quadcopter locally by taking into account a predetermined error limit by another BLF. Finally, simulation results from the MATLAB-Simulink environment are provided to show the accuracy of the proposed method. A numerical comparison with an optimization-based technique is also provided to prove the superiority of the proposed method in terms of the computational cost, steady-state error, and response time.

KEYWORDS

multiagent systems, formation control, intervehicle collision avoidance, barrier Lyapunov function (BLF), formation tracking control, backstepping controller

1 Introduction

Quadcopters are one of the most important categories of multirotor drones and consist of four arms, four motors, and four propellers. The control and navigation of quadcopters in a single or cooperative form have been the subject of various

studies to enhance their capabilities for various applications (Montazeri et al., 2021; Sadeghzadeh-Nokhodberiz et al., 2023, 2021). A multi-quadcopter system is a form of multi-agent system that is used in various extreme environment applications, including nuclear decommissioning (Martin et al., 2016; Allahyar and Koubaa, 2023), volcanology (James et al., 2020), wildfire monitoring (Julian and Kochenderfer, 2019), and underground mining (Neumann et al., 2014). A multi-agent system consists of several interacting intelligent agents that can cooperate their movements, sensing, and computations to achieve a common goal. Multi-quadcopter systems are ideal solutions for different challenges imposed by humans working in extreme environments (Burrell et al., 2018); for example, using multiple quadcopters improves the performances, time and energy efficiencies, coverage areas, and redundancies of multiple robots performing the same task (Mansfield and Montazeri, 2024).

One of the most important issues in controlling a multi-agent system is formation control to achieve consensus. Formation control is an important consideration in coordinating the control of a group of unmanned robots or quadcopters in the present study. It is assumed that each drone can fly and share information with the other robots in its neighborhood. Formation control is used in many applications relevant to environmental monitoring, such as coverage, patrolling, autonomous exploration, search and rescue, source seeking, and boundary tracking (Mansfield and Montazeri, 2024). In Liu and Bucknall (2018), the problem of formation control and cooperative motion planning of multiple unmanned vehicles was investigated and various approaches were reviewed; this work provides good insights into the challenges and techniques available for cooperative path planning and formation control.

One of the most investigated techniques to address the formation control problem is consensus-based formation control (Peng et al., 2020; Patil and Shah, 2021). Consensus is a displacement-based control mechanism, meaning that the agents simply need to know the relative locations (displacements) of their neighbors in a local reference system linked to a global system to achieve the desired formation. Displacement-based formation control is typically divided into three primary strategies: virtual structure (VS), behavior based (BB), and leader-follower (LF). The basic idea of consensus is that each vehicle updates its information state based on the information states of its local (time-varying) neighbors such that the final information state of each vehicle converges to a common value. The main purpose of a distributed formation control technique is to derive appropriate control commands for each agent based on the information provided by the agents that are only in the neighborhood of that agent. Here, the aim is that the team of robots should maintain a specific geometric shape while closely tracking the desired trajectory defined for the leader in the LF configuration or virtual leader in the formation control setting (Can et al., 2022; Imran and Montazeri, 2022). In such scenarios, the desired trajectory of each robot in a robotic fleet is not defined separately; instead, the trajectory should be defined, for example, for the center of the quad formation shape, under the connectivity assumption of the system graph that all agents can coordinate with the leader. Although this is a fully decentralized configuration, less centralized scenarios have also been reported in literature (Lizzio et al., 2022), in which the navigation was carried out at the ground control station and the desired trajectory was then

transmitted to each drone that communicates with the neighboring agents to share their position and run the distributed on-board control algorithm to attain the desired trajectories. We adopted one such configuration in our investigation in this work.

The basic form of a formation control algorithm does not take into account the possibility of agent collisions while the agents attempt to reach their intended positions. For this reason, formation approaches considering interagent collision and/or obstacle avoidance have been the subject of investigations by some researchers. A comprehensive review comparing various collision avoidance strategies for unmanned aerial vehicle (UAV) applications can be found in Yasin et al. (2020). In the context of consensus-based formation control, similar collision avoidance strategies were surveyed and discussed by Sadeghzadeh-Nokhodberiz et al. (2023). When two drones generate a formation, they may collide with each other and obstacles in the transient phase as well as when reaching their desired positions and orientations. The consensus-based collision-free methods reviewed in both Lizzio et al. (2022) and Sadeghzadeh-Nokhodberiz et al. (2023) can be grouped under four main categories: (i) optimization-based techniques, (ii) force-field or artificial potential field (APF) techniques, (iii) geometric approaches, and (iv) sense-and-avoid approaches. As reviewed in Sadeghzadeh-Nokhodberiz et al. (2023), each of these approaches has its advantages/disadvantages for application to real life. Operationally speaking, static situations wherein obstacles are known ahead of time or are picked up on by the entire formation are more suited for optimization-based techniques. Although the force-field and geometric approaches are more effective for handling dynamic settings, the former may result in local minima due to cancellation of several APFs. Furthermore, the computational burden of a geometric technique in a busy dynamic environment may be very high when computing the collision-free trajectories. In terms of the operational requirements, optimization-based techniques are typically utilized in situations when the swarm must adhere to a predetermined reference trajectory. Depending on the algorithm chosen, each drone in the swarm either has a preloaded path or is aware of all the other reference trajectories. Nonetheless, the force-field and geometric techniques typically depend more on relative sensing or interagent communications.

Among the APF approaches reported for collision-free formation control, the method proposed by Yan et al. (2017) is notable because it causes the control signal to be limited and affected by the type of the potential field. Liang et al. (2020) studied a network of swarm drones, in which they followed a collision-free path by considering the system uncertainty in the presence of network constraints; the APF method was adopted here to address possible collisions between the UAVs, leading to a limited control signal. An example of an optimization-based technique used to design a collision-free formation control was reported by Kuriki and Namerikawa (2015); here, the problem was studied through the design of a consensus-based model predictive controller (MPC) by assuming that each UAV was located in a safe space and that the control input was updated as needed. The asymptotic stability of the proposed control method was also studied in detail. However, this method relied on the linear model of the system where the control system fails if the communication with the leader fails. Moreover, the collision avoidance strategy was considered only in the vertical direction. Jin et al. (2021) proposed a new framework to address the

formation control of multiple robots; here, two types of problems were studied, namely the performance issues as well as feasibility of implementing the constraints when their requirements were in the tracking errors and distances between the paths.

Recently, reinforcement learning (RL) and deep reinforcement learning (DRL) techniques have been proven to be effective for decision-making and operation of cooperative robots in complex environments under time-varying and uncertain conditions. For example, [Mansfield and Montazeri \(2024\)](#) reviewed different multiagent RL (MARL) techniques used as advanced tools in the design of optimal cooperative trajectories for multirobot systems in environmental monitoring applications by optimizing not only the individual rewards of each of the robots but also their collective reward; although the focus of this was control and not desired trajectory design for quads operating in uncertain and complex environments, it was assumed that the target trajectory of each robot was designed and made available using the techniques of [Liu and Bucknall \(2018\)](#) or [Mansfield and Montazeri \(2024\)](#). Further, as mentioned in [Mansfield and Montazeri \(2024\)](#), the RL technique can be used to avoid interagent collisions and obstacles.

The above works do not use the barrier Lyapunov function (BLF) as an effective tool for collision-free formation tracking of quadcopters. More recently, [Sadeghzadeh-Nokhodberiz and Meskin \(2023\)](#) presented the problem of consensus-based formation tracking of multiquadcopter systems using logarithmic BLFs; however, the problem of collision avoidance was not considered. Instead, the method involved the use of a centralized approach that was then transformed to distributed control using highpass consensus filters. Although the performance of the proposed distributed method asymptotically converged to that of the centralized one, the convergence time was rather large. Therefore, the problem of collision-free formation tracking control of multiquadcopter systems is derived from scratch in a distributed manner in the present work.

Generally, the BLF is used to prevent the states from violating the constraints. Therefore, the BLF can be used to ensure safety and collision avoidance while guaranteeing convergence with a predefined accuracy. The BLF is a positive-definite function that grows to infinity when its arguments approach certain limits. [Kumar and Kumar \(2022\)](#) discussed the three-dimensional trajectory tracking problem of an unmanned vehicle with restrictions on the flight path during operations; to ensure that the quadrotor followed the desired trajectory while satisfying the imposed motion constraints, a BLF approach was proposed. Moreover, a six-degrees-of-freedom dynamic model of the system was considered to achieve high-accuracy tracking performance; this controller could avoid singularities in the attitude subsystem. [Tang et al. \(2013\)](#) proposed a single-input single-output non-linear control system using the BLF to avoid deviating from the safety range. [Tee and Ge \(2011\)](#) presented a feedback control system design with constraints on the states. [Chen et al. \(2020\)](#) studied the problem of obstacle avoidance for a system with multiple agents avoiding obstacles in the environment; in this method, a hybrid decentralized monitoring controller that guarantees collision avoidance was proposed. The method is scalable and can be applied to general non-linear robot dynamics. Recently, advanced model-based and uncertain optimal control laws have been developed and implemented in real time for impaired UAVs ([Ahmadi et al., 2023](#)). [Ganguly \(2022\)](#) used the

BLF technique to design a controller for an N-degrees-of-freedom Euler–Lagrange system and numerically evaluated its effectiveness; this method was recently used for multirobot applications for interagent collision avoidance and tracking using second-order kinematics in two-dimensional cases ([Jin et al., 2021](#)). It is worth mentioning that [Khadhraoui et al. \(2023\)](#) and [Mughees and Ahmad \(2023\)](#) used BLFs for single quadcopter systems, in addition to [Sadeghzadeh-Nokhodberiz and Meskin \(2023\)](#), who recently employed BLFs for formation tracking of multiquadcopter systems without considering the collision avoidance problem.

Based on the above literature, a decentralized (distributed) collision-free formation tracking control method is proposed in this work for cooperative control of quadcopter systems. The proposed method is used for interagent collision avoidance and trajectory tracking with a predefined accuracy. Compared to the aforementioned works, the proposed method has a lower computational burden, is easily scalable, and can be used in dynamic environments. Further, contrary to the APF approaches, the proposed method does not limit the control signal for collision avoidance. The major contribution of the present study is that the barrier Lyapunov method is used to derive a distributed collision-free formation tracking control in which both formation tracking and interagent collision avoidance are considered simultaneously. Accordingly, BLFs are first proposed for the position subsystems (x , y , and z axes) and controllers are designed by augmenting a quadratic Lyapunov function, leading to a backstepping procedure. Owing to the underactuated nature of the quadcopter system, virtual inputs are considered for the translational (x and y axes) subsystems that are then used to generate the desired values for the roll and pitch angles for the attitude control subsystem. This provides a hierarchical controller structure for each quadcopter.

The distributed formation tracking controller derived herein not only guarantees convergence of the formation tracking error with a predefined accuracy but also avoids interagent collisions during the transient responses of the formation. Thus, both collision avoidance and trajectory tracking with a predefined bound on the tracking error are achieved in a distributed manner. The novelty of this work is briefly summarized as follows:

- Formulating multiple problems, including trajectory tracking, formation tracking control, and interagent collision avoidance, of a multiquadcopter system using the proposed BLF.
- Deriving decentralized (distributed) hierarchical control laws for collision-free formation tracking control of the altitude as well as translational x and y axes subsystems using virtual inputs in a backstepping framework.
- Designing attitude control laws separately for each agent using desired signals generated via BLFs while considering a predefined accuracy.

