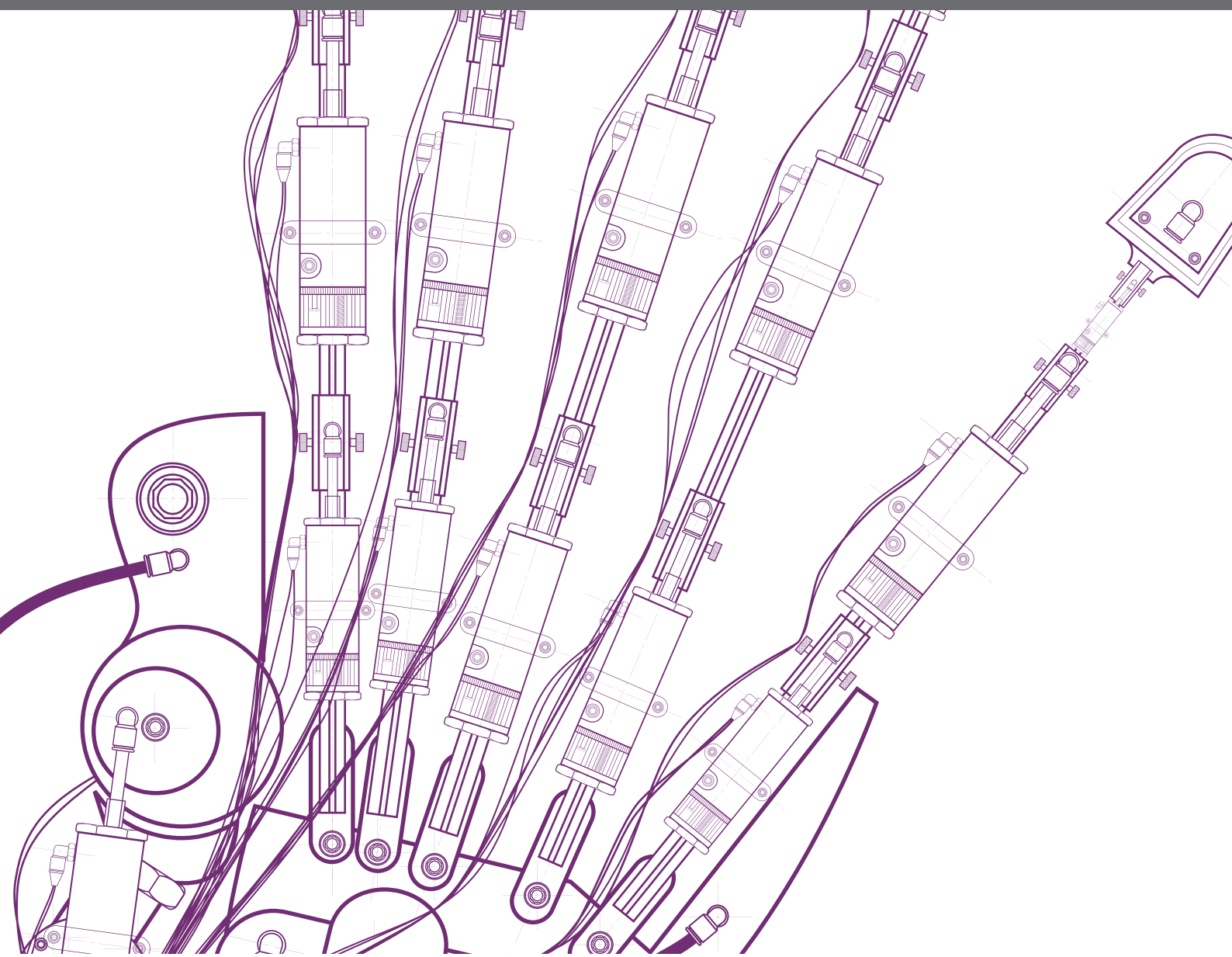
ADVANCES IN MODELLING AND CONTROL OF SOFT ROBOTS

EDITED BY: Concepción A. Monje and Cecilia Laschi

PUBLISHED IN: Frontiers in Robotics and AI



frontiers Research Topics

frontiers

ISSN 1664-8714
ISBN 978-2-88971-047-8
DOI 10.3389/978-2-88971-047-8

About Frontiers

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

Frontiers Journal Series

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

Dedication to Quality

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

What are Frontiers Research Topics?

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

ADVANCES IN MODELLING AND CONTROL OF SOFT ROBOTS

Topic Editors:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain
Cecilia Laschi, National University of Singapore, Singapore

Citation: Monje, C. A., Laschi, C., eds. (2021). Advances in Modelling and Control of Soft Robots. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88971-047-8

Table of Contents

frontiers in Robotics and AI

EDITORIAL
published: 21 May 2021
doi: 10.3389/frobt.2021.706514

Editorial: Advances in Modeling and Control of Soft Robots

Concepción Alicia Monje Micharet[1]* *and Cecilia Laschi*[2]

[1]Robotics Lab, Universidad Carlos III de Madrid, Madrid, Spain, [2]Department of Mechanical Engineering, National University of Singapore, Singapore, Singapore

Keywords: soft robotics, modeling of soft robots, control of soft robots, soft actuators, soft robotics applications

Editorial on the Research Topic

Advances in Modelling and Control of Soft Robots

The emerging field of soft robotics is nowadays looking at innovative ways to create and apply robotic technology in our lives. It is a relatively new domain in the field of robotics, but one that has a lot of potential to change how we relate with robots and also how they are used. In natural environments and human-centric operations, where safety and adaptability to uncertainty are fundamental requirements, soft robots may beneficially show these characteristics: they can conform to variable but sensitive environments, adaptively move, manipulate and grasp unknown objects varying in size and shape and their soft condition allows them to squeeze through confined spaces.

In addition to the many challenges and research achievements on the material side, actuation and sensing techniques, and fabrication technologies, the questions on how to model soft robots and how to control their movements are challenging scientifically and important from the application perspective. Classical control approaches in robotics are nonlinear-model-based. However, the highly complex and nonlinear models necessary for a soft robotic system make this approach a difficult task and therefore seem to come to a limit in the presence of soft robots. Therefore, other methods have been applied seemingly being more useful in this context, such as learning-based control algorithms, model-free approaches like bang-bang control, control algorithms motivated by neuroscience, or morphological computation. These methods add new perspectives to the well-known model-based approach.

Such research challenges and the current achievements in the field have been discussed by the soft robotics community in a second forum on this topic, at the second workshop on "Advances in Soft Robots Control", held on November 4th, 2019, in Macau, China, during the 2019 IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS 2019). The workshop wanted to answer questions like "Do we have to rethink the basic approach in robot control, which is model-based, when it comes to controlling soft robotic systems?" The papers collected in this issue come from that discussion and compose the pictures of the achievements presented, with extensions following the discussion and analysis done in that interactive context. They cover achievements from the theoretical modeling of soft robots to their control, up to specific application-driven developments.

A few works answer the workshop question by rethinking the modeling approaches and techniques. They address dynamic modeling and related control of soft robots, starting from the current approaches based on statics, or second-order dynamics, and model predictive control (MPC), using basic lumped-parameters. Thuruthel et al. show how the dynamic model of a soft robot can be reduced to first-order dynamical equation, thanks to high damping and low inertia, with minimal loss in accuracy. The work by Hyatt et al. demonstrates that online model adaptation is key in soft robot dynamic modeling and shows their results with a model predictive control. It is based on

OPEN ACCESS

Edited and reviewed by:
Cosimo Della Santina,
Delft University of Technology,
Netherlands

***Correspondence:**
Concepción Alicia Monje Micharet
cmonje@ing.uc3m.es

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *07 May 2021*
Accepted: *10 May 2021*
Published: *21 May 2021*

Citation:
Monje Micharet CA and Laschi C (2021) Editorial: Advances in Modeling and Control of Soft Robots. Front. Robot. AI 8:706514. doi: 10.3389/frobt.2021.706514

the widely adopted piecewise constant curvature (PCC) assumption and shows an adaptive behavior, thanks to a model reference adaptive control. Dutra Gollob et al. present a model for predicting the output force profile of their vacuum-powered soft actuators, that uses a simplified geometrical approach and the principle of virtual work. The paper by Suphapol Diteesawat et al. addresses the specific case of electro-ribbon actuators, promising in soft robotics and challenging for control, as they exhibit pull-in instability and a very narrow contraction range for feedforward control: small contraction below the pull-in voltage threshold, complete contraction above that. The authors can access intermediate steady-states, not accessible using traditional feed-forward control, with a time-varying voltage profile that starts above pull-in threshold but is reduced afterward.

Schiller et al. move the focus on the whole robot body. They control the gait of a multi-limb robot by closing the control loop in Cartesian space, under the assumption of constant curvature (CC) and by reducing the joint space dimension from nine to two, describing the robot velocity space, i.e., the walking speed and the rotational speed. Angelini et al. also take a higher view and introduce a hierarchical, two-level, control architecture that takes neuroscience findings to ensure natural movements in articulated soft robots, such as learning by repetition, anticipatory behavior, reactive re-planning. It combines the low level of dynamic inversion and trajectory tracking with the high level that manages the degree of freedom (DOF) redundancy, allowing to control the system through a reduced set of variables.

Another way to rethink the basic approach in robot control, as in the workshop question, is by moving from model-based to model-free approaches. In the work by Al-Ibadi et al., a neural network (NN) controller laid in parallel with a proportional controller (P) tracks the non-linear behavior (elongation and bending) of a pneumatic muscle actuator (PMA). The parallel neural network proportional (PNNP) controllers provide a high level of precision and fast-tracking control system.

Some other works address the workshop question by outlining the importance of sensing and showing its instrumental role in control. Ibrahim et al. add sensing (an inertial measurement unit (IMU) and pressure sensing) to their fiber-reinforced actuator to couple it with PCC modeling and close the control loop on pressure and chamber lengths. Rupert et al. also address sensing and propose methods for placing length sensors on a soft continuum robot joint and for configuration estimation, with drastic error reduction. Chen et al. instead propose a different viewpoint and show how we can use soft arm compliant behavior to gain useful information. They show how they can estimate external loads acting on their soft arm, using a static model and controller.

Modeling is also relevant for rethinking soft robot design and improve their overall performance and usability. Lee et al. propose a design methodology for soft grippers that are customized to grasp single dedicated objects. They propose a fabrication method that can rapidly customize and fabricate soft grippers, thanks to a simplified analytical model based on geometric approximations and pseudo-rigid-body modeling theory. Yoder et al. address the design of prosthetic limbs with the aim of more closely mimicking intact neuromuscular systems and improve the capabilities of prosthetic users. They evaluate the performance of a hydraulically amplified self-healing electrostatic (HASEL) soft actuator, by using a kinematic model of the prosthetic finger to inform the design of their improved Peano-HASEL actuator with the goal of increasing the fingertip pinch force of the prosthetic finger.

This collection of papers provides a useful insight on recent, diverse, approaches to soft robot modeling and control. It shows how important scientific questions are addressed by the lively and productive scientific community in this field and outlines the scientific challenges that are still open and provide interesting opportunity for further research and progress.

AUTHOR CONTRIBUTIONS

CM was the main organizer of the workshop from which this Research Topic comes from, she prepared the call text and revised this Editorial. CL contributed to the workshop and topic organization and drafted the Editorial.

FUNDING

The research leading to these results has received funding from the HUMASOFT project, with reference DPI2016-75330-P, funded by the Spanish Ministry of Economy and Competitiveness, and from RoboCity2030-DIH-CM, Madrid Robotics Digital Innovation Hub (Robótica aplicada a la mejora de la calidad de vida de los ciudadanos, FaseIV; S2018/NMT-4331), funded by "Programas de Actividades I+D en la Comunidad de Madrid" and cofunded by Structural Funds of the EU.

ACKNOWLEDGMENTS

The authors wish to acknowledge the contribution of speakers and attendees of the IEEE/RSJ IROS 2019 Workshop on "Advances in Modeling and Control of Soft Robots".

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.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 02 July 2020
doi: 10.3389/frobt.2020.00087

A Gait Pattern Generator for Closed-Loop Position Control of a Soft Walking Robot

Lars Schiller[1]*, *Arthur Seibel*[2] and *Josef Schlattmann*[1]

[1] Workgroup on System Technologies and Engineering Design Methodology, Hamburg University of Technology, Hamburg, Germany, [2] Fraunhofer Research Institution for Additive Manufacturing Technologies IAPT, Hamburg, Germany

This paper presents an approach to control the position of a gecko-inspired soft robot in Cartesian space. By formulating constraints under the assumption of constant curvature, the joint space of the robot is reduced in its dimension from nine to two. The remaining two generalized coordinates describe respectively the walking speed and the rotational speed of the robot and define the so-called velocity space. By means of simulations and experimental validation, the direct kinematics of the entire velocity space (mapping in Cartesian task space) is approximated by a bivariate polynomial. Based on this, an optimization problem is formulated that recursively generates the optimal references to reach a given target position in task space. Finally, we show in simulation and experiment that the robot can master arbitrary obstacle courses by making use of this gait pattern generator.

Keywords: mobile robotics, gait pattern generator, closed-loop position control, gecko-inspired soft robot, locomotion controller

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Cosimo Della Santina, Massachusetts Institute of Technology, United States
Chaoyang Song, Southern University of Science and Technology, China

***Correspondence:**
Lars Schiller
lars.schiller@tuhh.de

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *18 February 2020*
Accepted: *02 June 2020*
Published: *02 July 2020*

Citation:
Schiller L, Seibel A and Schlattmann J (2020) A Gait Pattern Generator for Closed-Loop Position Control of a Soft Walking Robot. Front. Robot. AI 7:87. doi: 10.3389/frobt.2020.00087

1. INTRODUCTION

Soft robotics is an emerging field in the robotics sciences and enjoys increasing attention in the scientific community (Bao et al., 2018). An important part of this field is mobile soft robotics, which allows locomotion in unknown and unstructured (Katzschmann et al., 2018) as well as potentially dangerous environments (Tolley et al., 2014). In order to navigate a robot through any environment, some sort of feedback is needed. As discussed in Santina et al. (2017), high gain feedback control results in good tracking performance, but imposes a reduction in the compliance of the controlled system. Therefore, it takes away the essential characteristic and greatest advantage of a soft robot—its softness (Rus and Tolley, 2015). When it comes to soft robots, usually the dynamics of inputs are indirectly coupled with the dynamics of outputs and the coupling is time-delayed (PneuNets: pressure—angle, SMA: heat—contraction, refer to Lee et al., 2013). In order to take this into account, a cascaded control architecture has been established (see, e.g., Marchese et al., 2014; Hofer and D'Andrea, 2018). In the case of pneumatically operated robots, the inner loop controls the pressure and the outer loop controls the pressure reference (see also **Figure 11B**). In order to preserve softness, the feedback gain of the outer control loop needs to be low. Most of the pressure reference should therefore be generated by a feed forward term (Santina et al., 2017). There is a trend to implement the feed forward term by using Iterative Learning Control (Bristow et al., 2006; see, e.g., Santina et al., 2017; Zhang and Polygerinos, 2018; Hofer et al., 2019). As shown in Santina et al. (2020), the typical soft properties of a soft robot can also be preserved with a model-based feed forward term when doing position control.

All the soft robots discussed so far are stationary. Thus, position control refers to the position of the end effector and not to the position of the entire robot. However, the same principles are also valid for mobile soft robots. Most mobile soft robots, such as in Shepherd et al. (2011), Godage et al. (2012), Tolley et al. (2014), Qin et al. (2019), and Schiller et al. (2019), are feed forward-controlled with predefined gait patterns. In order to enable such robots to move autonomously even in unknown terrain, a locomotion controller is needed that can generate any gait path. For solving this task, different methods have been employed, such as sine generators, central pattern generators (CPG), predefined trajectories, finite state machines, or heuristic control laws (Pratt et al., 2001). An example for sine generator-based locomotion control is presented in Horvat et al. (2015, 2017) for a salamander-like robot. The method enables to operate the robot by only two drive signals, i.e., forward and rotational speed. The main contribution here is the skilful synchronization of spine and legs motion, which is very robot-specific. In Ijspeert (2008), the suitability of central pattern generators, i.e., biologically inspired neural circuits capable of producing coordinated patterns for robot's locomotion, are discussed. It is concluded that CPGs are well-suited in general and especially for distributed implementations (e.g., for snake-like or reconfigurable robots). However, there is neither a sound design methodology to solve a specific locomotor problem nor a solid theoretical foundation. In order to implement CPGs in a meaningful way, the basic gait pattern must therefore be known from the outset, which again is robot-specific. An example for the automatic generation of optimal joint-trajectories is given in Bern et al. (2019). By using a forward shooting method and an FEM-based direct kinematics simulation, high-level goals, such as forward speed or direction of movement of various soft walking robots can be met. This method does not require a priori knowledge of a motion pattern, but can not be used online without restrictions (computation time, stability, ...). However, it can be well used to find robot-specific gait patterns.

Hence, for locomotion control of a robotic platform, a robot-specific motion strategy must be known. This paper analytically derives a robot-specific mapping of desired motion (forward and rotational speed) to joint coordinates for the gecko-inspired robot from Schiller et al. (2019), which is briefly described in section 2. The mapping function is referred to as "gait law" and is presented in section 3. In section 4, the direct kinematics of the robot are approximated by a polynomial by means of simulation and experiments to allow a fast evaluation. This is necessary to implement a control strategy in section 5 that maintains the softness of the robot and allows it to approach arbitrary references in the task space. The control strategy is referred to as Gait Pattern Generator. **Figure 1** shows the systematic procedure of this paper. To summarize, the paper contributes in two ways: (i) it derives the robot-specific motion strategy for the gecko-inspired robot and (ii), for a given robot-specific motion strategy, it provides a method to control the robot's position. However, the underlying assumptions of the former can also be transferred to other soft robots, since the ability to adapt to the environment is exploited herein.

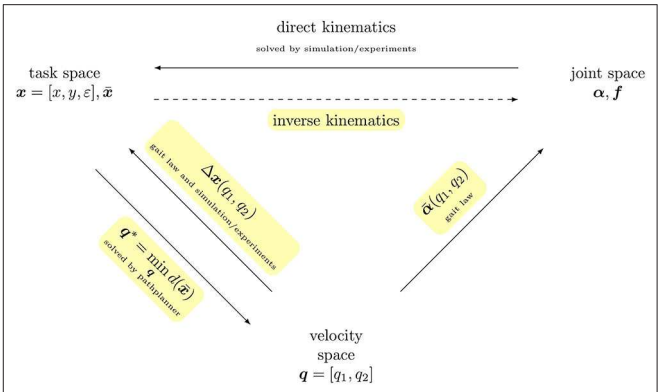


FIGURE 1 | Overview of spaces: in order to approximate the inverse kinematics, the joint space of the robot is reduced by formulating constraints referred to as gait law $\bar{\boldsymbol{\alpha}}$. The remaining two generalized coordinates $\boldsymbol{q}$ define the so-called velocity space. The direct kinematics of the entire velocity space (mapping in Cartesian task space) is approximated by a bivariate polynomial $\Delta \boldsymbol{x}$. By formulating an optimization problem $\min d(\bar{\boldsymbol{x}})$ that recursively generates the reference minimizing the distance to a given target position $\bar{\boldsymbol{x}}$, the robot can be operated in task space.

2. ROBOT AND EXPERIMENTAL SETUP

The soft robot this paper deals with has five limbs (four legs and a torso) and four feet that can be operated independently. Therefore, its joint space has nine dimensions: the five bending angles of the limbs $\boldsymbol{\alpha} = [\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4]$ and the four states of fixation actuators $\boldsymbol{f} = [f_0, f_1, f_2, f_3]$. Since its locomotion is only possible within two dimensions, its description in task space needs only three coordinates: the x and y position of the robot ${}^O\boldsymbol{x}$ and its orientation ${}^O\varepsilon$, described in the global (Cartesian) coordinate system $\{O\}$. Thus, the task space has three dimensions. A photograph of the prototype of this robot is depicted in **Figure 2A** and **Table 1** summarizes its specifications. In order to evaluate the performance of the robot, the test bench shown in **Figure 2B** was built with an embedded camera system. To measure the bending angles $\boldsymbol{\alpha}$, the robot orientation ε, and the robot position $\boldsymbol{x}$, apriltags (Wang and Olson, 2016) were fixed on its body. For a more detailed description of the experimental setup, refer to the **Supplementary Material**.

3. GAIT LAW

The straight gait of the robot can be described by a single variable—the reference bending angle of the torso $\bar{\alpha}_2$. All other variables of the joint space can then be described as a function of $\bar{\alpha}_2$ by means of the gait law for the straight gait, which was derived in Seibel and Schiller (2018):

$$\bar{\boldsymbol{\alpha}}_{\text{straight}} = \begin{bmatrix} \frac{\pi}{4} - \frac{\bar{\alpha}_2}{2} \\ \frac{\pi}{4} + \frac{\bar{\alpha}_2}{2} \\ \bar{\alpha}_2 \\ \frac{\pi}{4} - \frac{\bar{\alpha}_2}{2} \\ \frac{\pi}{4} + \frac{\bar{\alpha}_2}{2} \end{bmatrix}, \boldsymbol{f} = \begin{bmatrix} 0 \text{ if } \bar{\alpha}_2 < 0 \text{ else } 1 \\ 1 \text{ if } \bar{\alpha}_2 < 0 \text{ else } 0 \\ 1 \text{ if } \bar{\alpha}_2 < 0 \text{ else } 0 \\ 0 \text{ if } \bar{\alpha}_2 < 0 \text{ else } 1 \end{bmatrix}. \quad (1)$$

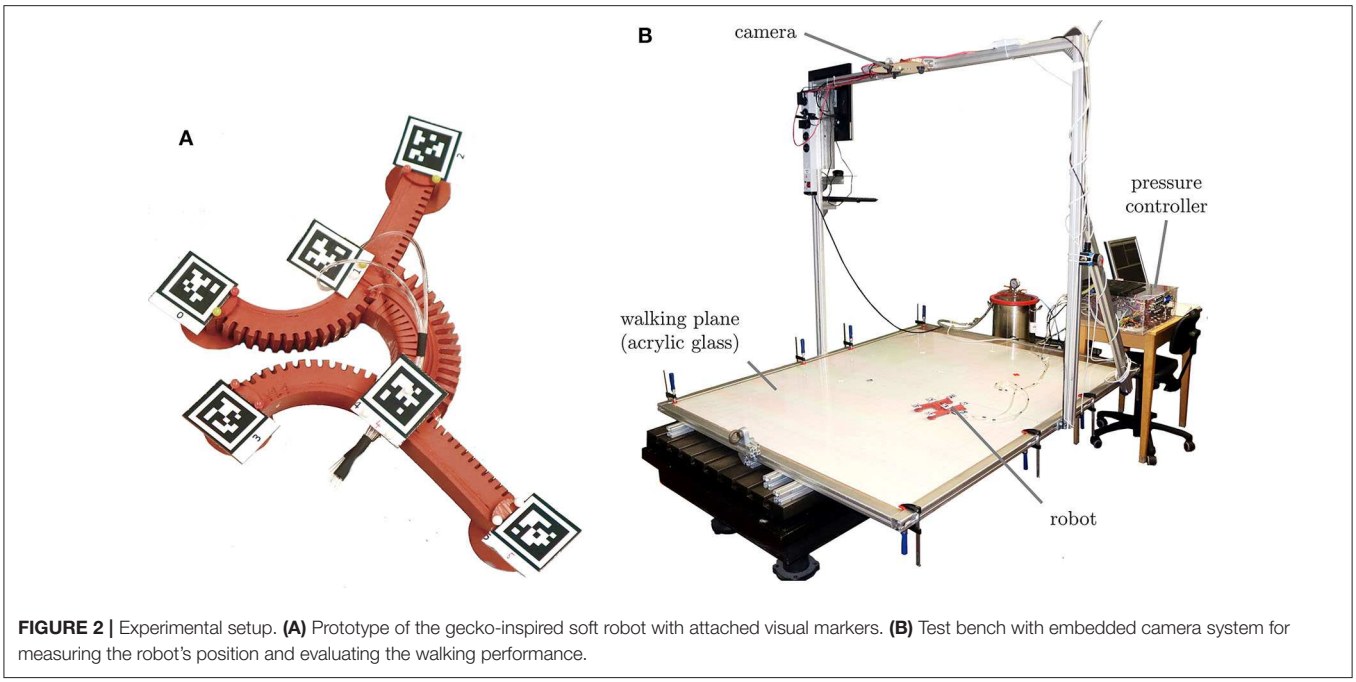


FIGURE 2 | Experimental setup. **(A)** Prototype of the gecko-inspired soft robot with attached visual markers. **(B)** Test bench with embedded camera system for measuring the robot's position and evaluating the walking performance.