The remainder of this article is organized as follows. [Section 2](#) details the problem formulation and preliminaries. [Section 3](#) presents a decentralized collision-free formation tracking controller design for a multiquadcopter system using BLFs. [Section 4](#) presents the simulation results, and [Section 5](#) contains a summary of the conclusions.

2 Preliminaries and problem formulation

This section presents some preliminaries on the required theoretical materials.

2.1 Graph theory

Consider the graph $G = \{V, \xi, W\}$ containing N nodes, where $V = \{1, 2, \dots, N\}$ is the set of nodes and ξ is the set of all the edges of the graph. It is assumed that the edge (i, j) between nodes i and j exists, where i and j are adjacent to each other, such that $\xi = (i, j) \in V \times V$. If $(i, j) \in \xi \Leftrightarrow (j, i) \in \xi$, then the graph is undirected. Matrix $A = [a_{ij}]$ is the adjacency matrix such that if there is a path from i to j in the system graph, then $a_{ij} = a_{ji} = 1$. A path from i to j is a sequence of distinct nodes starting at i and ending at j , such that each pair of consecutive nodes is adjacent. If there is a path from i to j , then the nodes are connected. If all the paths of a graph are connected, then the graph is connected. The degree matrix of a graph D is a diagonal matrix with elements d_i that are equal to the set of neighboring nodes. $N_i = \{j \in V: (i, j) \in \xi\}$, where N_i is the set of neighbors surrounding i . The matrix L is the Laplacian matrix of the graph that is equal to $L = D - A$, and the sum of its rows is equal to zero (Hu et al., 2021).

2.2 Barrier Lyapunov theory

Consider the non-linear system given by Eq. (1) as follows:

$$\begin{aligned}\dot{x}_1(t) &= f_1(x_1(t)) + g_1(x_1(t))x_2(t), \\ \dot{x}_2(t) &= f_2(x_1(t), x_2(t)) + g_2(x_1(t), x_2(t))u(t),\end{aligned}\quad (1)$$

where $x_1(t) \in \mathbb{R}^{n_1}$ and $x_2(t) \in \mathbb{R}^{n_2}$ are system states, $u(t) \in \mathbb{R}^{n_2}$ is the system input, and vector functions f_1, f_2, g_1 , and g_2 are assumed to be smooth. The goal here is to design the control law $u(t)$ such that $x_1(t)$ follows the desired trajectory $x_{1d}(t)$ with a predefined accuracy. In other words, if $e_1(t) = x_1(t) - x_{1d}(t)$, the control objective is to ensure that the tracking error remains within a compact set defined by $D_1 = \{e_1(t) \in \mathbb{R}^{n_1} | \|e_1(t)\| < \Omega_1, t \geq 0\}$, where Ω_1 is a predefined positive scalar. Next, the idea of using the BLF in a backstepping procedure (Tee et al., 2009; Ngo et al., 2005; Tee et al., 2008) is extended to the vector form case. Therefore, assuming that the BLF $V_1(t)$ is defined as in Eq. (2), the Lyapunov candidate function $V(t)$ is defined by augmenting $V_2(t)$ to $V_1(t)$ as follows:

$$\begin{aligned}V_1(t) &= \frac{1}{2}\eta_1^2(t), \\ V(t) &= V_1(t) + V_2(t),\end{aligned}\quad (2)$$

where $\eta_1(t) = \frac{\Omega_1 d_1(t)}{\Omega_1 - d_1(t)}$ and $V_2 = \frac{1}{2}z_2^T z_2$ with $z_2(t) = x_2(t) - \alpha(t)$ is defined as an auxiliary tracking error for the virtual control input; $\alpha(t)$ is a stabilizing vector function that must be designed. According to Tee et al. (2009) and Lemma 1 therein, if the inequality $\dot{V}(t) \leq 0$ holds $\forall t \geq 0$, it can be concluded that $e_1(t) \in D_1$ if $e_1(0) \in D_1$.

2.3 Quadcopter model

Assume we have a group of quadcopters consisting of N agents communicating with each other. The dynamic of the attitude subsystem of the i th quadcopter (assuming a small Euler angle) for $i = 1, \dots, N$ can be written as follows:

$$\begin{aligned}\ddot{\phi}_i(t) &= \frac{I_{yy_i} - I_{zz_i}}{I_{xx_i}} \dot{\theta}_i(t) \dot{\psi}_i(t) - I_{r_i} \Omega_{r_i} \frac{\dot{\theta}_i(t)}{I_{xx_i}} + \frac{u_{2i}(t)}{I_{xx_i}}, \\ \ddot{\theta}_i(t) &= \frac{I_{zz_i} - I_{xx_i}}{I_{yy_i}} \dot{\phi}_i(t) \dot{\psi}_i(t) + I_{r_i} \Omega_{r_i} \frac{\dot{\phi}_i(t)}{I_{yy_i}} + \frac{u_{3i}(t)}{I_{yy_i}}, \\ \ddot{\psi}_i(t) &= \frac{I_{zz_i} - I_{xx_i}}{I_{yy_i}} \dot{\phi}_i(t) \dot{\theta}_i(t) + \frac{u_{4i}(t)}{I_{zz_i}}.\end{aligned}\quad (3)$$

where the roll angle $\phi_i(t)$, pitch angle $\theta_i(t)$, and yaw angle $\psi_i(t)$ represent the rotations about the x, y , and z axes in the inertial frame, respectively. The input signals $u_{2i}(t)$, $u_{3i}(t)$, and $u_{4i}(t)$ represent torques in the corresponding directions for the i th quadcopter in the body frame. I_{xx_i} , I_{yy_i} , and I_{zz_i} are the inertia tensors, and I_{r_i} is the inertia of the propellers. Further, Ω_{r_i} describes the relative speed of the propeller.

The translational dynamics of the i th quadcopter can be presented as follows:

$$\begin{aligned}\ddot{x}_i(t) &= \frac{u_{1i}(t)}{m_i} (\cos(\psi_i(t)) \sin(\theta_i(t)) \cos(\phi_i(t)) + \sin(\psi_i(t)) \sin(\phi_i(t))), \\ \ddot{y}_i(t) &= \frac{u_{1i}(t)}{m_i} (\sin(\psi_i(t)) \sin(\theta_i(t)) \cos(\phi_i(t)) - \cos(\psi_i(t)) \sin(\phi_i(t))), \\ \ddot{z}_i(t) &= -g + \frac{u_{1i}(t)}{m_i} \cos(\phi_i(t)) \cos(\theta_i(t)),\end{aligned}\quad (4)$$

where $[x_i(t) \ y_i(t) \ z_i(t)]^T$ represents the position of the i th quadcopter in the inertial frame, $u_{1i}(t)$ defines the main thrust created by the combined forces of the rotors, g is the gravitational constant, and m_i refers to the mass of the i th quadcopter (Sadeghzadeh-Nokhodberiz et al., 2021).

The above dynamic system can be represented in the state-space form, and the system is divided into three subsystems for simplicity as altitude, translational, and attitude subsystems (Sadeghzadeh-Nokhodberiz et al., 2021).

The altitude subsystem can be decomposed as follows:

$$\begin{aligned}\dot{x}_{1i}(t) &= x_{2i}(t), \\ \dot{x}_{2i}(t) &= -g + g_{2i}(t)u_{1i}(t),\end{aligned}\quad (5)$$

where $x_{1i}(t) \equiv z_i(t)$ and $x_{2i}(t) \equiv \dot{z}_i(t)$ refer to the altitude and velocity of the i th quadcopter in the z direction, respectively; $u_{1i}(t)$ is the control input indicating the thrust force applied to the i th quadcopter in the z direction; $g_{2i}(t) = \frac{1}{m_i} \cos(\theta_i(t)) \cos(\phi_i(t))$ is an auxiliary variable defined to convert the last expression in Eq. (4) to a more compact form.

The translational subsystem is defined as follows:

$$\begin{aligned}\dot{x}_{3i}(t) &= x_{4i}(t), \\ \dot{x}_{4i}(t) &= g_{4i}(t)u_{iv3}(t), \\ \dot{x}_{5i}(t) &= x_{6i}(t), \\ \dot{x}_{6i}(t) &= g_{6i}(t)u_{iv5}(t),\end{aligned}\quad (6)$$

where $x_{3i}(t) \equiv x_i(t)$ and $x_{4i}(t) \equiv \dot{x}_i(t)$ refer to the position and velocity of the i th quadcopter in the x direction, and $x_{5i}(t) \equiv y_i(t)$ and $x_{6i}(t) \equiv \dot{y}_i(t)$ refer to the position and velocity of the i th quadcopter in the y direction, respectively; $g_{4i}(t) = g_{6i}(t) = \frac{u_{iv}(t)}{m_i}$ are auxiliary variables defined to convert the last expression in Eq. (4) to a more compact form. Moreover, $u_{iv3}(t)$ and $u_{iv5}(t)$ are virtual controller inputs to enable control of the underactuated position subsystem and are defined as follows:

$$u_{iv3}(t) = \cos(\psi_i(t)) \sin(\theta_i(t)) \cos(\phi_i(t)) + \sin(\psi_i(t)) \cos(\phi_i(t)) \quad (7)$$

$$u_{iv5}(t) = \sin(\psi_i(t)) \sin(\theta_i(t)) \cos(\phi_i(t)) - \cos(\psi_i(t)) \sin(\phi_i(t)) \quad (8)$$

Finally, the attitude subsystem can be defined using Eq. (3) by assuming that I_{r_i} is very small:

$$\begin{aligned} \dot{x}_{7i}(t) &= x_{8i}(t), \\ \dot{x}_{8i}(t) &= f_{2i}(t) + G_{8i}u_i(t), \end{aligned} \quad (9)$$

where $x_{7i}(t) \equiv [\phi_i(t) \ \theta_i(t) \ \psi_i(t)]^T$ and $x_{8i}(t) \equiv [\dot{\phi}_i(t) \ \dot{\theta}_i(t) \ \dot{\psi}_i(t)]^T$ are the respective attitude and angular velocity vectors in the inertial frame; $u_i(t) = [u_{2i}(t) \ u_{3i}(t) \ u_{4i}(t)]^T$ is the control input vector including the torques in the corresponding directions for the i th quadcopter in the body frame; $f_{2i}(t) = [a_{1i}\dot{\phi}_i(t)\dot{\psi}_i(t) \ a_{3i}\dot{\phi}_i(t)\dot{\psi}_i(t) \ a_{5i}\dot{\phi}_i(t)\dot{\theta}_i(t)]^T$ is an auxiliary vector with auxiliary variables defined by $a_{1i} = \frac{I_{yyi} - I_{xxi}}{I_{xxi}}$,

$$a_{3i} = \frac{I_{zzj} - I_{xxj}}{I_{yyj}}, \text{ and } a_{5i} = \frac{I_{zzj} - I_{xxj}}{I_{yyj}}; G_{8i} = \begin{bmatrix} b_{1i} & 0 & 0 \\ 0 & b_{3i} & 0 \\ 0 & 0 & b_{5i} \end{bmatrix} \text{ is an auxiliary}$$

matrix with auxiliary variables $b_{1i} = \frac{1}{I_{xxi}}$, $b_{3i} = \frac{1}{I_{yyi}}$, and $b_{5i} = \frac{1}{I_{zzj}}$ defined to ensure that the attitude dynamics defined in Eq. (3) are in a compact form.

Now, the following problems are considered in this work:

Problem 1: Formulating multiple problems, including trajectory tracking, formation tracking control, and interagent collision avoidance, for a multiquadcopter system such that the proposed barrier Lyapunov theory can be applied.

Problem 2: Deriving the decentralized (distributed) hierarchical control laws for collision-free formation tracking control for the altitude subsystem as well as translational x and y subsystems with virtual inputs in a backstepping framework.

Problem 3: Designing the attitude control laws separately for each agent using the desired signals generated via BLFs while considering a predefined accuracy.

3 Control objectives

Problem 1 is considered in this section. In this work, the goal is to design controllers $u_{1i}(t), \dots, u_{4i}(t)$ such that the control objectives are achieved. The control objective in this study is formation tracking control, which consists of two parts. First, each quadcopter should follow its specified desired trajectory with a predefined accuracy for position and orientation. Second, interagent collisions should be avoided based on specified bounds regarding how close the quadcopters can be.

3.1 Trajectory tracking error

Let $x_{1id}(t), i = 1, 2, \dots, N$ (where N indicates the number of quadcopters) be the desired altitude trajectory for the i th quadcopter that is continuous in time and has finite first- and second-order derivatives. Further, we define the altitude tracking error for the i th quadcopter as $e_{1i}(t) = x_{1i}(t) - x_{1id}(t)$. The first control objective here is to ensure that the altitude of the i th quadcopter tracks the desired trajectory $x_{1id}(t)$ with a predefined accuracy; this can be formulated by ensuring that the altitude tracking error for the i th quadcopter remains with a compact set defined as follows:

$$D_{1ie} = \{e_{1i}(t) \in \mathbb{R} | d_{1ie}(t) = |e_{1i}(t)| < \Omega_{1idH}, t \geq 0\}, \quad (10)$$

where Ω_{1idH} is a positive scalar defined for the i th quadcopter with an upper bound for the tracking error.

Similar to the altitude, $x_{3id}(t)$ and $x_{5id}(t)$ are the desired translational trajectories for the i th quadcopter in the x and y directions, respectively. It is assumed that these desired trajectories are continuous in time and have limited first- and second-order derivatives. Further, $e_{3i}(t) = x_{3i}(t) - x_{3id}(t)$ and $e_{5i}(t) = x_{5i}(t) - x_{5id}(t)$ represent the tracking errors in the x and y directions for the i th quadcopter, respectively. The control objective here is to track the desired translational trajectories $x_{3id}(t)$ and $x_{5id}(t)$ with a predetermined accuracy; this can be formulated by ensuring that the translational tracking error for the i th quadcopter remains within a compact set defined as follows:

$$\begin{aligned} D_{3ie} &= \{e_{3i}(t) \in \mathbb{R} | d_{3ie}(t) = |e_{3i}(t)| < \Omega_{3idH}, t \geq 0\}, \\ D_{5ie} &= \{e_{5i}(t) \in \mathbb{R} | d_{5ie}(t) = |e_{5i}(t)| < \Omega_{5idH}, t \geq 0\}, \end{aligned} \quad (11)$$

where Ω_{3idH} and Ω_{5idH} are two separate positive scalars defined for the i th quadcopter in the x and y directions, respectively, with upper bounds for the tracking errors.

Finally, for the attitude subsystem, $x_{7id}(t)$ is considered as the desired trajectory vector for the i th quadcopter and assumed to be continuous in time with limited first- and second-order derivatives. Further, $e_{7i}(t) = x_{7i}(t) - x_{7id}(t)$ represents the attitude tracking error vector. The control objective here is to track the desired trajectory vector $x_{7id}(t)$ with a predefined accuracy; this can be formulated by ensuring that the attitude tracking error for the i th quadcopter remains within the compact set defined as follows:

$$D_{7ie} = \{e_{7i}(t) \in \mathbb{R}^3 | d_{7ie}(t) = \|e_{7i}(t)\| < \Omega_{7idH}, t \geq 0\}, \quad (12)$$

where Ω_{7idH} is a predefined positive scalar with an upper bound for the tracking error, and $\|\cdot\|$ is the 2-norm of the vector.

3.2 Collision avoidance and formation control

As introduced earlier, $x_{1i}(t)$ is the altitude of the i th quadcopter with $x_{1j}(t), j \in N_i$ being the altitudes of its neighboring agents. The goal here is that the distances of the real altitudes of each of the agents with their neighbors, i.e., $d_{1ij}(t) \triangleq |x_{1i}(t) - x_{1j}(t)|, j \in N_i$, will track the desired distances, i.e., $L_{1ij}(t) \triangleq |x_{1id}(t) - x_{1jd}(t)|, j \in N_i$, which are expressed by $d'_{1ije}(t) \triangleq |d_{1ij}(t) - L_{1ij}(t)|, j \in N_i$ with a predefined accuracy. Thus, formation control and interagent collision avoidance

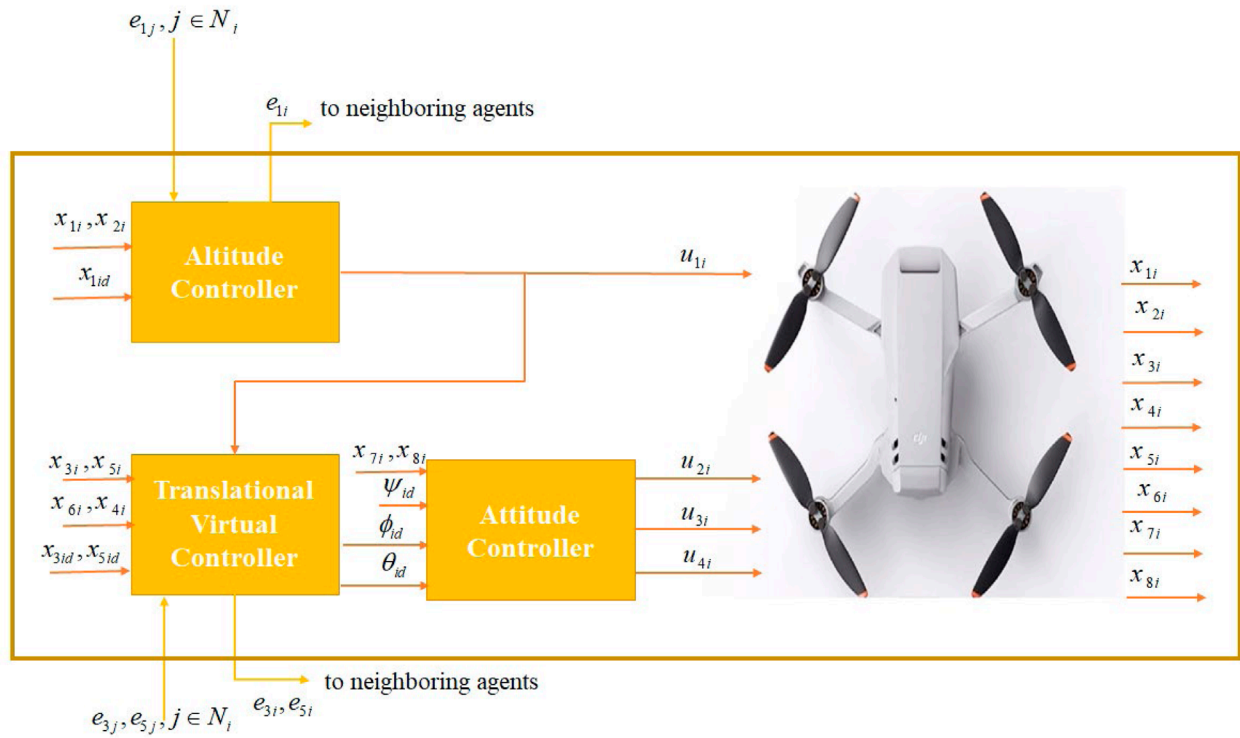


FIGURE 1
General framework of the proposed controller for the i th agent.



FIGURE 2
Interconnections between the agents in the simulation.

in the z direction are both guaranteed. This is achieved by ensuring that the error $d'_{1je}(t)$ for the i th quadcopter remains within the compact set defined as follows:

$$D_{1ijH} = \{x_{1i}(t) \in \mathbb{R} | d'_{1je}(t) < \Omega_{1ijH}, j \in N_i, t \geq 0\}, \quad (13)$$

where Ω_{1ijH} is a positive predefined scalar with an upper bound for formation tracking and a collision avoidance bound.

The real distance of the i th quadcopter from its neighboring agents in the x direction is given by $d_{3ij}(t) \triangleq |x_{3i}(t) - x_{3j}(t)|, j \in N_i$, while the desired distance is represented by $L_{3ij}(t) \triangleq |x_{3id}(t) - x_{3jd}(t)|, j \in N_i$. Then, the goals of formation control and interagent collision avoidance in the x direction for the i th quadcopter are guaranteed with a predefined accuracy if $d'_{3ije}(t) \triangleq |d_{3ij}(t) - L_{3ij}(t)|, j \in N_i$ remains within the compact set defined as follows:

$$D_{3ijH} = \{x_{3i}(t) \in \mathbb{R} | d'_{3ije}(t) < \Omega_{3ijH}, j \in N_i, t \geq 0\}, \quad (14)$$

where Ω_{3ijH} is a positive predefined scalar with an upper bound for formation tracking and a collision avoidance bound.