TABLE 1 | Specifications of the soft robot.

Total weight	150 g
Max. speed	6 cm/s
Body length	12 cm
Body span	17 cm
Average applied pressure	0.76 bar
Pull-off force of suction cups	47 N

For a constant cycle time, the torso's bending angle is the essential measure for the forward velocity. Therefore, q_1 as the signal driving the forward velocity is introduced, and for straight gait, $\bar{\alpha}_2 = q_1$ is set. In order to operate the robot with different velocities, the angle reference $\bar{\boldsymbol{\alpha}}(q_1)$ for a given step size q_1 is inverted after a certain time interval t_{move}. Hence, it jumps from $\bar{\boldsymbol{\alpha}}(q_1)$ to $\bar{\boldsymbol{\alpha}}(-q_1)$. The corresponding fixation reference $\boldsymbol{f}$ must also be inverted.

3.1. Derivation for General Case

The above gait law can only generate gait patterns for straight motion. It is based on the idea that the orientations of the feet always remain constant. Now, we will loosen this restriction and demand only constant orientations for the fixed feet, while the unfixed feet are allowed to rotate. This implies two cases to be considered:

1. What should be the rule for a fixed foot so that its orientation remains constant regardless of the rotation of the body?
2. What should be the rule for a free foot so that its change of orientation matches that of the body and enables a suitable initial pose for the next cycle?

For both cases, the rules are based on the change of orientations of the feet. The orientations of feet $\boldsymbol{\varphi} = [\varphi_0\ \varphi_2\ \varphi_3\ \varphi_5]^\top$ described in the global coordinate system—and consequently their change during the change of pose—can be calculated assuming constant curvature as follows:

$$\boldsymbol{\varphi}(\boldsymbol{\alpha}, \varepsilon) = \begin{bmatrix} \varepsilon - \frac{1}{2}\alpha_2 - \alpha_0 \\ \varepsilon - \frac{1}{2}\alpha_2 + \alpha_1 \\ \varepsilon + \frac{1}{2}\alpha_2 + \alpha_3 + \pi \\ \varepsilon + \frac{1}{2}\alpha_2 - \alpha_4 + \pi \end{bmatrix}. \tag{2}$$

Since the feet's orientations depend on the robot's orientation ε and the bending angles $\boldsymbol{\alpha}$, a description for the latter two is required. First, it will be discussed how to describe and how to change the walking direction of the robot ε, i.e., its orientation. From Schiller et al. (2020), it is known that the asymmetrical actuation of the torso leads to a rotation of the body. In order to describe an asymmetrical actuation, the steering factor q_2 is introduced. The reference angle for the torso $\bar{\alpha}_2$ is then described as follows:

$$\bar{\alpha}_2 = q_1 + |q_1| q_2, \tag{3}$$

where $q_2 \in [-0.5, 0.5]$ is dimensionless and shifts the reference angle of the torso $\bar{\alpha}_2$ in the direction of q_2 (compare **Figure 3**). In this way, the left side of the torso is actuated by $|q_1| q_2$ more in the first half of a cycle and the right side by the same amount less in the second half of the cycle. It should be noted that Equation (3) describes only one possible model for asymmetric actuation. Several models have been tested and this one has been established. Clearly, the change of orientation per cycle $\Delta\varepsilon$ is related to the steering factor q_2 and the step length q_1. Simulation and experiment show that, for asymmetric

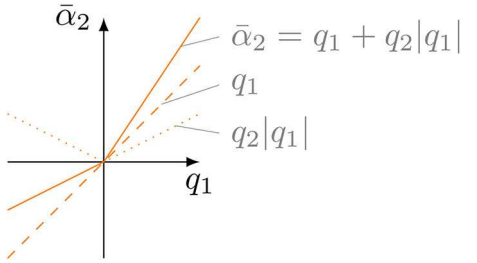


FIGURE 3 | Illustration of how the steering q_2 influences the reference angle of the torso $\bar{\alpha}_2$.

actuation with positive q_2, a negative change of orientation occurs, and vice versa. The change in orientation per cycle is therefore negatively proportional to the steering factor and the step length:

$$\Delta\varepsilon \sim -|\Delta q_1| q_2 \,,$$

where $|\Delta q_1|$ is the amount of change in torso actuation from initial pose to subsequent pose. This results in a model for orientation change per cycle $\Delta\hat{\varepsilon}$ of the body:

$$\frac{1}{2}\Delta\hat{\varepsilon} = -\tilde{c}_1 |\Delta q_1| q_2 \,, \tag{4}$$

where the robot-specific constant $\tilde{c}_1$ describes the ability of the robot to rotate. Here, it is assumed that the robot rotates consistently within the cycle. Therefore, the change in orientation after a pose change (half cycle) is exactly half as much as after the entire cycle; compare to **Figure 4** where $\frac{1}{2}\Delta\varepsilon = \varepsilon_1 - \varepsilon_0 = \varepsilon_2 - \varepsilon_1$.

The second parameter for calculating the feet's orientations Equation (2) is the bending angles of the legs. Hence, a specification for the legs is needed. The structure of the straight gait law from Equation (1) was adopted for this purpose, whereby the reference angles of the legs are extended with a yet unknown term $g(q_1, q_2)$. In the following, the procedure is shown for the front left leg only (α_0). However, it can be transferred to all other legs. With this extension, the reference angle for the front left leg results in:

$$\bar{\alpha}_0 = \frac{\pi}{4} - \frac{\bar{\alpha}_2}{2} + g(q_1, q_2) \,. \tag{5}$$

Now, the change of foot orientation when changing the pose $\Delta\varphi$ can be derived from Equations (2)–(5) by treating the references of the bending angles as the actual bending angles and assuming the body rotates according to the model from Equation (4):

$$\begin{aligned} \Delta\varphi_0 &= \varphi_{0,1} - \varphi_{0,0} \\ &= \left(\varepsilon_1 - \tfrac{\bar{\alpha}_{2,1}}{2} - \bar{\alpha}_{0,1}\right) - \left(\varepsilon_0 - \tfrac{\bar{\alpha}_{2,1}}{2} - \bar{\alpha}_{0,0}\right) \\ &= \left(\varepsilon_1 - \tfrac{\pi}{4} - g(q_{1,1}, q_2)\right) - \left(\varepsilon_0 - \tfrac{\pi}{4} - g(q_{1,0}, q_2)\right) \,, \\ &= (\varepsilon_1 - \varepsilon_0) - \left(g(q_{1,1}, q_2) - g(q_{1,0}, q_2)\right) \\ &= \tfrac{1}{2}\Delta\hat{\varepsilon} - \Delta g(q_1, q_2) \end{aligned} \tag{6}$$

where $q_{1,0}$ describes the step length of the initial pose and $q_{1,1}$ that of the subsequent pose. When changing poses, the robot always jumps from $\bar{\boldsymbol{\alpha}}(q_1, \cdot)$ to $\bar{\boldsymbol{\alpha}}(-q_1, \cdot)$. Therefore, $q_{1,1} = -q_{1,0}$ and $g(q_{1,1}, q_2) - g(q_{1,0}, q_2)$ can be combined to $\Delta g(q_1, q_2)$. Furthermore, it is assumed that the steering factor q_2 remains unchanged when changing poses. Next, a specification for the additional term $g(q_1, q_2)$ is derived for the two cases under consideration (fixed and unfixed leg).

3.1.1. Fixed Leg

Figure 4 shows one cycle of trotting gait. Within the transition from the initial pose (black) to the middle pose (gray), the front left foot is fixed and thus its orientation should remain constant. The bending angle must be determined in such a way that the foot's orientation is kept constant, i.e., independent of q_1 or q_2:

$$\Delta\varphi_{0,f} = \frac{1}{2}\Delta\hat{\varepsilon} - \Delta g_f(q_1, q_2) = 0 \quad \forall\, q_1, q_2 \,, \tag{7}$$

where the index f denotes a fixed foot/leg. This means that the robot can change from any pose described by the general gait law to a subsequent pose without changing the orientation of its fixed feet, with the limitation that the steering factor q_2 remains constant with this change. From Equations (7) and (4), the additional term for the fixed leg results in:

$$\Delta g_f(q_1, q_2) = \frac{1}{2}\Delta\hat{\varepsilon} = -\tilde{c}_1 |\Delta q_1| q_2 \,. \tag{8}$$

Since the sign of q_1 is always swapped when changing poses, the change of the torso actuation always results in $|\Delta q_1| = 2|q_1|$, and thus, the additional term becomes (with $c_1 = -4\tilde{c}_1$):

$$g_f(q_1, q_2) = c_1 |q_1| q_2 \,. \tag{9}$$

Inserted in Equation (5), the reference for a fixed leg results in:

$$\alpha_{0,f} = \frac{\pi}{4} - \frac{q_1}{2} - \frac{1}{2}|q_1| q_2 + c_1 |q_1| q_2 \,. \tag{10}$$

3.1.2. Free Leg

As the foot was previously fixed, the rotation of the body must affect its orientation in the non-fixed phase. The free foot should therefore rotate in the unfixed phase exactly as much as the body does during the entire cycle. This is illustrated in **Figure 4** where the change in orientation of the front left foot between the final pose (lightgray) and the middle pose (gray) matches exactly the rotation of the body $\Delta\hat{\varepsilon}$. With the model for the change of foot orientation from Equation (6), it must hold:

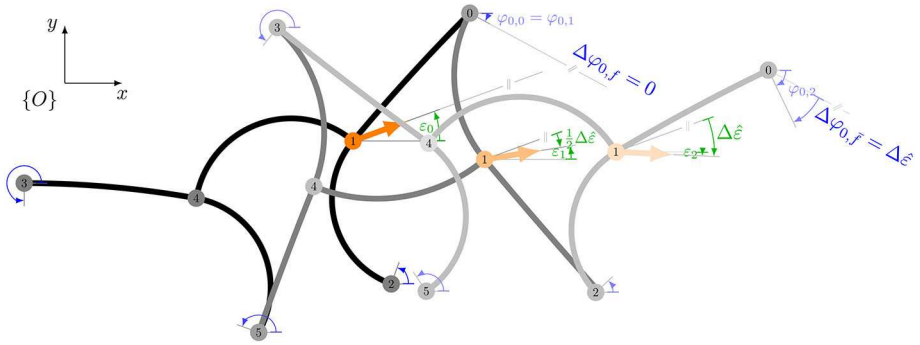


FIGURE 4 | Problem statement: which bending angles must be applied in order to turn the robot while keeping the orientation of its fixed feet? During the first change of pose, the orientation of the front left foot (φ_0) and the rear right foot (φ_5) should be kept constant. During the second change of pose, the front right foot (φ_2) and the rear left foot (φ_3) should not rotate.

$$\Delta\varphi_{0,\bar{f}} = \frac{1}{2}\Delta\hat{\varepsilon} - \Delta g_{\bar{f}}(q_1, q_2) = \Delta\hat{\varepsilon} \quad \forall\, q_1, q_2\,, \tag{11}$$

where $\bar{f}$ indicates an unfixed foot. Clearly, this only applies if the same additional term is added again, but with swapped sign:

$$g_{\bar{f}}(q_1, q_2) = -g_f(q_1, q_2) = -c_1|q_1|q_2\,. \tag{12}$$

According to Equation (5), the reference for a free leg results in:

$$\alpha_{0,\bar{f}} = \frac{\pi}{4} - \frac{q_1}{2} - \frac{1}{2}|q_1|q_2 - c_1|q_1|q_2\,. \tag{13}$$