Similarly, the real distance of the i th quadcopter from its neighboring agents in the y direction is given by $d_{5ij}(t) \triangleq |x_{5i}(t) - x_{5j}(t)|, j \in N_i$, while the desired distance is represented by $L_{5ij}(t) \triangleq |x_{5id}(t) - x_{5jd}(t)|, j \in N_i$. Then, the goals of formation control and interagent collision avoidance in the y direction for i th quadcopter are guaranteed with a predefined accuracy if $d'_{5ije}(t) \triangleq |d_{5ij}(t) - L_{5ij}(t)|, j \in N_i$ remains within the compact set defined as follows:

$$D_{5ijH} = \{x_{5i}(t) \in \mathbb{R} | d'_{5ije}(t) < \Omega_{5ijH}, j \in N_i, t \geq 0\}, \quad (15)$$

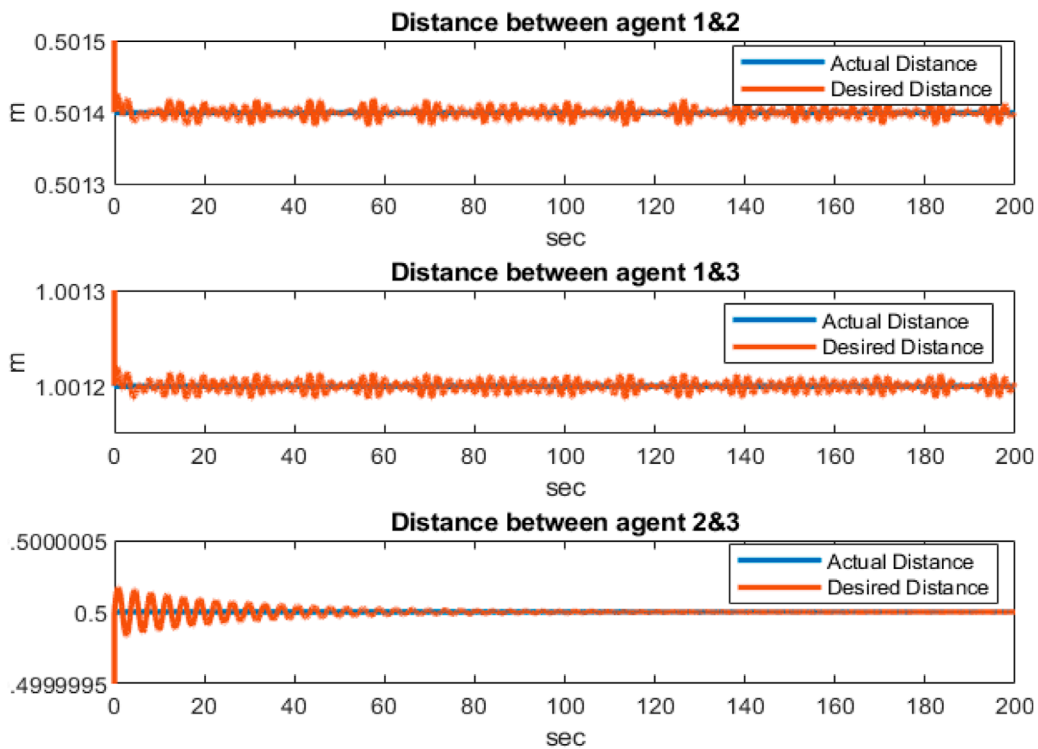


FIGURE 3
Distances between the agents in the simulation.

where Ω_{5ijH} is a positive predefined scalar with an upper bound for formation tracking and a collision avoidance bound.

Remark 1: It is worth noting that the problem considered here is neither LF control nor formation producing with/without a virtual leader. However, the shape of the formation is imposed on the method by appropriate design of the desired trajectory for each quadcopter.

4 Proposed distributed collision-free formation tracking control

Problem 2 is considered in this section, and the decentralized (distributed) hierarchical control laws for collision-free formation tracking control for the altitude and translational x and y subsystems with virtual inputs are designed in a backstepping framework for the multi-quadcopter system. As mentioned earlier, because of the underactuated nature of the quadcopter system, a hierarchical procedure is employed. As the first step, the altitude controller is designed, and its result is used to design the controller for the translational subsystems along with virtual control inputs.

4.1 Altitude subsystem

Theorem 1: Assume that the altitude subsystem of the i th quadcopter in a fleet of N quadcopters is described by Eq. (5).

Then, the altitude control input for the i th quadcopter $u_{1i}(t)$ can be designed as

$$u_{1i}(t) = g_{2i}^{-1}(t)[g + \dot{\alpha}_{1i}(t) - A_{1i}(t) - k_{2i}z_{2i}(t)]; \quad (16)$$

where

$$\begin{aligned} \alpha_{1i}(t) &= \dot{x}_{1id}(t) + \frac{-k_{1i}e_{1i}^2(t)}{D_{1i}(t)e_{1i}(t) - 2 \sum_{j \in N_i} B_{1ij}(t)e_{1j}(t)}, \\ A_{1i}(t) &= D_{1i}(t)e_{1i}(t) - 2 \sum_{j \in N_i} B_{1ij}(t)e_{1j}(t), \\ D_{1i}(t) &= B_{1i}(t) + 2 \sum_{j \in N_i} B_{1ij}(t), \end{aligned} \quad (17)$$

with $B_{1i} = \frac{\Omega_{1idH}^3}{(\Omega_{1idH} - d'_{1ie}(t))^3}$, $B_{1ij} = \frac{\Omega_{1ijH}^3}{(\Omega_{1ijH} - d'_{1ije}(t))^3}$, $\Omega_{1ijH} > 0$, $\Omega_{1idH} > 0$, and $k_{1i}, k_{2i} > 0$. Then, the altitude tracking error and interagent collision avoidance conditions are guaranteed by remaining within the sets defined by Eqs. (10) and (13) if the quadcopter starts with the initial conditions such that the tracking errors remain within the same sets, i.e., $d_{1ie}(0) < \Omega_{1idH}$ and $d'_{1ije}(0) < \Omega_{1ijH}$, respectively.

Proof: We choose the following BLF candidate that contains the BLFs for each of the agents ($V_{1ie}(t)$) as well as those related to the interagents ($V_{1ij}(t)$):

$$V_{1i}(t) = \sum_{i=1}^N \left(V_{1ie}(t) + \sum_{j=1, j \neq i}^N a_{ij} V_{1ij}(t) \right), \quad (18)$$

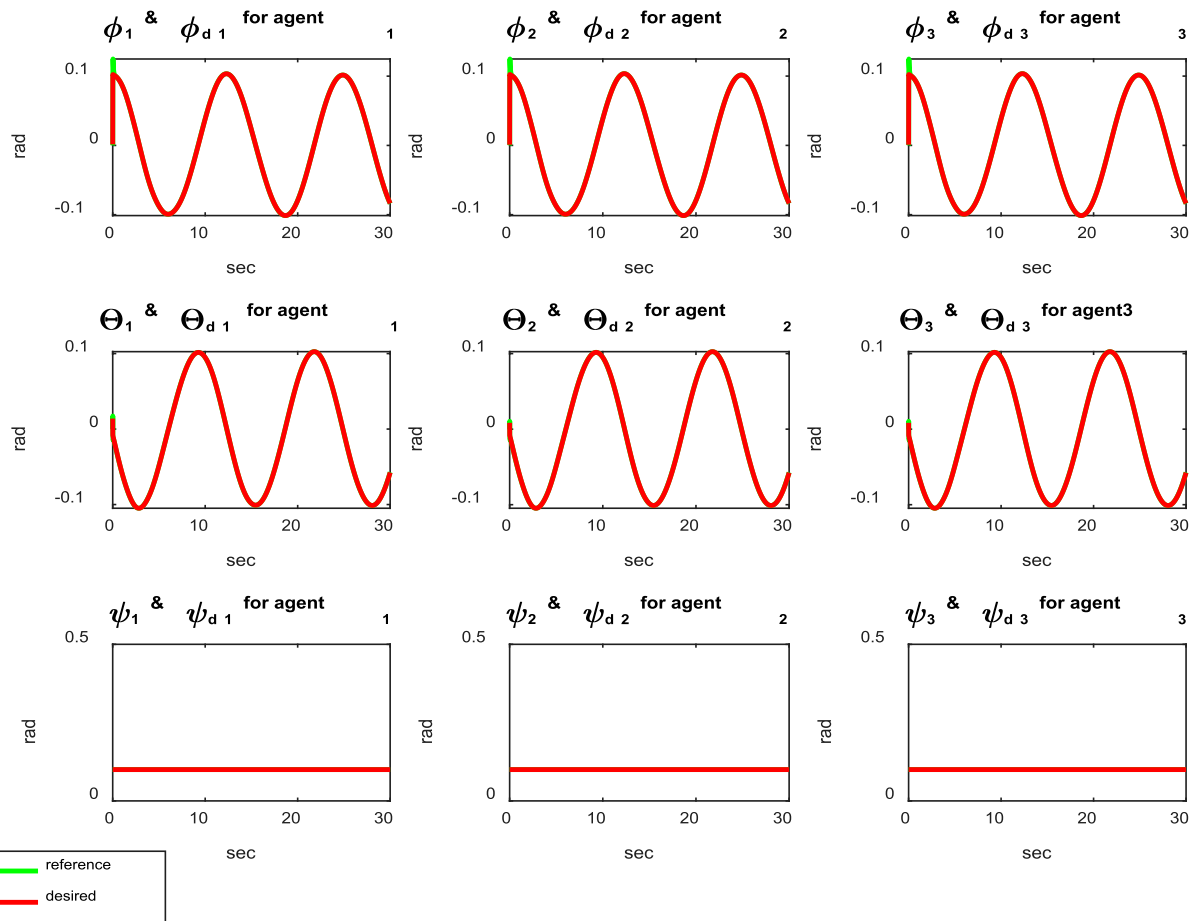


FIGURE 4
Comparison of the reference and actual trajectory states for the attitude subsystem.

where $V_{1ie}(t) = \frac{1}{2}\eta_{1ie}^2(t)$ and $V_{1ij}(t) = \frac{1}{2}\eta_{1ij}^2(t)$, with $\eta_{1ie}(t) = \frac{\Omega_{1idH}d_{1ie}(t)}{\Omega_{1idH}-d_{1ie}(t)}$ and $\eta_{1ij}(t) = \frac{\Omega_{1ijH}d'_{1ije}(t)}{\Omega_{1ijH}-d'_{1ije}(t)}$ according to Eq. (2).

It is obvious from Eq. (18) that V_{1i} is a positive-definite function. Therefore,

$$\begin{aligned}\dot{V}_{1i}(t) &= \sum_{i=1}^N \left(\dot{V}_{1ie}(t) + \sum_{j=1, j \neq i}^N a_{ij} \dot{V}_{1ij}(t) \right) \\ &= \sum_{i=1}^N \left(\eta_{1ie}(t) \dot{\eta}_{1ie}(t) + \sum_{j=1, j \neq i}^N a_{ij} \eta_{1ij}(t) \dot{\eta}_{1ij}(t) \right),\end{aligned}\quad (19)$$

where $\dot{V}_{1ie}(t) = \eta_{1ie}(t) \dot{\eta}_{1ie}(t) = \frac{\Omega_{1idH}^3}{(\Omega_{1idH}-d_{1ie}(t))^3} d_{1ie}(t) \dot{d}_{1ie}(t)$; further, by letting $B_{1i}(t) = \frac{\Omega_{1idH}^3}{(\Omega_{1idH}-d_{1ie}(t))^3}$, it can be concluded that $\dot{V}_{1ie}(t) = B_{1i}(t) d_{1ie}(t) \dot{d}_{1ie}(t)$. Since $d_{1ie} = |e_{1i}|$, we have

$$\begin{aligned}\dot{V}_{1ie}(t) &= B_{1i}(t) |e_{1i}(t)| \frac{d}{dt} |e_{1i}(t)| \\ &= B_{1i}(t) |e_{1i}(t)| \operatorname{sgn}(e_{1i}(t)) \dot{e}_{1i}(t) \\ &= B_{1i}(t) e_{1i}(t) \dot{e}_{1i}(t).\end{aligned}\quad (20)$$