If a foot is fixed, we add the term $g(q_1, q_2) = c_1|q_1|q_2$ to the reference angle of the corresponding leg. If the leg is free, the additional term g is subtracted. Whether a leg is fixed or not is determined by the sign of the torso reference (see Equation 1): q_1 positive $\rightarrow$ foot fixed, q_1 negative $\rightarrow$ foot free. Thus, the distinction between free and fixed leg can be avoided by dropping the amount operation of q_1 in the additional term g. The sign of q_1 then automatically controls the corrective direction of the additional term g. This procedure can be performed for all legs and results in the general gait law, which is formally described as follows:

$$\bar{\boldsymbol{\alpha}} = \begin{bmatrix} \frac{\pi}{4} - \frac{q_1}{2} - \frac{1}{2}|q_1|q_2 + c_1 q_1 q_2 \\ \frac{\pi}{4} + \frac{q_1}{2} + \frac{1}{2}|q_1|q_2 + c_1 q_1 q_2 \\ q_1 + |q_1|q_2 \\ \frac{\pi}{4} - \frac{q_1}{2} - \frac{1}{2}|q_1|q_2 + c_1 q_1 q_2 \\ \frac{\pi}{4} + \frac{q_1}{2} + \frac{1}{2}|q_1|q_2 + c_1 q_1 q_2 \end{bmatrix}, \boldsymbol{f} = (1)\,. \tag{14}$$

The value of additional leg bending c_1 is to be determined via simulations or experiments. This is demonstrated in the **Supplementary Material** and results in $c_1 = 1$. The visualization of this law is shown in **Figure 5**. Note that the middle layer shows the special case for straight motion from Equation (1). By introducing the index k specifying the extreme poses, references for a gait can be generated recursively by

$$\bar{\boldsymbol{\alpha}}_k = \bar{\boldsymbol{\alpha}}(-q_{1,k-1}, q_2), \boldsymbol{f}_k = \neg\boldsymbol{f}_{k-1}, \tag{15}$$

where $\neg\boldsymbol{f}$ is the logical negation of $\boldsymbol{f}$.

The gait law generates reference angles for the robot, depending on step length (forward velocity) q_1 and steering factor (rotational velocity) q_2. These two generalized coordinates define the so-called velocity space of trotting gaits, since each pair (q_1, q_2) describes another trotting gait. If q_1 and q_2 remain constant during gait, theoretically, the orientation of the fixed feet does not change. However, the derivation of this law did not examine whether the fixed feet also remain in position when switching poses. Also, the robot should have the ability to change its gait over time and should not always run the same circle with the same velocity. Therefore, q_1 and q_2 must vary. The next section examines whether this law provides useful references, despite neglecting the feet positions.

3.2. Experimental Validation

Within an experiment, it shall be analyzed whether the orientations of fixed feet actually remain constant during a cycle or ignoring the feet positions leads to significant discrepancies. The gait was slowed down ($t_{\text{move}} = 10\,\text{s}$) as highly dynamic changes smear the camera images and the tags can no longer be detected by image processing. **Figure 6** shows an exemplary cycle of a gait for $q_1 = 80°$ and $q_2 = -0.5$. The figure shows the mean values and standard deviations of five experiments in total. For the detailed processing steps in the evaluation, refer to the **Supplementary Material**. The upper graph shows the progression of the bending angles $\boldsymbol{\alpha}$ and the lower graph shows the progression of the orientations $\boldsymbol{\varphi}$ and ε during a cycle. Initially, all feet are fixed (pose 1a). The bending angle of the front left (red line) and the rear right leg (dark blue line) differs significantly from the reference at this point in time because the robot is forced into this pose by the fixation of its feet. After about five percent of the cycle time, the front left and rear right foot are released (pose 1b). At this point, a jump in the bending

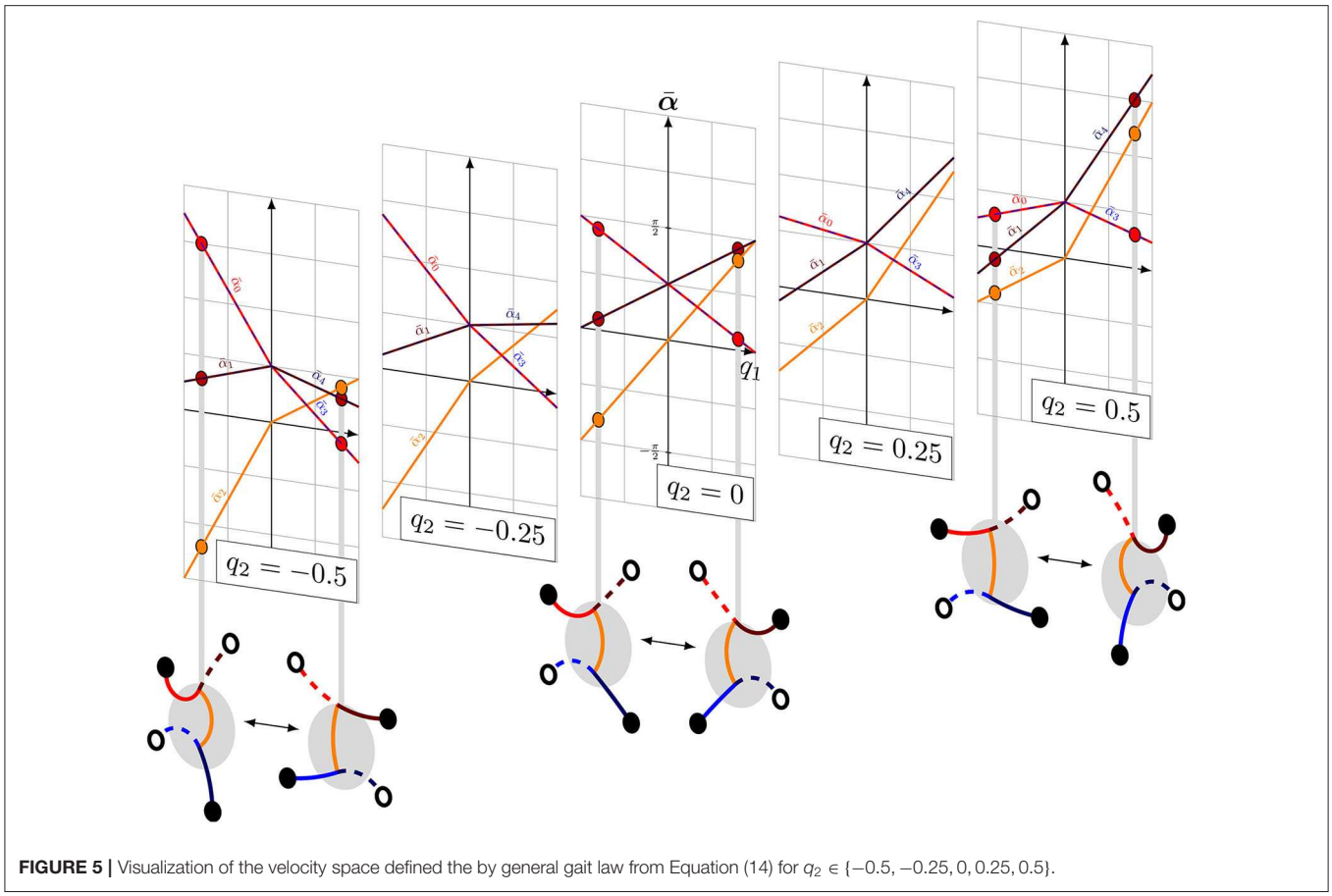


FIGURE 5 | Visualization of the velocity space defined the by general gait law from Equation (14) for $q_2 \in \{-0.5, -0.25, 0, 0.25, 0.5\}$.

angle of the two corresponding legs can be observed—the angles jump to their reference. The same effect can be observed when changing the feet fixation in the middle of the cycle (pose 2a $\rightarrow$ 2b). From this observation, it can be deduced that the robot cannot match the reference generated by the gait law because the closed kinematic chain of its parallel structure prevents it from adopting the specified bending angles. The $\boldsymbol{\varphi}$ graph shows that the orientations of the fixed feet remain nearly constant as assumed when deriving the gait law. An exception is the rear left foot (blue line): its orientation changes significantly during the fixed phase. As already seen in Schiller et al. (2020), the suction cups of the robot have a certain margin of rotation. This must now be utilized; otherwise, the feet would have to move (which is not possible because of the fixation). In summary, it can be concluded from the experiment in **Figure 6** that the gait law provides references which cannot be fully realized due to the closed kinematic chain, but nevertheless lead to the desired behavior.

4. APPROXIMATING THE DIRECT KINEMATICS

The next step is to determine how the robot behaves in the task space for each point (q_1, q_2) in the velocity space—that is, how it moves per cycle and by how much it rotates. Thus, the bivariate polynomial $\Delta\boldsymbol{x}(q_1, q_2)$ is searched for which approximates the transformation of the velocity space into the task space (compare **Figure 1**). The form of the polynomial is defined as follows:

$$\Delta\boldsymbol{x}(q_1, q_2) = \begin{bmatrix} \Delta\varepsilon \\ \Delta x \\ \Delta y \end{bmatrix}, \Delta\varepsilon, \Delta x, \Delta y := \sum_{i,j} a_{i,j} q_1^i q_2^j \,. \tag{16}$$

In order to identify the coefficients, the velocity space is gridded and for each set of values the motion of the robot is measured. This can either be done experimentally or the simulation model is used and the movement is simulated. The result of both approaches depends on the way they are implemented. Therefore, the influencing factors must be identified and their value must be meaningfully determined. **Table 2** summarizes the conditions under which the following experiments or simulation were carried out. A detailed discussion of the experimental conditions can be found in the **Supplementary Material**.

Figure 7 shows the results of simulation (**Figure 7A**) and experiment (**Figure 7B**). In both cases, the velocity space was gridded with $q_1 \in \{50, 60, \cdots, 90\}$ and $q_2 \in \{-0.5, -0.3, \cdots 0.5\}$ and a measurement was performed for each grid point. A simplified representation of the extreme poses of the resulting gait illustrates the movement. The tip of the torso of the initial pose is always at the position (q_1, q_2) and the orientation of

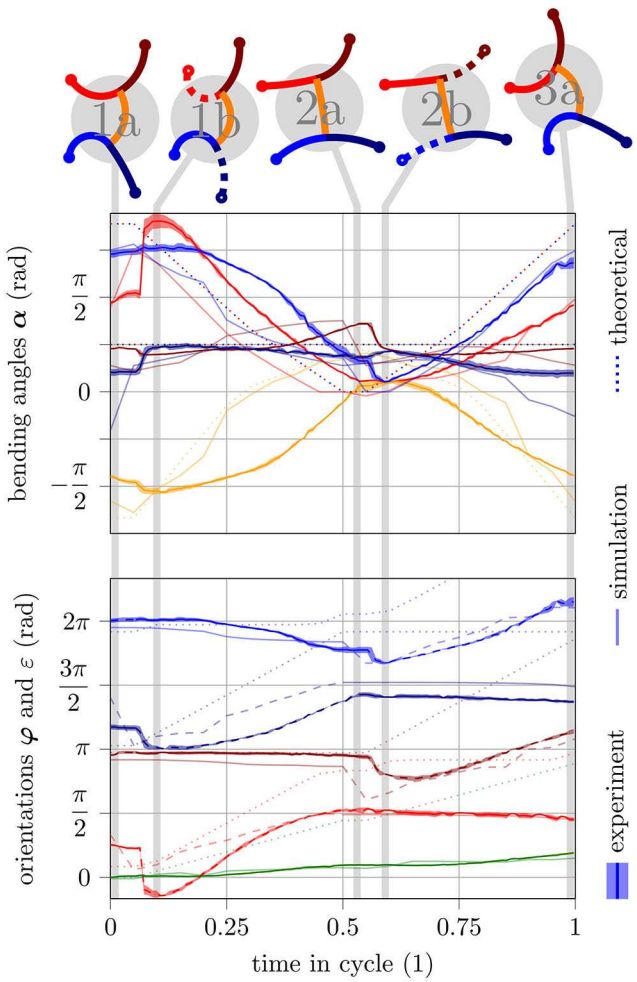


FIGURE 6 | Simulation and experiment of one gait cycle for $\boldsymbol{q} = [80° \; -0.5]$ and $c_1 = 1$. Theoretical values (according to the gait law) are illustrated with light dotted lines. Simulated values are illustrated with light solid lines. Experimental values are illustrated as solid lines together with an area indicating the standard deviation. In the orientation plot and the poses shown above, lines representing unfixed feet/legs are illustrated as dashed lines. The switch of fixation happens at half cycle time. Color code as follows: front left leg (red), front right leg (dark red), torso (orange), rear left leg (blue), rear right leg (dark blue), robot orientation (green).

TABLE 2 | Influencing factors on simulation and experiment.

Initial pose	$\boldsymbol{\alpha}_0 = \bar{\boldsymbol{\alpha}}(q_1, q_2), \boldsymbol{f}_0 = [1, 0, 0, 1]$
Number of cycles	min. $n_{\text{cyc}} = 2$, drop initial cycle
Dimensions of robot	$\ell_{\text{leg}} = 9.1\,\text{cm}$, $\ell_{\text{torso}} = 10.3\,\text{cm}$
Weighting parameters (only simulation)	$f_l, f_o, f_a = 89, 10, 5.9$
Model order of $\Delta\boldsymbol{x}(q_1, q_2)$	order = 2

the robot faces upwards. The resulting translation $[\Delta x \, \Delta y]^\top$ per cycle is indicated by a red arrow. Besides, the orientation of the robot after a cycle is represented by a green line. The heat map in the background shows the resulting rotation $\Delta\varepsilon(q_1, q_2)$ per cycle. The numerical value of this function is noted in a green box below the individual measurements. In the figure of the experiment (**Figure 7B**), the standard deviation of the translation is shown as a red ellipse with the corresponding semi axes. The standard deviation of the rotation is visualized as a light green triangle with an opening angle of 2std($\delta\varepsilon$). The blue arrow shows the polynomial fit of the translation and the blue line the polynomial fit of the rotation at the corresponding grid point. A detailed view of a single experiment is shown in **Figure 8**.

In contrast to the experiment in section 3.2, in **Figure 7**, a clear deviation between simulated and experimental results can be observed. The resulting rotation and the shift in transverse direction are noticeably higher for all grid points. The simulation model does not reproduce friction effects or external disturbances, such as the influence of the supply tubes. In the previous experiment (from section 3.2), these effects played a subordinate role because of the reduced speed and the relatively short distance traveled. This experiment was executed at full speed ($t_{\text{move}} = 1\,\text{s}$); thus, friction has a significantly larger influence. Furthermore, we can observe that the experiment is not symmetrical, meaning that swapping the sign of q_2 does not yield to mirrored behavior $[\Delta\boldsymbol{x}(q_1, q_2) \nsim -\Delta\boldsymbol{x}(q_1, -q_2)]$. This can be attributed to manufacturing inaccuracies of the robot and an optimizable pressure-bending angle calibration. The calibration procedure and associated difficulties are also discussed in the **Supplementary Material**. A final observation is that, in the experiment, the resulting rotation decreases for a large step length q_1. This is different to the simulation, where the resulting rotation increases steadily with increasing step length. For large values of q_1 and q_2, the gait law prescribes relatively large reference angles. If these are out of range of calibration of the respective actuator, the reference pressure is saturated to prevent damage to the robot. Exactly this effect occurs in the upper part ($q_1 \geq 80°$) of **Figure 7B**. Therefore, the poses here deviate much more from their simulated counterparts in **Figure 7A** than in the lower part of the figure ($q_1 < 80°$). Apart from the "over-simulation" and the missing saturation effect, the simulation reproduces the behavior very well. It can be seen as the behavior of a robot that has been perfectly manufactured and calibrated, consisting of actuators as robust as saturation is no longer necessary, whose feet have the optimum torsional stiffness, and where all friction effects have been reduced to a minimum. For the implementation of the Gait Pattern Generator, however, the actual interest focuses on the polynomial fit of the motion. In most cases, the second-order fit shown in blue matches the measurement or is at least within the standard deviation. The coefficients for the polynomial $\Delta\boldsymbol{x}(q_1, q_2)$ for Equation (16) are listed in **Table 3**.

5. GAIT PATTERN GENERATOR

The last step to control the robot's position is the calculation of the optimal tuple $\boldsymbol{q}^*$ to move from the current position $\boldsymbol{x}$ closer to a given target position $\bar{\boldsymbol{x}}$ (compare **Figure 1**).

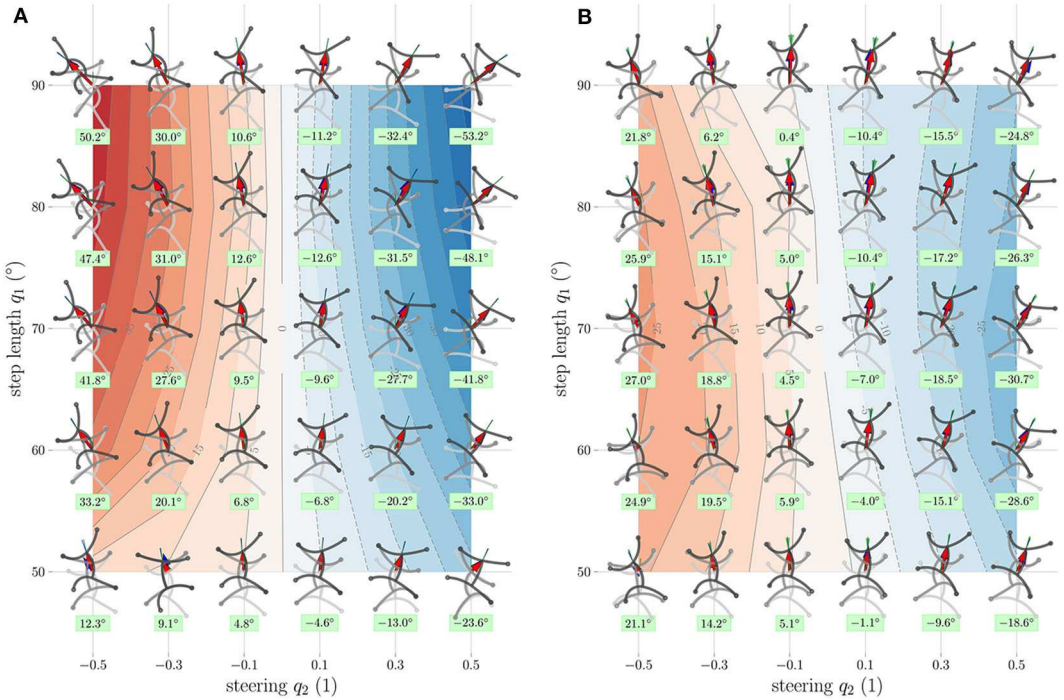


FIGURE 7 | Resulting experimental gaits according to the gait law in Equation (14) for a variation of step length and steering factor. The rows each have a constant step length q_1 and the columns a constant steering factor q_2. Each frame shows the resulting motion of one cycle with the pattern corresponding to (q_1, q_2). Below each frame, the rotation per cycle in degrees $\Delta\varepsilon$ is stated. The heat map in the background shows the polynomial fit of $\Delta\varepsilon$. The bold red vector pointing from the initial position of each individual gait to its end position is called $[\Delta x\ \Delta y]^\top$. **(A)** Simulation (89, 10, 5.9) and **(B)** experiment.

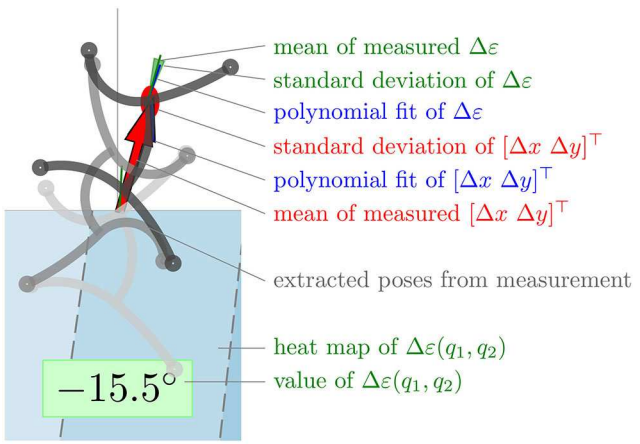


FIGURE 8 | Detailed view of resulting experimental gait according to the gait law for $(q_1, q_2) = (90°, 0.3)$.

5.1. Derivation

As derived in section 4, the robot turns around $\Delta\varepsilon$ and moves by $[\Delta x\ \Delta y]^\top$ with each cycle. Therefore, the position of the $(n+1)$th pose given in the coordinate system of the nth pose can be described by

$$ {}^{R(n)}\boldsymbol{x}_{(n+1)} = \begin{bmatrix} \Delta x(q_1, q_2) \\ \Delta y(q_1, q_2) \end{bmatrix}, \tag{17} $$

where the index n starts from 0 indicating the initial pose and accordingly the subsequent poses. If step length q_1 and steering factor q_2 do not change during gait ($\boldsymbol{q} = \text{const.}$), the translation and rotation per cycle will remain the same. Let us assume that it would be possible to reach the target position in a finite number of cycles without changing the gait. Accordingly, the vector ${}^{R(n)}\bar{\boldsymbol{x}}_{(n)}$ pointing from the nth pose to the target, can be described in the coordinate system of the nth pose as a function of the target vector of the $(n-1)$th pose:

$$ {}^{R(n)}\bar{\boldsymbol{x}}_{(n)} = \boldsymbol{R}(-\Delta\varepsilon)\left({}^{R(n-1)}\bar{\boldsymbol{x}}_{(n-1)} - {}^{R(n-1)}\boldsymbol{x}_{(n)}\right), \tag{18} $$

where $\boldsymbol{R} \in \mathbb{R}^{2\times 2}$ is the rotation matrix. Since for multiple rotations around the same axis $\boldsymbol{R}^k(x) = \boldsymbol{R}(kx)$ applies, this can be formulated explicitly:

$$ {}^{R(0)}\bar{\boldsymbol{x}}_{(n)} = \boldsymbol{R}(-n\Delta\varepsilon)\,{}^{R(0)}\bar{\boldsymbol{x}}_0 - \sum_{i=0}^{n} \boldsymbol{R}(-i\Delta\varepsilon)\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}. \tag{19} $$

Figure 9 visualizes these formulas, whereby the opacity of poses that lie further in the future decreases. Now, the distance d_n to the target position $\bar{\boldsymbol{x}}$ after n cycles of trotting with the pattern corresponding to the gait law $\bar{\boldsymbol{\alpha}}(q_1, q_2)$ can be calculated with

$$ d_n({}^{R(0)}\bar{\boldsymbol{x}}_0, \Delta\boldsymbol{x}) = \left|{}^{R(0)}\bar{\boldsymbol{x}}_{(n)}\right|_2. \tag{20} $$

TABLE 3 | Coefficients of the bivariate polynomial fit of the motion per cycle $\Delta \boldsymbol{x}$ for the experiment.

$f(q_1, q_2) =$	$a_{0,0}$	$a_{1,0} \cdot q_1$	$a_{0,1} \cdot q_2$	$a_{2,0} \cdot q_1^2$	$a_{0,2} \cdot q_2^2$	$a_{1,1} \cdot q_1 q_2$
$\Delta\varepsilon$ (°)	5.4154	−0.0457	−44.1944	−0.0006	0.778	−0.0832
Δx (cm)	0.1106	0.2225	11.4146	−0.0008	−17.5133	−0.1213
Δy (cm)	1.9498	−0.0682	−3.6997	0.0004	−0.0333	−0.0580

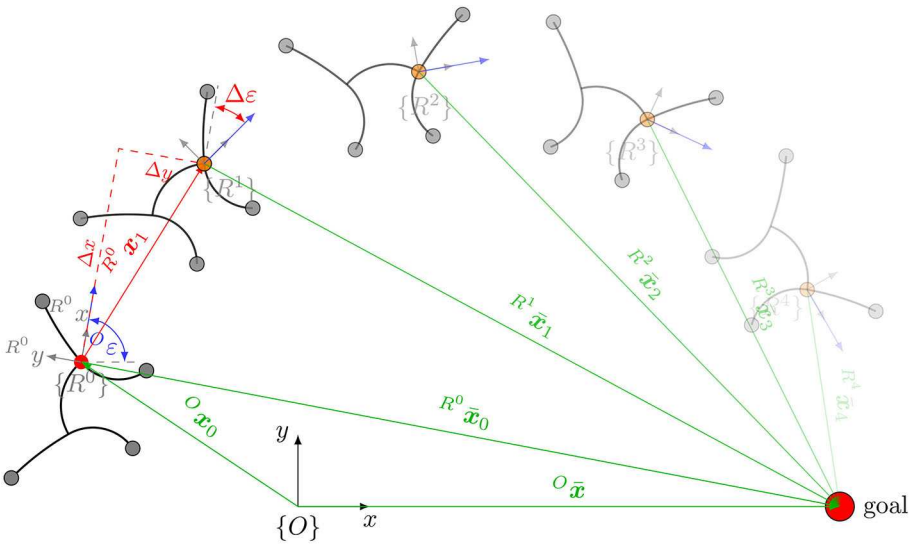


FIGURE 9 | Visualization of Equations (17)–(22). By using the approximation of the direct kinematics $\Delta \boldsymbol{x}$, the approximated position in Cartesian space after n cycles can be easily calculated. In the figure, the opacity decreases with increasing cycle number.