Moreover, $\dot{V}_{1ij}(t) = \eta_{1ij}(t) \dot{\eta}_{1ij}(t) = \frac{\Omega_{1ijH}^3}{(\Omega_{1ijH}-d'_{1ije}(t))^3} d'_{1ije}(t) \dot{d}'_{1ije}(t)$. Now,

by letting $B_{1ij}(t) = \frac{\Omega_{1ijH}^3}{(\Omega_{1ijH}-d'_{1ije}(t))^3}$, it is concluded that

$$\begin{aligned}\dot{V}_{1ij}(t) &= B_{1ij}(t) d'_{1ije}(t) \dot{d}'_{1ije}(t) \\ &= B_{1ij}(t) (e_{1i}(t) - e_{1j}(t)) (\dot{e}_{1i}(t) - \dot{e}_{1j}(t)).\end{aligned}\quad (21)$$

Finally, by replacing Eqs. (20) and (21) in Eq. (19), we obtain

$$\dot{V}_{1i}(t) = \sum_{i=1}^N \left(B_{1i}(t) e_{1i}(t) \dot{e}_{1i}(t) + \sum_{j=1, j \neq i}^N a_{ij} B_{1ij}(t) (e_{1i}(t) - e_{1j}(t)) (\dot{e}_{1i}(t) - \dot{e}_{1j}(t)) \right).\quad (22)$$

By rearranging Eq. (22), we have

$$\begin{aligned}\dot{V}_{1i}(t) &= \sum_{i=1}^N B_{1i}(t) e_{1i}(t) \dot{e}_{1i}(t) + \sum_{i=1}^N \sum_{j=1}^N a_{ij} B_{1ij}(t) (e_{1i}(t) - e_{1j}(t)) \dot{e}_{1i}(t) \\ &\quad - \sum_{i=1}^N \sum_{j=1}^N a_{ij} B_{1ij}(t) (e_{1i}(t) - e_{1j}(t)) \dot{e}_{1j}(t).\end{aligned}\quad (23)$$

Using the summation properties and the fact that $a_{ij} = a_{ji}$ and $B_{1ij}(t) = B_{1ji}(t)$, it is concluded that $-\sum_{i=1}^N \sum_{j=1}^N a_{ij} B_{1ij}(t) (e_{1i}(t) - e_{1j}(t)) \dot{e}_{1j}(t)$

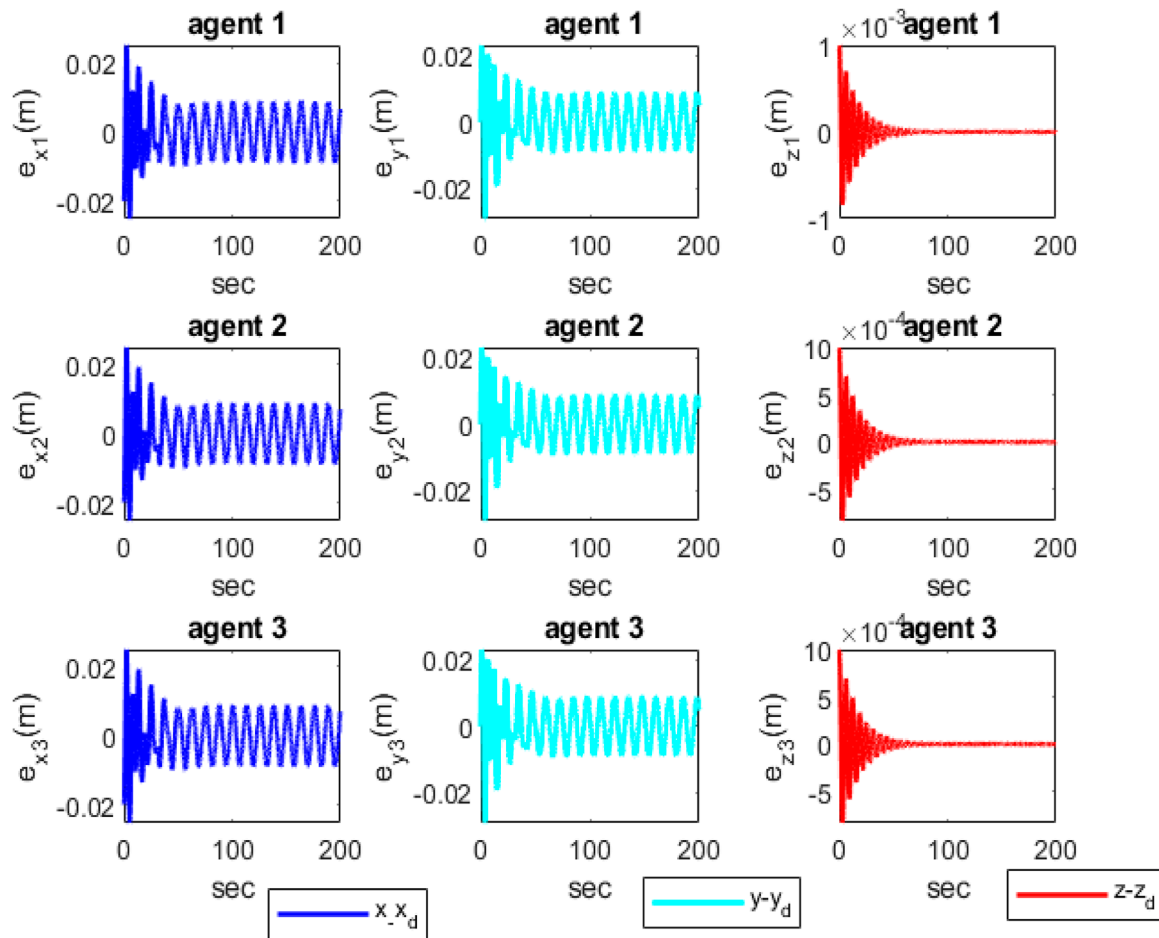


FIGURE 5
Position tracking errors of the agents in the x , y , and z axes.

$(t) = \sum_{i=1}^N \sum_{j=1}^N a_{ij} B_{1ij}(t) (e_{1i}(t) - e_{1j}(t)) \dot{e}_{1i}(t)$; hence, Eq. (23) can be rewritten as

$$\dot{V}_{1i}(t) = \sum_{i=1}^N \left(B_{1i}(t) e_{1i}(t) + 2 \sum_{j=1}^N a_{ij} B_{1ij}(t) (e_{1i}(t) - e_{1j}(t)) \right) \dot{e}_{1i}(t). \quad (24)$$

Assuming that $z_{2i}(t) = x_{2i}(t) - \alpha_{1i}(t)$, we have $x_{2i}(t) = z_{2i}(t) + \alpha_{1i}(t)$. Since $\dot{e}_{1i}(t) = x_{2i}(t) - \dot{x}_{1id}(t)$, the expression can be rewritten as $\dot{e}_{1i}(t) = z_{2i}(t) + \alpha_{1i}(t) - \dot{x}_{1id}(t)$. Now, substituting this into Eq. (24), the stabilizing function α_{1i} is derived as in Eq. (17). Therefore, Eq. (24) can be rewritten as $\dot{V}_{1i}(t) = \sum_{i=1}^N [A_{1i}(t) z_{2i} - k_{1i} e_{1i}^2(t)]$. By defining a backstepping-type Lyapunov function candidate and adding a quadratic function to $V_{1i}(t)$, we have $V_{2i}(t) = V_{1i}(t) + \frac{1}{2} \sum_{i=1}^N z_{2i}^2(t)$. Taking the derivative of the Lyapunov function gives

$$\begin{aligned} \dot{V}_{2i}(t) &= \dot{V}_{1i}(t) + \sum_{i=1}^N z_{2i}(t) \dot{z}_{2i}(t) \\ &= \dot{V}_{1i}(t) + \sum_{i=1}^N z_{2i}(t) (-g + g_{4i}(t) u_{1i}(t) - \dot{\alpha}_{1i}(t)). \end{aligned} \quad (25)$$

Replacing $u_{1i}(t)$ from Eq. 16 into 25 gives

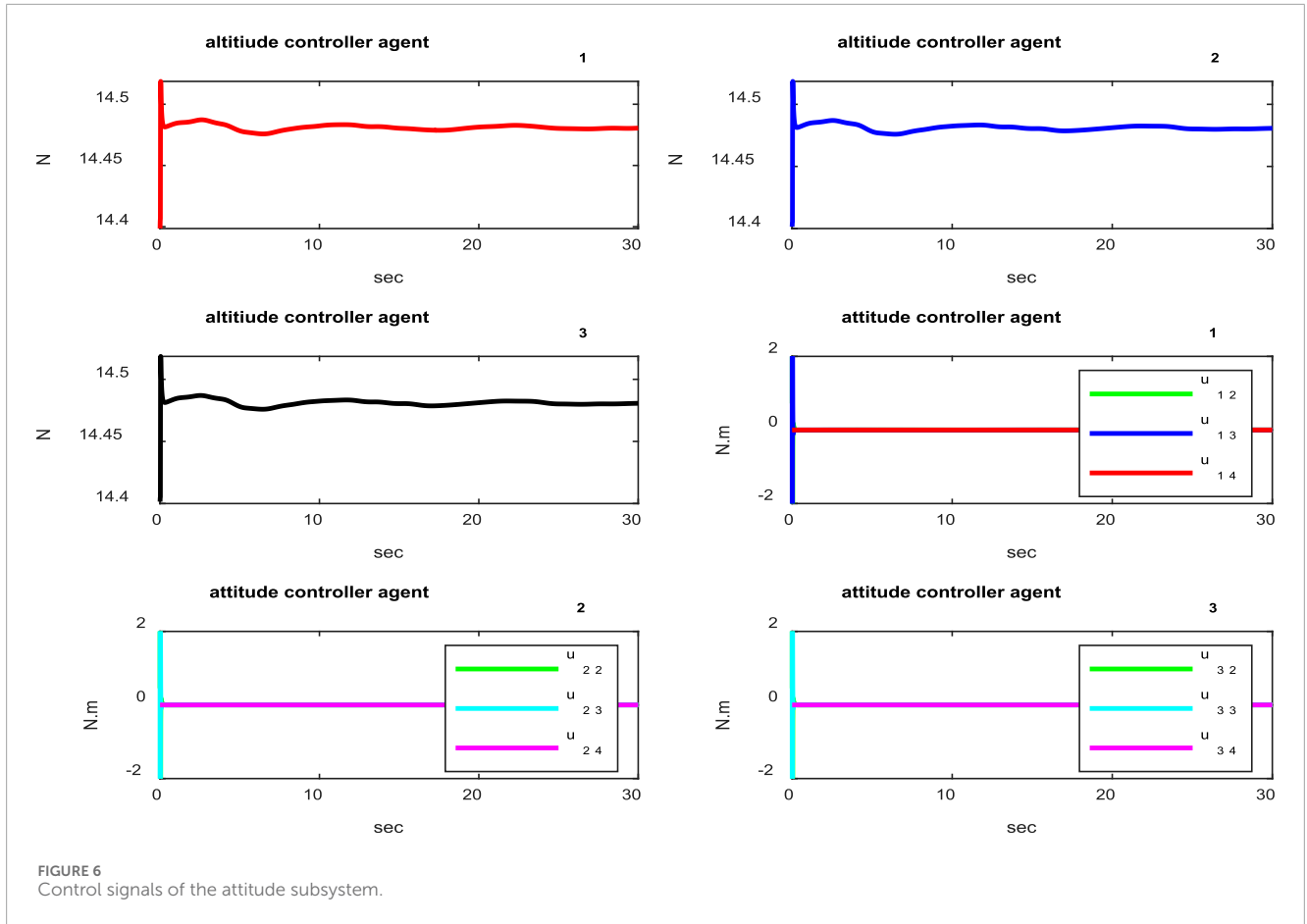
$$\begin{aligned} \dot{V}_{2i}(t) &= - \sum_{i=1}^N k_{1i} e_{1i}^2(t) + \sum_{i=1}^N A_{1i}(t) z_{2i}(t) \\ &\quad + \sum_{i=1}^N z_{2i}(t) (-g - \dot{\alpha}_{1i}(t) + g + \dot{\alpha}_{1i}(t) - A_{1i}(t) - k_{2i} z_{2i}(t)) \\ &= - \sum_{i=1}^N k_{1i} z_{1i}^2(t) - \sum_{i=1}^N k_{2i} z_{2i}^2(t). \end{aligned} \quad (26)$$

Therefore, one can conclude from Eq. (26) that $\dot{V}_{2i}(t) < 0$, which completes the proof.