For a given target, the optimal tuple for n cycles can then be calculated as the minimum of the distance function

$$q_1^*, q_2^* = \min_{q_1, q_2 \in \mathcal{Q}} d_n^2 \left({}^{R^{(0)}}\bar{\boldsymbol{x}}_0, \Delta \boldsymbol{x} \right) , \tag{21}$$

where $\mathcal{Q}$ describes the set of feasible values for q_1 and q_2, respectively. Note that the vector ${}^{R^{(0)}}\bar{\boldsymbol{x}}_0$ describes the target position in the coordinate system of the initial pose. In the test bed with an external camera measurement system, this vector must be calculated from the measurements of the current pose ${}^{O}\boldsymbol{x}$ and the target position ${}^{O}\bar{\boldsymbol{x}}$:

$${}^{R^{(0)}}\bar{\boldsymbol{x}}_0 = \boldsymbol{R}(-{}^{O}\varepsilon) \left({}^{O}\bar{\boldsymbol{x}} - {}^{O}\boldsymbol{x} \right) . \tag{22}$$

However, the target measurement could also happen with a camera directly mounted on the robot without having to reformulate the equations, as the Gait Pattern Generator demands the target position in the robot coordinate system.

Figure 10 shows a visualization of the distance function d_n for different target points and the patterns corresponding to its minimum. In **Figure 10A**, the target is located slanted right in front of the robot and a planning horizon of $n = 1$ is considered. The minimum of the distance function is at full step length $q_1 = 90°$ and a medium steering factor $q_2 = 0.3$. The resulting reference allows the robot to move precisely to the front right. In **Figure 10B**, the target is located behind the robot. With a planning horizon of $n = 1$, the minimum distance results in the smallest allowed step length and steering. However, this solution does not bring the robot closer to the target, but it is the solution that minimizes the increase in distance. There is simply no gait pattern that can bring the robot closer to the target within only one cycle. For this reason, the planning horizon in **Figure 10C** was increased to $n = 4$. The minimum of $d_{n=4}$ is now at maximum step length and maximum steering factor for the same target position. The resulting reference leads to the desired behavior: the tightest possible right turn.

5.2. Implementation

As seen in the previous section, the distance to the target cannot always be reduced in just one cycle. The simplest strategy to solve this problem is to incrementally increase the planning horizon as long as the minimum possible distance to the target within the next n cycles $d_{n,\min}$ is larger than the current distance d_0. Furthermore, a strategy for transitioning between different gait patterns is required. So far, all simulations and experiments have only studied the motion of consistent gaits ($\boldsymbol{q} = \text{const.}$). However, the pattern generator should be able to dynamically change both step length and steering factor. The easiest way to make this possible is to assume that the robot is able to switch between any gait pattern, which means to allow all possible references regardless of the current pose. Here, it is questionable whether the output $\boldsymbol{q}^*$ actually minimizes the distance to the

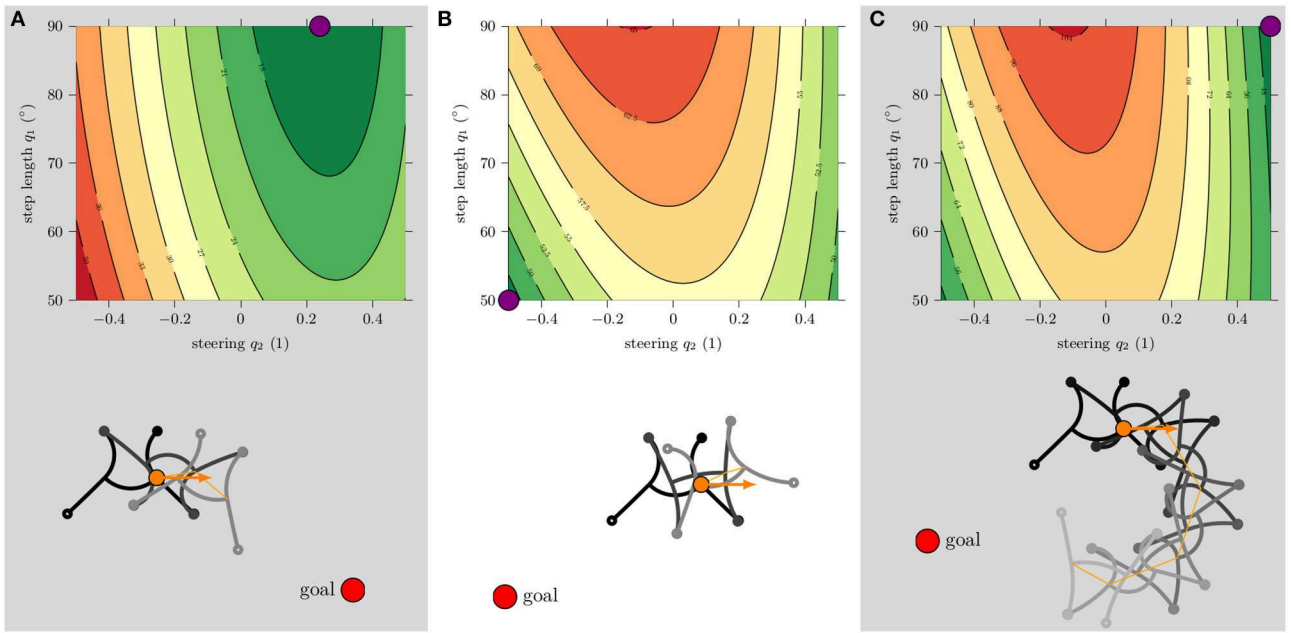


FIGURE 10 | Evaluation of the distance function d_n for different target positions $\bar{\boldsymbol{x}}$ and planning horizons n. The lowest values are represented by green and the highest values by red color. The lower image always shows the resulting simulated gait for n cycles, corresponding to the minimum distance (marked by a purple circle). Simulations were initialized with: ${}^{O}\boldsymbol{x} = (0, 0)$, ${}^{O}\varepsilon = 0°$, $\boldsymbol{\alpha}_0 = [90\ 0\ -90\ 90\ 0]$, $\boldsymbol{f}_0 = [1\ 0\ 0\ 1]$. **(A)** Planning horizon $n = 1$ for target at $\bar{\boldsymbol{x}} = (35, -20)$, **(B)** $n = 1$ for target at $\bar{\boldsymbol{x}} = (-35, -20)$, and **(C)** $n = 4$ for target at $\bar{\boldsymbol{x}} = (-35, -20)$.

target or whether another solution might be more suitable, since the calculation in most cases will be based on a different initial pose. Thus, it can be assumed that a different $\boldsymbol{q}^*$ would be calculated when considering the current pose of the robot. But by feeding back the current position after each step and a recalculation of the reference, reaching the target position can still be ensured. Algorithm 1 implements exactly this strategy. **Figure 11A** shows the procedure as a block diagram. The sampling rate of this control loop depends on the length of half a cycle and is slightly less than 1 Hz. The Gait Pattern Generator is paused as soon as the actual distance to the target is less than a defined value $\epsilon = 5\,\text{cm}$. For better comprehension, **Figure 11B** shows the low-level control architecture of the robotic system for a single actuator. Note that the simulation model mimics the coupled behavior of six of these blocks.

Algorithm 1: Gait Pattern Generator.

procedure GENERATE REFERENCE ($\bar{\boldsymbol{x}}_k$)
  $n \leftarrow 1$ ▷ start with planning horizon of 1 cycle
  $\boldsymbol{q}_k^*,\ d_{n,\min} \leftarrow \min_{\boldsymbol{q}} d_n\left(\bar{\boldsymbol{x}}, \Delta\boldsymbol{x}(\boldsymbol{q})\right)$ ▷ minimal distance to goal after 1 cycle
  while $d_0 > d_{n,\min}$ **do** ▷ stop if we get closer to goal
    $n \leftarrow n + 1$ ▷ increase planning horizon
    $\boldsymbol{q}_k^*,\ d_{n,\min} \leftarrow \min_{\boldsymbol{q}} d_n\left(\bar{\boldsymbol{x}}, \Delta\boldsymbol{x}(\boldsymbol{q})\right)$ ▷ minimal distance to goal after n cycles
  end while
  $q_{1,k}^* \leftarrow -\text{sign}(q_{1,k-1}^*)\, q_{1,k}^*$ ▷ switch sign of step length
  $\bar{\boldsymbol{\alpha}}_k \leftarrow \bar{\boldsymbol{\alpha}}\left(q_{1,k}^*, q_{2,k}^*\right)$ ▷ reference according to Eq. (14)
  $\boldsymbol{f}_k \leftarrow \neg\boldsymbol{f}_{k-1}$ ▷ switch fixation
  return $\bar{\boldsymbol{\alpha}}_k, \boldsymbol{f}_k$ ▷ next reference
end procedure

5.3. Experiments

In **Figure 12A**, the simulation for a list of four different target positions is shown. The next target position becomes active when $d_0 < \epsilon$ applies, i.e., the robot has almost reached the current target. In **Figure 12B**, the corresponding course of $\boldsymbol{q}$ is shown. It is clear to see that both values change over time. This proves that the robot can transition between different gait patterns, at least in the simulation. The same situation is now studied in the experiment shown in **Figure 12C** where the tracks of the tags of five independent experiments are overlaid. The difference between the right and left curves is significant. While the right-hand curves have a relatively small radius, the radii of the left-hand curves are much larger. This difference has already been noticed in the experiment from **Figure 7**; here, it is especially pronounced. The difference is due to manufacturing inaccuracies and pressure angle calibration, as discussed in section 4. The lacking ability to control the exact time of fixation of the feet also plays a role: since the strong actuation of a leg also deforms the suction cup, it may no longer be able to suck, despite negative pressure is applied. This effect is most prominent in the rear right foot. All other feet usually fix according to plan. However, the delayed fixation of the rear right leg supports a fast execution of the right turn (see **Supplementary Video**). **Figure 12D** shows the mean values and standard deviations of the step size q_1

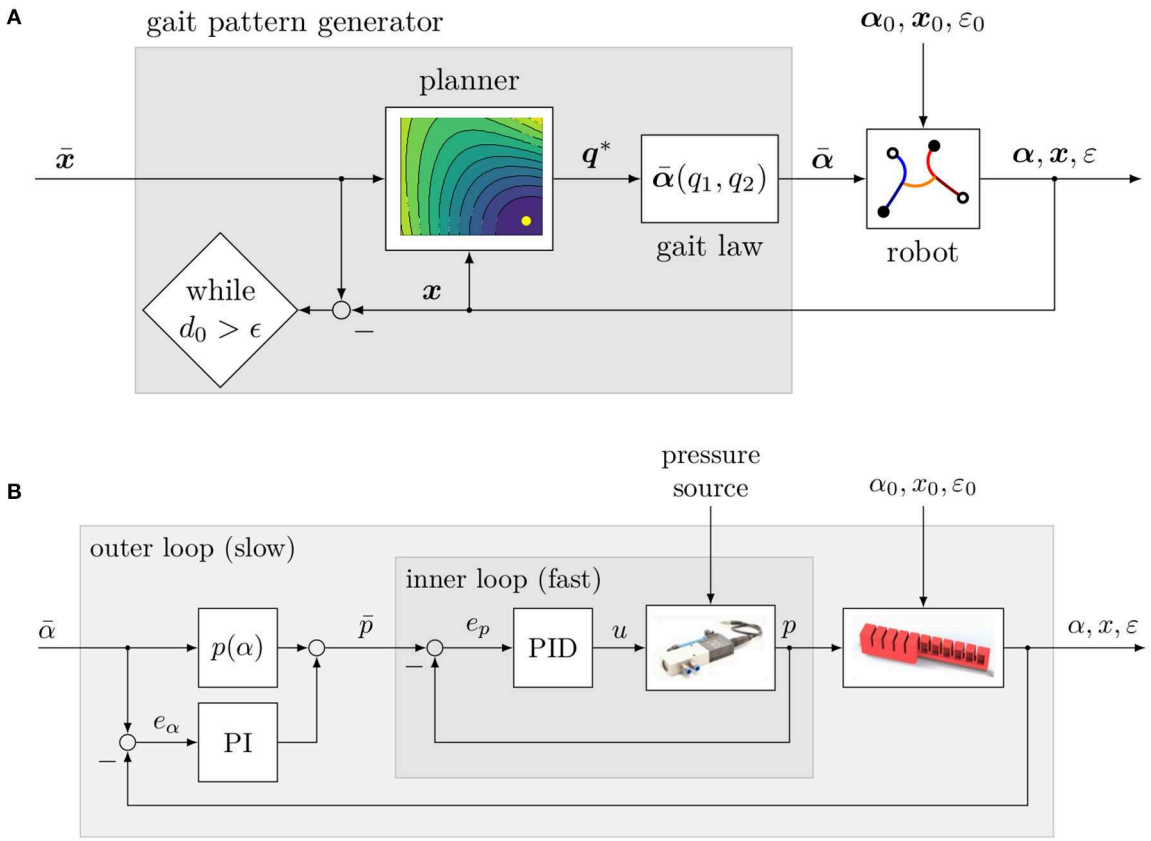


FIGURE 11 | Control architecture of the Gait Pattern Generator. **(A)** For a given target position $\bar{\boldsymbol{x}}$ and the position of the robot $\boldsymbol{x}$, the optimal step length q_1 and steering factor q_2 are calculated and then mapped into reference bending angles $\bar{\boldsymbol{\alpha}}(\cdot)$ by the gait law, which are then fed into the robotic system. In **(B)**, the block diagram of a single actuator is shown. The reference bending angle $\bar{\alpha}$ is mapped by a calibration function $p(\alpha)$ into a reference pressure $\bar{p}$ (feed forward term), which in turn is corrected by a saturated PI controller (feedback term). The reference pressure is then fed into the inner loop, where a PID controller generates the control input u for a proportional valve, which causes the pressure p to be applied to the actuator.

(blue) and the steering factor q_2 (red). Here, the mean value was calculated over the number of steps. The different number of steps required results in a high standard deviation in the region of the four target positions. In order to reach the final target position, 45 steps were required in the fastest run and 51 steps in the worst run. The course of the mean value is similar to the simulation in **Figure 12B** and is not constant. Nevertheless, in all cases, the robot reaches the final goal and always follows a similar path. This proves that also the physical robot can transition between different gait patterns and the reproducibility of the experiments to a certain extent. **Figures 12E,F** show the results of the same experiment now performed with the robot from Seibel and Schiller (2018). The robot is basically the same, but is a little bigger (body length/span: 15/25). For the experiment, the same approximation of the direct kinematics was used (see **Table 3**), and still the robot shows the desired behavior. This shows that $\Delta \boldsymbol{x}$ only needs to reflect the qualitative trend. The exact values are not particularly important because as **Figures 12B,D,F** show, the step length is most of the time at the maximum and therefore the goal cannot be reached within one cycle anyway.

6. CONCLUSION

The aim of this work was position control of the gecko-inspired soft robot from Schiller et al. (2019) in Cartesian space. The solution to this complex task is based on two major simplifications: (i) the formulation of a gait law to reduce the state space of the robot from nine to two dimensions and (ii) the approximation of the direct kinematics to allow a fast evaluation. The gait law restricts the choice of possible references extremely; e.g., only specific trotting gaits are allowed. In this work, it was successfully examined whether a position control system can function with this limitations. However, it has not been investigated whether a larger permitted choice of references leads to better results. In fact, it is possible that the introduction of additional generalized coordinates or a different gait law may lead to a better performance of the robot. Furthermore, neither frictional effects nor any dynamics were considered. Also, by approximating the direct kinematics in the polynomial $\Delta \boldsymbol{x}$, an assumption is made which is fulfilled only in very few cases (compare section 5.2). Instead of using the approximation, the simulation model could also be employed to find the best

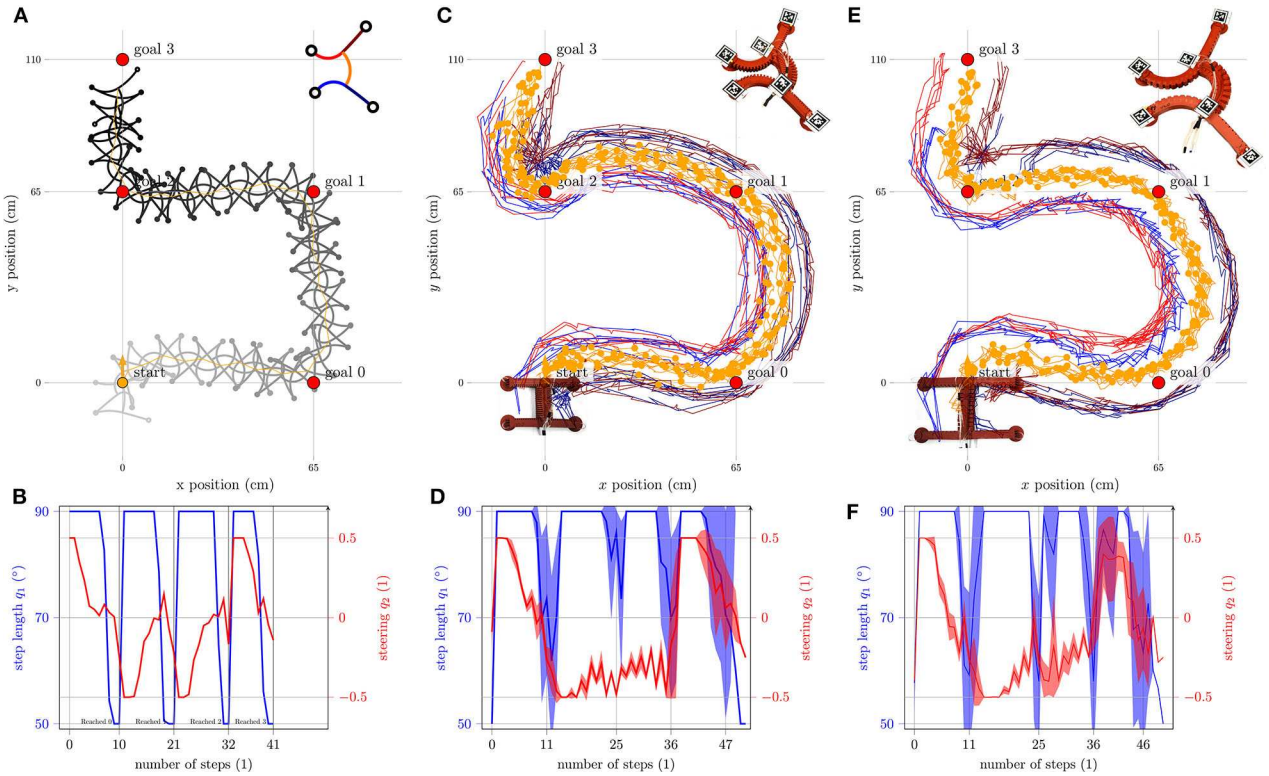


FIGURE 12 | Simulation and experiment with the Gait Pattern Generator in action. In **(A)**, the simulation of gait for a list of four target positions is shown. In **(B)**, the course of q_1 and q_2 is plotted over the number of steps. In **(C,D)**, the corresponding plots are shown for the experiment with the small prototype. **(E,F)** Show the experiment with the large prototype. For the experiments, the color code is as follows: front left foot (red), front right foot (dark red), tip of the torso (orange), torso's end (dark orange), rear left foot (blue), and rear right foot (dark blue).

possible reference for the current situation. But the simulation of one step takes an average of 0.1 s on an *AM335x 1GHz ARM® Cortex-A8* processor, which is used for control. With an average of 10 evaluations of the direct kinematics required to find the reference leading to the minimum distance, this adds up to 1 s. In contrast to a polynomial approximation where the Jacobi matrix can be easily formed to find the minimum efficiently, no analytical Jacobi matrix has been formulated for the simulation model so far. This means that when the simulation model is used, calculation would require most of the time of the cycle. However, the experiments show that the robot always reaches the target, even if the assumptions made in the derivation of the Gait Pattern Generator are not fulfilled and the approximation of the direct kinematics was done for a robot of different dimensions.

The path planning algorithm implemented is very basic, as it minimizes the Euclidean norm of the target vector, i.e., it dictates the direct path from the current position to the target. The gait law provides an intuitive way (forward and rotational speed) to control a quite complex robot and the approximation of the direct kinematics provides the resulting quantitative motion. This opens an interface to a wide variety of more dedicated path planning algorithms, as the robot can now be treated as a unicycle. For example, the path could be planned using Cartesian polynomials (Siciliano et al., 2010) and thus the robot orientation could also be controlled. Although the softness of the robot is very complex to model, it also allows the formulation of very drastic references, even if these cannot be fulfilled at all, as hindered by the closed kinematic chain. How these contradictory demands are solved is then "computed" by the body itself. Conventional parallel kinematic robots, such as the Stewart-Gough platform, would be damaged in this case. The gecko-inspired soft robot is therefore a good example of Embodied Intelligence (Cangelosi et al., 2015) or Morphological Computation (Pfeifer and Gómez, 2009) since it does the right thing "intuitively." This is in agreement with the principle of controlling soft robots mainly in a feed forward way in order to maintain and make use of their softness (Santina et al., 2017). The cascaded controller structure, as discussed in the introduction, can therefore also be applied to position control of mobile robots. The method of deriving a basic locomotion strategy like the presented gait law by very simple (feet rotate only in swing phase), but mathematically (with the constant curvature model) unfulfillable assumptions, can be transferred to any other soft mobile robot. Although this needs to be done individually for each robotic platform, this work can serve as a reference for future and/or existing robots.

DATA AVAILABILITY STATEMENT

The raw data generated for this study are available on request.

AUTHOR CONTRIBUTIONS

LS derived the Gait Pattern Generator, performed the experiments, and discussed the results. LS and AS wrote and revised the manuscript. AS and JS supervised the project. All authors contributed to the article and approved the submitted version.

FUNDING

The publication of this work was supported by the German Research Foundation (DFG) and Hamburg University of Technology (TUHH) in the funding programme "Open Access Publishing."