4.2 Translational subsystems

Herein, the virtual controllers for the translational subsystems presented in Eq. (6) for the x and y coordinates are formulated in accordance with Theorem 2.

Theorem 2: Assume that the translational subsystems of the i th quadcopter in a fleet of N quadcopters can be described by Eq. (6) in the x and y directions using the virtual control inputs defined in Eqs. (7) and (8). Then, the virtual control



inputs $u_{iv3}(t)$ and $u_{iv5}(t)$ for the i th quadcopter can be designed as

$$\begin{aligned} u_{iv3}(t) &= g_{4i}^{-1}(t) [\dot{\alpha}_{3i}(t) - A_{3i}(t) - k_{4i}z_{4i}(t)], \\ u_{iv5}(t) &= g_{6i}^{-1}(t) [\dot{\alpha}_{5i}(t) - A_{5i}(t) - k_{6i}z_{6i}(t)], \end{aligned} \quad (27)$$

where

$$\begin{aligned} \alpha_{3i}(t) &= \dot{x}_{3id}(t) + \frac{-k_{3i}e_{3i}^2(t)}{D_{3i}(t)e_{3i}(t) - 2 \sum_{j \in N_i} B_{3ij}(t)e_{3j}(t)}, \\ A_{3i}(t) &= D_{3i}(t)e_{3i}(t) - 2 \sum_{j \in N_i} B_{3ij}(t)e_{3j}(t), \\ D_{3i}(t) &= B_{3i}(t) + 2 \sum_{j \in N_i} B_{3ij}(t), \\ \alpha_{5i}(t) &= \dot{x}_{5id}(t) + \frac{-k_{5i}e_{5i}^2(t)}{D_{5i}(t)e_{5i}(t) - 2 \sum_{j \in N_i} B_{5ij}(t)e_{5j}(t)}, \\ A_{5i}(t) &= D_{5i}(t)e_{5i}(t) - 2 \sum_{j \in N_i} B_{5ij}(t)e_{5j}(t), \\ D_{5i}(t) &= B_{5i}(t) + 2 \sum_{j \in N_i} B_{5ij}(t), \end{aligned} \quad (28)$$

with $B_{3i}(t) = \frac{\Omega_{3idH}^3}{(\Omega_{3idH} - d'_{3ie}(t))^3}$, $B_{3ij}(t) = \frac{\Omega_{3ijH}^3}{(\Omega_{3ijH} - d'_{3ije}(t))^3}$, $B_{5i}(t) = \frac{\Omega_{5idH}^3}{(\Omega_{5idH} - d'_{5ie}(t))^3}$, $B_{5ij}(t) = \frac{\Omega_{5ijH}^3}{(\Omega_{5ijH} - d'_{5ije}(t))^3}$, $z_{4i}(t) = x_{4i}(t) - \alpha_{3i}(t)$, $z_{6i}(t) = x_{6i}(t) - \alpha_{5i}(t)$, $k_{3i} > 0$, $k_{4i} > 0$, $k_{5i} > 0$, $k_{6i} > 0$, $\Omega_{1ijH} > 0$, and $\Omega_{1idH} > 0$.

Then, the translational tracking error and interagent collision avoidance conditions are guaranteed by remaining within the sets defined by Eqs. (11), (14), and (15) if the quadcopter starts with the initial conditions such that the tracking errors remain within the same sets, i.e., $d_{3ie}(0) < \Omega_{3idH}$, $d_{5ie}(0) < \Omega_{5idH}$, $d'_{3ije}(0) < \Omega_{3ijH}$, and $d'_{5ije}(0) < \Omega_{5ijH}$ respectively.

Proof: A proof similar to that of Theorem 1 can be considered here and has been omitted for brevity.

5 Proposed attitude control system

Problem 3 is considered in this section, and a BLF-based controller is designed for the attitude subsystem with the dynamics presented in Eq. (9). First, according to Eqs. (7) and (8) as well as the virtual controllers designed for the translational subsystems in Eqs. (27) and (28) the desired angles for the roll ($\phi_{id}(t)$) and pitch ($\theta_{di}(t)$) are computed in Eq. (29) as follows:

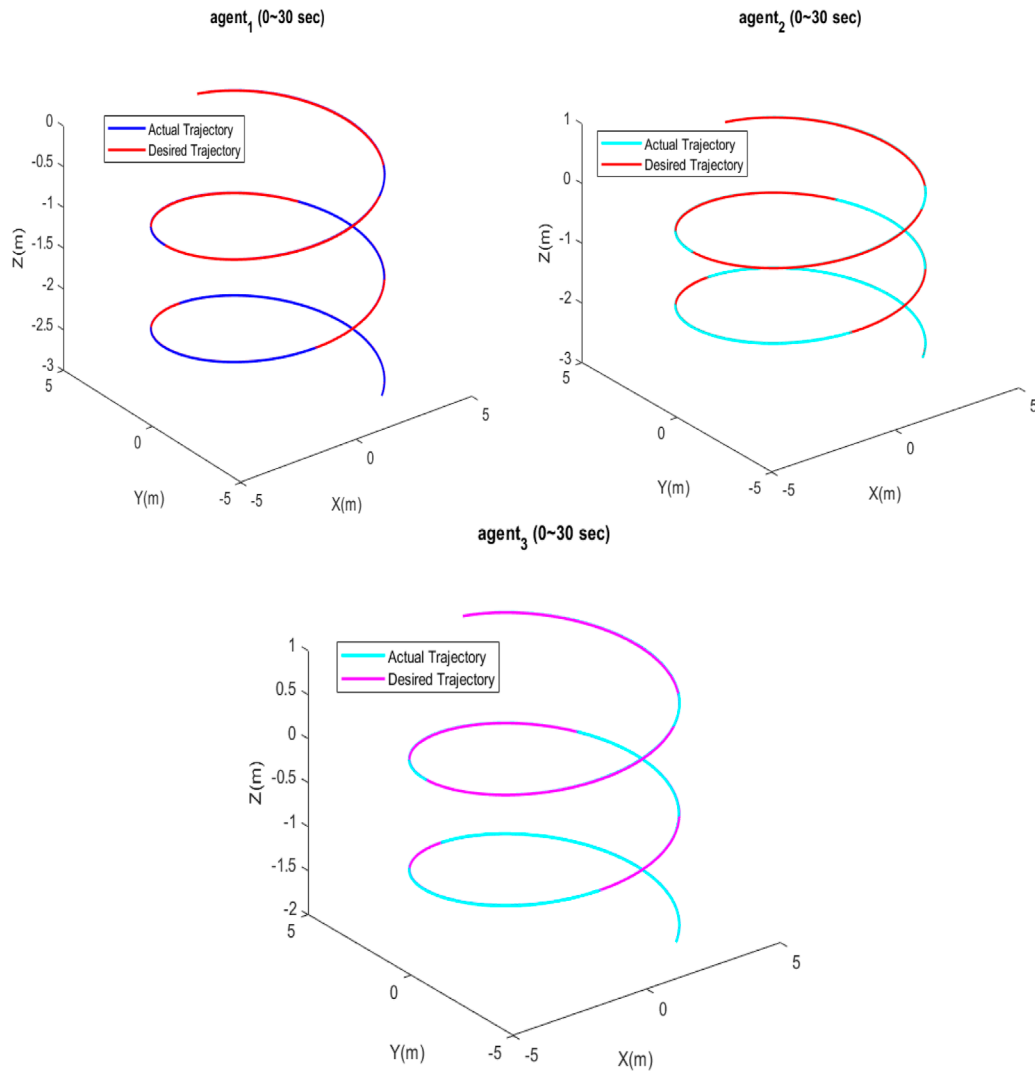


FIGURE 7
Actual and desired 3D positions of the quadcopters.

$$\begin{aligned}\sin(\phi_{id}(t)) &= u_{iv3}(t)\sin(\psi_i(t)) - u_{iv5}(t)\cos(\psi_i(t)) \\ \sin(\theta_{di}(t)) &= \frac{u_{iv3}(t)\cos(\psi_i(t)) + u_{iv5}(t)\sin(\psi_i(t))}{\cos(\phi_{id}(t))}\end{aligned}\quad (29)$$

The desired yaw angle ($\psi_{id}(t)$) can be set freely.

Theorem 3: Assume that the attitude subsystem of the i th quadcopter in a fleet of N quadcopters can be described by Eq. (9); then, the control input vector $u_i(t)$ for the i th quadcopter can be designed as

$$u_i(t) = G_{8i}^{-1}(-f_{2i}(t) + \dot{\alpha}_{8i}(t) - B_{7i}(t)e_{7i}(t) - k_{8i}z_{8i}(t)), \quad (30)$$

where $z_{8i}(t) = x_{8i}(t) - \alpha_{8i}(t)$ and $\alpha_{8i}(t) = \dot{x}_{7id} - (k_{7i}e_{7i}(t)/B_{7i}(t))$, with $B_{7i} = \frac{\Omega_{7idH}^3}{(\Omega_{7idH} - d_{7ie})^3}$ and k_{7i} , k_{8i} , and Ω_{7idH} being scalar positive constants.

Then, the attitude tracking error is guaranteed by remaining within the set defined in Eq. (12) if the quad starts from the initial

conditions such that the tracking errors remain within the same sets, i.e., $d_{7ie}(0) = \|e_{7i}(0)\| < \Omega_{7idH}$.

Proof: We consider the BLF $V_{7i}(t) = \frac{1}{2}\eta_{7ie}^2(t)$ with $\eta_{7ie}(t) = \frac{\Omega_{7idH}d_{7ie}(t)}{\Omega_{7idH} - d_{7ie}(t)}$; therefore,

$$\begin{aligned}\dot{V}_{7i}(t) &= \frac{\Omega_{7idH}^3}{(\Omega_{7idH} - d_{7i}(t))^3} e_{7i}^T(t) \dot{e}_{7i}(t) \\ &= B_{7i}(t) e_{7i}^T(t) \dot{e}_{7i}(t).\end{aligned}\quad (31)$$

Since $\dot{e}_{7i}(t) = x_8(t) - \dot{x}_{7id}(t) = z_{8i}(t) + \alpha_{8i}(t) - \dot{x}_{7id}(t)$, if we select $\alpha_{8i}(t) = \dot{x}_{7id} - (k_{7i}e_{7i}(t)/B_{7i}(t))$, we obtain $\dot{V}_{7i}(t) = B_{7i}(t)e_{7i}^T(t)z_{8i}(t) - k_{7i}e_{7i}^T(t)e_{7i}(t)$. Now, a Lyapunov function is chosen by adding a quadratic function to V_{7i} as follows:

$$V_{8i}(t) = V_{7i}(t) + \frac{1}{2}z_{8i}^T(t)z_{8i}(t). \quad (32)$$

Therefore, using Eq. (32) one can conclude Eq. (33) as follows:

$$\begin{aligned}\dot{V}_{8i}(t) &= \dot{V}_{7i}(t) + z_{8i}^T(t)\dot{z}_{8i}(t) \\ &= B_{7i}(t)e_{7i}^T(t)z_{8i}(t) - k_{7i}e_{7i}^T(t)e_{7i}(t) \\ &\quad + z_{8i}^T(t)(\dot{x}_{8i}(t) - \dot{\alpha}_{8i}(t)).\end{aligned}\quad (33)$$

Now, according to Eq. (9), by selecting $u_i(t)$ as Eq. (30) and using Eq. (31) it is concluded that $\dot{V}_{8i}(t) = -k_{7i}e_{7i}^T(t)e_{7i}(t) - k_{8i}z_{8i}^T(t)z_{8i}(t) < 0$. Therefore, the attitude tracking objective in Eq. (12) is satisfied if $d_{7ie}(0) = \|e_{7i}(0)\| < \Omega_{7idH}$, hence completing the proof.

Figure 1 depicts the general structure of the proposed controller for the i th agent. The overall quadcopter system has three subsystems. The design of $u_{1i}(t)$ starts from the altitude subsystem. Then, this controller is used to design the virtual controllers in the translational subsystems. Finally, the control inputs of the attitude subsystem are designed to meet the desired control objectives. Owing to the fact that the graph topology of the quadcopter system is connected, the neighboring information is used to achieve safety, collision avoidance, and stability.

6 Simulation results

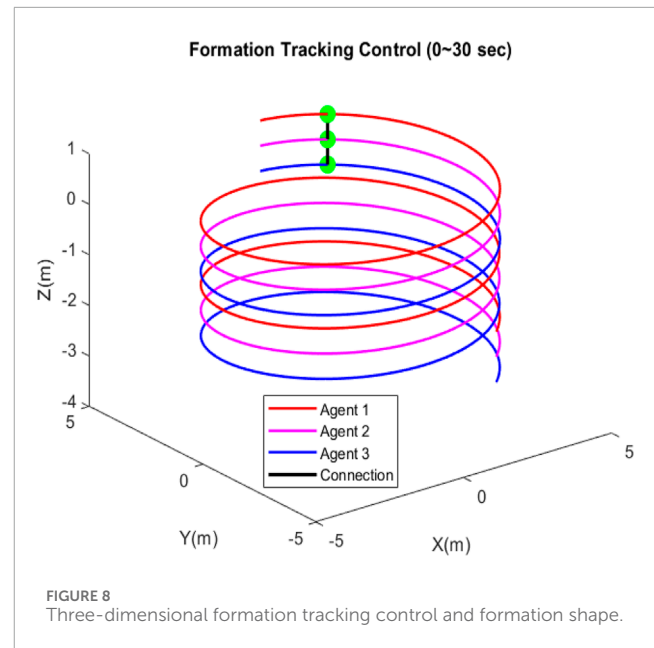
In this section, simulation results are provided to demonstrate the efficiency of the proposed method. Figure 2 depicts the interconnection of three quadcopters considered for the simulation.

The initial conditions are considered as follows: $x_{11}(0) = 0, x_{12}(0) = 0.5, x_{13}(0) = 1, x_{21}(0) = x_{22}(0) = x_{23}(0) = -0.08, x_{31}(0) = x_{32}(0) = x_{33}(0) = 0, x_{41}(0) = x_{42}(0) = x_{43}(0) = 1.5, x_{51}(0) = x_{52}(0) = x_{53}(0) = 3.9, x_{61}(0) = x_{62}(0) = x_{63}(0) = 0, x_{7i}(0) = [0 \ 0 \ 0], i = 1, 2, 3$, and $x_{8i}(0) = [0 \ 0 \ 0], i = 1, 2, 3$. The physical parameters of the quadcopters are as follows: $m_i = 1.47\text{ kg}, I_{xxi} = I_{yyi} = 0.01152\text{ kgm}^2, I_{zzi} = 0.0218\text{ kgm}^2$, and $L_i = 0.28, i = 1, 2, 3$. The reference trajectory for the movement of the quadcopters is given in Eq. (34) as follows:

$$\begin{aligned}z_{1d}(t) &= -0.1t, z_{2d}(t) = -0.1t + 0.5, z_{3d}(t) = -0.1t + 1 \\ x_{1d}(t) &= x_{2d}(t) = x_{3d}(t) = 4 \sin(0.5t) \\ y_{1d}(t) &= y_{2d}(t) = y_{3d}(t) = 4 \cos(0.5t).\end{aligned}\quad (34)$$

As mentioned previously, the values $\Omega_{1idH} > 0$ to $\Omega_{1=7idH} > 0$ are the upper limits for distance tracking errors d_{1ie} to d_{7ie} , while $\Omega_{1ijH} > 0, \Omega_{3ijH} > 0$, and $\Omega_{5ijH} > 0$ are the respective upper limits for the distance tracking errors d'_{1ije}, d'_{3ije} , and d'_{5ije} . These two sets of parameters determine the safe sets for the movements of the quadcopters. If these values are selected to be large, although the safety set will be larger, it may cause problems for the system in terms of safety as a wider range of errors would be considered acceptable. If these values are too small, then the safe set will be too small and forces the selection of the initial values to be very close to the real ones, which is unrealistic and may force the algorithm to be very sensitive to small deviations of the errors. Therefore, the selection of these two sets of parameters is very important. In the simulations, they are selected as follows: $\Omega_{7idH} = 1.5, \Omega_{1idH} = \Omega_{1ijH} = 0.1, \Omega_{3idH} = \Omega_{3ijH} = 0.49$, and $\Omega_{5idH} = \Omega_{5ijH} = 0.25$. The simulation results are as follows.

Figure 3 shows the distances between the agents, indicating that the agents are collision-free and maintain distances specified by the reference trajectories between the quadcopters during movement.



According to Figure 4, it is clear that the attitude control subsystem is well designed as the states (ϕ, θ, ψ) follow the desired trajectories. The position tracking errors of the agents in the x, y , and z axes are depicted in Figure 5, according to which the error is less than 0.03; this shows that the controller is well designed and that the tracking error is acceptable. It is clear from Figure 6 that the value of the control signal $u_{1i}, i = 1, 2, 3$ converges approximately to 14.48 N and that the values of the control signals for the attitude subsystem converge to 0 N·m.

Figure 7 depicts that each quadcopter follows its desired path; thus, each quadcopter tracks its desired trajectory successfully during flight. The results of formation tracking as well as the formation shape are depicted in Figure 8; it is obvious from the figure that the quadcopters follow their trajectories in formation without any collisions and that the desired distances between them are maintained. Figure 9 shows that the error value converges to a constant equal to 0.0083, with a settling time of 50 s.

The method proposed in this work is compared with that of Kuriki and Namerikawa (2015) in Table 1, from which it is obvious that the proposed method is significantly superior based on different aspects. The root mean-squared error (RMSE) as well as steady-state error values for the proposed method are considerably lower, and our method is significantly faster. Although the settling time in our method is a bit large, it still outperforms the oscillating behavior of the method proposed by Kuriki and Namerikawa (2015).

7 Conclusion

The purpose of this work was to design a distributed collision-free formation tracking control scheme for multi-quadcopter systems using the BLF in a backstepping procedure. The controllers were designed in a hierarchical structure to tackle the underactuated nature of the quadcopter system. Accordingly, the altitude controller was designed first, followed by the translational controller with virtual inputs. The desired Euler angles were then obtained using

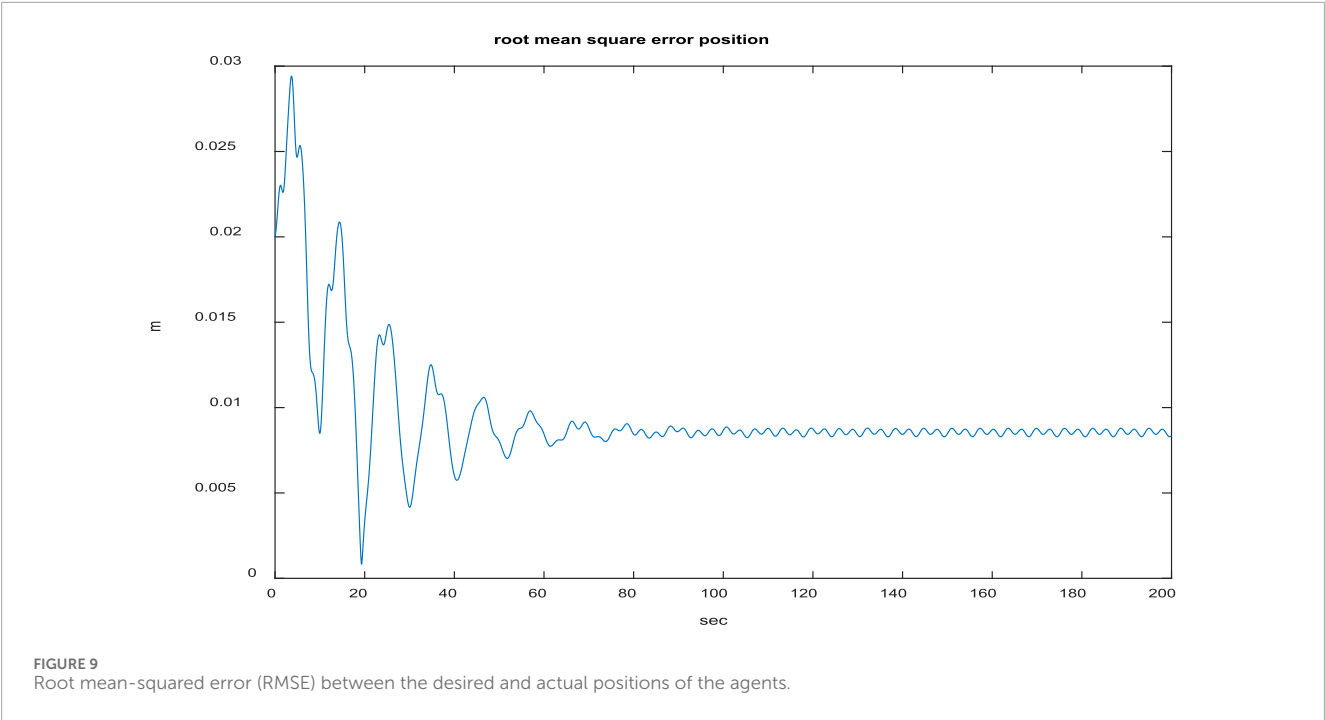


TABLE 1 Comparison of the proposed method with the system of Kuriki and Namerikawa (2015).

	Settling time (s)	RMSE (m)	Average calculation time (s)	Steady-state error (m)
Proposed method	50	0.0083	0.000053741	0.0083
Kuriki and Namerikawa (2015)	Fluctuating behavior	0.2374	0.005	0.2288

the virtual control signals and were finally employed to derive the proposed BLF-based attitude control subsystem. Simulations were performed to demonstrate the control objectives designed and achieved herein, including safety (staying in a safe set) and collision avoidance as well as formation tracking control. By adding the uncertainty terms and noise to the dynamics of the system, the controller can be designed such that it meets the control goals when the specified cases occur; this can be considered as a suggestion for future work. Formulating the problem of obstacle avoidance using the BLF is also suggested as a future work.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, and any further inquiries may be directed to the corresponding authors.

Author contributions

MS: Investigation, Software, Visualization, Writing–original draft. NS-N: Methodology, Project administration, Supervision, Writing–review and editing. RB: Project administration, Writing–review and editing, Investigation, Validation. AM: Validation, Writing–review and editing, Funding acquisition, Methodology, Resources, Supervision.

Funding

The author(s) declare that no financial support was received for the research, authorship, and/or publication of this article.

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 author(s) declare that they were an editorial board member of Frontiers at the time of submission. This had no impact on the peer review process or the final decision.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations or those of the publisher, editors, and reviewers. Any product that may be evaluated in this article or claim that may be made by its manufacturer is not guaranteed or endorsed by the publisher.

References

- Ahmadi, K., Asadi, D., Nabavi-Chashmi, S.-Y., and Tutsoy, O. (2023). Modified adaptive discrete-time incremental nonlinear dynamic inversion control for quadrotors in the presence of motor faults. *Mech. Syst. Signal Process.* 188, 109989. doi:10.1016/j.ymssp.2022.109989
- Allahyar, S. D. W. Z. M. (2023). "Robotics and artificial intelligence in the nuclear industry: from teleoperation to cyber physical systems," in *Artificial intelligence for robotics and autonomous systems applications* A. Editor T. A. A. Koubaa (Berlin, Germany: Springer), 123–166.
- Burrell, T., West, C., Monk, S. D., Montazeri, A., and Taylor, C. J. (2018). "Towards a cooperative robotic system for autonomous pipe cutting in nuclear decommissioning," in 2018 UKACC 12th International Conference on Control (CONTROL), Sheffield, UK, September, 2018. doi:10.1109/control.2018.8516841
- Can, A., Imran, I. H., Price, J., and Montazeri, A. (2022). Robust formation control and trajectory tracking of multiple quadrotors using a discrete-time sliding mode control technique. *IFAC-PapersOnLine* 55 (10), 2974–2979. doi:10.1016/j.ifacol.2022.10.184
- Chen, Y., Singletary, A., and Ames, A. D. (2020). Guaranteed obstacle avoidance for multi-robot operations with limited actuation: a control barrier function approach. *IEEE Control Syst. Lett.* 5 (1), 127–132. doi:10.1109/lcsys.2020.3000748
- Ganguly, S. (2022). Robust trajectory tracking and payload delivery of a quadrotor under multiple state constraints. Available at: <https://arxiv.org/abs/2201.03711>.
- Hu, J., Bhowmick, P., Jang, I., Arvin, F., and Lanzon, A. (2021). A decentralized cluster formation containment framework for multirobot systems. *IEEE Trans. Robotics* 37 (6), 1936–1955. doi:10.1109/tro.2021.3071615
- Imran, I. H., and Montazeri, A. (2022). Distributed robust synchronization control of multiple heterogeneous quadcopters with an active virtual leader. *IFAC-PapersOnLine* 55 (10), 2659–2664. doi:10.1016/j.ifacol.2022.10.111
- James, M., Carr, B., D'Arcy, F., Diefenbach, A., Dietterich, H., Fornaciai, A., et al. (2020). Volcanological applications of unoccupied aircraft systems (UAS): developments, strategies, and future challenges. *Volcanica* 3, 67–114. doi:10.30909/vol.03.01.67114
- Jin, X., Dai, S.-L., Liang, J., and Guo, D. (2021). Multirobot system formation control with multiple performance and feasibility constraints. *IEEE Trans. Control Syst. Technol.* 30 (4), 1766–1773. doi:10.1109/tcst.2021.3117487
- Julian, K. D. K. M. J., and Kochenderfer, M. J. (2019). Distributed wildfire surveillance with autonomous aircraft using Deep reinforcement learning. *J. Guid. Control, Dyn.* 48 (2), 1768–1778. doi:10.2514/1.g004106
- Khadhraoui, A., Zouaoui, A., and Saad, M. (2023). Barrier Lyapunov function and adaptive backstepping-based control of a quadrotor UAV. *Robotica* 41 (10), 2941–2963. doi:10.1017/s0263574723000735
- Kumar, S., and Kumar, S. R. (2022). "Barrier lyapunov-based nonlinear trajectory following for unmanned aerial vehicles with constrained motion," in 2022 International Conference on Unmanned Aircraft Systems (ICUAS), Dubrovnik, Croatia, June, 2022. doi:10.1109/icuas54217.2022.9836037
- Kuriki, Y., and Namerikawa, T. (2015). Formation control with collision avoidance for a multi-UAV system using decentralized MPC and consensus-based control. *SICE J. Control, Meas. Syst. Integration* 8 (4), 285–294. doi:10.9746/jcmsi.8.285
- Liang, Y., Qi, D., and Yanjie, Z. (2020). Adaptive leader-follower formation control for swarms of unmanned aerial vehicles with motion constraints and unknown disturbances. *Chin. J. Aeronautics* 33 (11), 2972–2988. doi:10.1016/j.cja.2020.03.020
- Liu, Y., and Bucknall, R. (2018). A survey of formation control and motion planning of multiple unmanned vehicles. *Robotica* 36 (7), 1019–1047. doi:10.1017/s0263574718000218
- Lizzio, F. F., Capello, E., and Guglieri, G. (2022). A review of consensus-based multi-agent UAV implementations. *J. Intelligent Robotic Syst.* 106 (2), 43. doi:10.1007/s10846-022-01743-9
- Mansfield, D., and Montazeri, A. (2024). A survey on autonomous environmental monitoring approaches: towards unifying active sensing and reinforcement learning. *Front. Robotics AI* 11, 1336612. doi:10.3389/frobt.2024.1336612
- Martin, P. G., Kwong, S., Smith, N., Yamashiki, Y., Payton, O., Russell-Pavier, F., et al. (2016). 3D unmanned aerial vehicle radiation mapping for assessing contaminant distribution and mobility. *Int. J. Appl. Earth Observation Geoinformation* 52, 12–19. doi:10.1016/j.jag.2016.05.007
- Montazeri, A., Can, A., and Imran, I. H. (2021). "Chapter 3 - unmanned aerial systems: autonomy, cognition, and control," in *Unmanned aerial systems*. Editors A. Koubaa, and A. T. Azar (Cambridge, Massachusetts, United States: Academic Press), 47–80.
- Mughees, A., and Ahmad, I. (2023). Multi-optimization of novel conditioned adaptive barrier function integral terminal SMC for trajectory tracking of a quadcopter System. *IEEE Access* 11, 88359–88377. doi:10.1109/access.2023.3304760
- Neumann, T. F. A., Kallweit, S., and Scholl, I., Towards a mobile mapping robot for underground mines. November 2014; pp. In Proceedings of the 2014 PRASA, RobMech and AflaT International Joint Symposium Cape Town, South Africa, 2014: p. 27–28.
- Ngo, K. B., Mahony, R., and Jiang, Z.-P. (2005). "Integrator backstepping using barrier functions for systems with multiple state constraints," in Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, December, 2005.
- Patil, A., and Shah, G. (2021). "Discrete time consensus algorithm in multi-agent system," in 2021 Seventh Indian Control Conference (ICC), Mumbai, India, December, 2021. doi:10.1109/icc54714.2021.9702911
- Peng, X., Wang, Q., and Xiong, S. (2020). Distributed leader-follower consensus tracking control for fixed-wind uavs with positive linear speeds under directed graphs. *IFAC-PapersOnLine* 53 (5), 487–490. doi:10.1016/j.ifacol.2021.04.134
- Sadeghzadeh-Nokhodberiz, N., Can, A., Stolkin, R., and Montazeri, A. (2021). Dynamics-based modified fast simultaneous localization and mapping for unmanned aerial vehicles with joint inertial sensor bias and drift estimation. *IEEE Access* 9, 120247–120260. doi:10.1109/access.2021.3106864
- Sadeghzadeh-Nokhodberiz, N., Iranshahi, M., and Montazeri, A. (2023). Vision-based particle filtering for quad-copter attitude estimation using multirate delayed measurements. *Front. Robotics AI* 10, 1090174. doi:10.3389/frobt.2023.1090174
- Sadeghzadeh-Nokhodberiz, N., and Meskin, N. (2023). Consensus-based distributed Formation Control of multi-quadcopter systems: barrier lyapunov function approach. *IEEE Access* 11, 142916–142930. doi:10.1109/access.2023.3340417
- Tang, Z.-L., Tee, K. P., and He, W. (2013). Tangent barrier Lyapunov functions for the control of output-constrained nonlinear systems. *IFAC Proc. Vol.* 46 (20), 449–455. doi:10.3182/20130902-3-cn-3020.00122
- Tee, K. P., and Ge, S. S. (2011). Control of nonlinear systems with partial state constraints using a barrier Lyapunov function. *Int. J. Control* 84 (12), 2008–2023. doi:10.1080/00207179.2011.631192
- Tee, K. P., Ge, S. S., and Tay, E. H. (2009). Barrier Lyapunov functions for the control of output-constrained nonlinear systems. *Automatica* 45 (4), 918–927. doi:10.1016/j.automatica.2008.11.017
- Tee, K. P., Ge, S. S., and Tay, F. E. H. (2008). Adaptive control of electrostatic microactuators with bidirectional drive. *IEEE Trans. control Syst. Technol.* 17 (2), 340–352. doi:10.1109/TCST.2008.2000981
- Yan, J., Guan, X., Luo, X., and Chen, C. (2017). Formation control and obstacle avoidance for multi-agent systems based on virtual leader-follower strategy. *Int. J. Inf. Technol. Decis. Mak.* 16 (03), 865–880. doi:10.1142/s0219622014500151
- Yasin, J. N., Mohamed, S. A. S., Haghighabayan, M. H., Heikkonen, J., Tenhunen, H., and Plosila, J. (2020). Unmanned aerial vehicles (uavs): collision avoidance systems and approaches. *IEEE access* 8, 105139–105155. doi:10.1109/access.2020.3000064

Frontiers in Robotics and AI

Explores the applications of robotics technology
for modern society

A multidisciplinary journal focusing on the theory
of robotics, technology, and artificial intelligence,
and their applications - from biomedical to space
robotics.

Discover the latest Research Topics

[See more →](#)

Frontiers

Avenue du Tribunal-Fédéral 34
1005 Lausanne, Switzerland
frontiersin.org

Contact us

+41 (0)21 510 17 00
frontiersin.org/about/contact



Frontiers in Robotics and AI