ACKNOWLEDGMENTS

We thank Rohat Yildiz, Duraikannan Maruthavanan, and Jakob Muchynski for the inspiration and preliminary work.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2020.00087/full#supplementary-material

REFERENCES

Bao, G., Fang, H., Chen, L., Wan, Y., Xu, F., Yang, Q., et al. (2018). Soft robotics: academic insights and perspectives through bibliometric analysis. *Soft Robot*. 5, 229–241. doi: 10.1089/soro.2017.0135

Bern, J., Banzet, P., Poranne, R., and Coros, S. (2019). "Trajectory optimization for cable-driven soft robot locomotion," in *Proceedings of Robotics: Science and Systems* (Freiburg im Breisgau). doi: 10.15607/RSS.2019.XV.052

Bristow, D. A., Tharayil, M., and Alleyne, A. G. (2006). A survey of iterative learning control. *IEEE Control Syst. Mag.* 26, 96–114. doi: 10.1109/MCS.2006.1636313

Cangelosi, A., Bongard, J., Fischer, M., and Nolfi, S. (2015). *Embodied Intelligence*. Berlin; Heidelberg: Springer. doi: 10.1007/978-3-662-43505-2_37

Godage, I. S., Nanayakkara, T., and Caldwell, D. G. (2012). "Locomotion with continuum limbs," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vilamoure), 293–298. doi: 10.1109/IROS.2012.6385810

Hofer, M., and D'Andrea, R. (2018). "Design, modeling and control of a soft robotic arm," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid), 1456–1463. doi: 10.1109/IROS.2018.8594221

Hofer, M., Spannagl, L., and D'Andrea, R. (2019). "Iterative learning control for fast and accurate position tracking with an articulated soft robotic arm," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau), 6602–6607. doi: 10.1109/IROS40897.2019.8967636

Horvat, T., Karakasiliotis, K., Melo, K., Fleury, L., Thandiackal, R., and Ijspeert, A. J. (2015). "Inverse kinematics and reflex based controller for body-limb coordination of a salamander-like robot walking on uneven terrain," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg), 195–201. doi: 10.1109/IROS.2015.7353374

Horvat, T., Melo, K., and Ijspeert, A. J. (2017). Spine controller for a sprawling posture robot. *IEEE Robot. Autom. Lett.* 2, 1195–1202. doi: 10.1109/LRA.2017.2664898

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw*. 21, 642–653. doi: 10.1016/j.neunet.2008.03.014

Katzschmann, R. K., DelPreto, J., MacCurdy, R., and Rus, D. (2018). Exploration of underwater life with an acoustically controlled soft robotic fish. *Sci. Robot*. 3:eaar3449. doi: 10.1126/scirobotics.aar3449

Lee, J., Jin, M., and Ahn, K. K. (2013). Precise tracking control of shape memory alloy actuator systems using hyperbolic tangential sliding mode control with time delay estimation. *Mechatronics* 23, 310–317. doi: 10.1016/j.mechatronics.2013.01.005

Marchese, A. D., Komorowski, K., Onal, C. D., and Rus, D. (2014). "Design and control of a soft and continuously deformable 2D robotic manipulation system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 2189–2196. doi: 10.1109/ICRA.2014.6907161

Pfeifer, R., and Gómez, G. (2009). "Morphological computation-connecting brain, body, and environment," in *Creating Brain-Like Intelligence*, eds B. Sendhoff, E. Körner, O. Sporns, H. Ritter and K. Doya (Berlin, Heidelberg: Springer), 66–83. doi: 10.1007/978-3-642-00616-6_5

Pratt, J., Chew, C.-M., Torres, A., Dilworth, P., and Pratt, G. (2001). Virtual model control: an intuitive approach for bipedal locomotion. *Int. J. Robot. Res*. 20, 129–143. doi: 10.1177/02783640122067309

Qin, L., Liang, X., Huang, H., Chui, C. K., Yeow, R. C.-H., and Zhu, J. (2019). A versatile soft crawling robot with rapid locomotion. *Soft Robot*. 6, 455–467. doi: 10.1089/soro.2018.0124

Rus, D., and Tolley, M. T. (2015). Design, fabrication and control of soft robots. *Nature* 521, 467–475. doi: 10.1038/nature14543

Santina, C. D., Bianchi, M., Grioli, G., Angelini, F., Catalano, M., Garabini, M., et al. (2017). Controlling soft robots: balancing feedback and feedforward elements. *IEEE Robot. Autom. Mag*. 24, 75–83. doi: 10.1109/MRA.2016.2636360

Santina, C. D., Katzschmann, R. K., Bicchi, A., and Rus, D. (2020). Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment. *Int. J. Robot. Res*. 39, 490–513. doi: 10.1177/0278364919897292

Schiller, L., Seibel, A., and Schlattmann, J. (2019). Toward a gecko-inspired, climbing soft robot. *Front. Neurorobot*. 13:106. doi: 10.3389/fnbot.2019.00106

Schiller, L., Seibel, A., and Schlattmann, J. (2020). A lightweight simulation model for soft robot's locomotion and its application to trajectory optimization. *IEEE Robot. Autom. Lett*. 5, 1199–1206. doi: 10.1109/LRA.2020.2966396

Seibel, A., and Schiller, L. (2018). Systematic engineering design helps creating new soft machines. *Robot. Biomimet*. 5. doi: 10.1186/s40638-018-0088-4

Shepherd, R. F., Ilievski, F., Choi, W., Morin, S. A., Stokes, A. A., Mazzeo, A. D., et al. (2011). Multigait soft robot. *Proc. Natl. Acad. Sci. U.S.A*. 108, 20400–20403. doi: 10.1073/pnas.1116564108

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*. London: Springer. doi: 10.1007/978-1-84628-642-1

Tolley, M. T., Shepherd, R. F., Mosadegh, B., Galloway, K. C., Wehner, M., Karpelson, M., et al. (2014). A resilient, untethered soft robot. *Soft Robot*. 1, 213–223. doi: 10.1089/soro.2014.0008

Wang, J., and Olson, E. (2016). "Apriltag 2: Efficient and robust fiducial detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 4193–4198. doi: 10.1109/IROS.2016.7759617

Zhang, W., and Polygerinos, P. (2018). "Distributed planning of multi-segment soft robotic arms," in *2018 Annual American Control Conference (ACC)* (Milwaukee, WI), 2096–2101. doi: 10.23919/ACC.2018.8430682

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.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 21 July 2020
doi: 10.3389/frobt.2020.00095

First-Order Dynamic Modeling and Control of Soft Robots

Thomas George Thuruthel[1]*, *Federico Renda*[2] *and Fumiya Iida*[1]

[1] *Bio-Inspired Robotics Lab, Department of Engineering, University of Cambridge, Cambridge, United Kingdom,* [2] *Khalifa University Center for Autonomous Robotic Systems, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates*

Modeling of soft robots is typically performed at the static level or at a second-order fully dynamic level. Controllers developed upon these models have several advantages and disadvantages. Static controllers, based on the kinematic relations tend to be the easiest to develop, but by sacrificing accuracy, efficiency and the natural dynamics. Controllers developed using second-order dynamic models tend to be computationally expensive, but allow optimal control. Here we propose that the dynamic model of a soft robot can be reduced to first-order dynamical equation owing to their high damping and low inertial properties, as typically observed in nature, with minimal loss in accuracy. This paper investigates the validity of this assumption and the advantages it provides to the modeling and control of soft robots. Our results demonstrate that this model approximation is a powerful tool for developing closed-loop task-space dynamic controllers for soft robots by simplifying the planning and sensory feedback process with minimal effects on the controller accuracy.

Keywords: soft robotics, control, machine learning, dynamic modeling, first-order dynamics, model reduction

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Zheyuan Gong, University of Toronto, Canada
Yisheng Guan, Guangdong University of Technology, China

***Correspondence:**
Thomas George Thuruthel
tg444@cam.ac.uk

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *25 March 2020*
Accepted: *12 June 2020*
Published: *21 July 2020*

Citation:
George Thuruthel T, Renda F and Iida F (2020) First-Order Dynamic Modeling and Control of Soft Robots. Front. Robot. AI 7:95. doi: 10.3389/frobt.2020.00095

1. INTRODUCTION

Soft robotic technologies are becoming increasingly prevalent in the design and development of robots (Kim et al., 2013). Subsequently, there has been growing interests in the modeling and control of soft bodied systems, Unlike robots designed with rigid components, soft robotic systems present novel challenges and opportunities in developing their controllers (George Thuruthel et al., 2018).

The most common modeling and control strategy for soft robots are based on steady-state models, which, under the steady-state assumption, can be equated to the kinematic model (George Thuruthel et al., 2018). For cylindrically-shaped soft robots, this leads to the popular constant curvature model (Webster and Jones, 2010). For other shapes, geometrically exact models or Finite Element Method have been proposed (Trivedi et al., 2008; Renda et al., 2012; Duriez, 2013; Gong et al., 2018). Machine learning techniques can also be used to develop such mappings in a model-free manner (Giorelli et al., 2013; George Thuruthel et al., 2017; Jiang et al., 2017). Refer to Sadati et al. (2017) for a detailed comparison into multiple static modeling techniques. Due to their steady-state assumptions, such controllers will, however, be limited in their reachability, efficiency, and speed. Therefore, controllers developed from dynamic models are much more desirable.

A popular method for developing dynamic models for soft robots is based on the cosserat-rod mechanics. Such models have been extensively used for soft robotic manipulators driven by tendon actuation (Rucker and Webster, 2011; Renda et al., 2014, 2018). For fluidic actuation, other models have been adopted (Marchese et al., 2016; Della Santina et al., 2019). Hybrid models based on

lumped mass systems also looks promising for general soft robots (Sadati et al., 2019). However, all these models will be more computationally intensive than their static counterparts. Learning-based models are a promising alternative in such cases (Thuruthel et al., 2017; Gillespie et al., 2018). Nevertheless, deriving control strategies from dynamic models, in general, introduces additional complexities in motion planning. For an alternate approach that introduces a control-oriented modeling of soft robots, the readers are suggested to look into Della Santina and Rus (2019).

Unlike static controllers, developing fully-dynamic controllers would involve a planning stage. Typically, this has to be performed using some optimization techniques irrespective of the modeling strategy. A good example is the use of trajectory optimization for the control of a soft robotic manipulator using a model-based (Marchese et al., 2016) and model-free method (Thuruthel et al., 2017). This process is time consuming and hence debilitating for closed-loop dynamic control. For fully-actuated soft robots, closed-loop dynamic controllers can be developed in the configuration space (Della Santina et al., 2018). For task-space closed loop control, model-based reinforcement learning is the only viable solution till now, however, they tend to be highly task specific and time consuming (Thuruthel et al., 2018).

This article investigates the viability of a first-order dynamic model for soft robots. It must be noted that unlike state-space dimensionality reduction methods (Thieffry et al., 2018), we are reducing the temporal dimensionality of the dynamic model. Such a model reduction is based on the hypothesis that soft robots typically have high damping and low inertial properties. This makes it possible to approximate the second-order dynamic model to a first-order dynamic model by ignoring the inertial terms (Strogatz, 2001). Even in nature, the ubiquitous muscle dynamics can be modeled as a first-order dynamical system (Zajac, 1989). This model-order reduction provides two advantages. First, first-order dynamical systems are computationally cheaper than second-order dynamical systems. Second, it opens the possibility to develop novel closed-loop control strategies using the reduced-order state feedback. Here, we show the direct learning of the operational space dynamics of the first-order dynamic model. Due to simplifying step, controllers can be easily developed using machine learning and a simple path planning algorithm. Moreover, the sensory requirements for closed-loop dynamic control is reduced because of the simplification.

We investigate the viability of this simplifying assumption using extensive simulation studies. First, we present the theoretical reasoning behind the first-order assumption and its corresponding controller. Then we briefly introduce the fully dynamic simulation model that is used to verify the learned forward models and the dynamic controller. Finally, we present details on the learning architecture together with results of the model and the closed-loop task-space controller.

2. THEORY

Given a soft robot that can be kinematically modeled by the configuration-space $q \in \mathbb{R}^n$, the task-space variable can be obtained by the kinematic transformation:

$$x = F(q) \tag{1}$$

Where, $x \in \mathbb{R}^m$ and $m \leq n$. The task-space variable is typically the pose of the end-effector and is to be controlled. The configuration space is the set of independent variables that fully defines the state of the robot. The fully dynamic model of the soft robot can then be represented using the configuration-space variable as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{2}$$

Here, $M(q)$ represents the inertial properties, $C(q,\dot{q})$ combines the coriolis, centrifugal and damping elements, $G(q)$ represents the gravitational and stiffness effects and τ is the generalized force applied internally by the robot.

Soft robots typically have high damping values with low inertial properties. This is because they are commonly fabricated with viscoelastic materials with low material density. After the initial transient motion of a soft robot from rest (when $\dot{q} = 0$), the first order term dominates the second order term [i.e., $C(q,\dot{q})\dot{q} >> M(q)\ddot{q}$]. Hence, the second-order term can be ignored without sacrificing the accuracy of the model (Zajac, 1989). The dominant modeling error will occur during the initial transient motion (Strogatz, 2001). This transforms the second-order dynamical model (Equation 2) into:

$$C(q,\dot{q})\dot{q} + G(q) = \tau \tag{3}$$

After discretizing the equation, the dynamic model can now be represented through the mapping : $(q_i, \tau_i) \rightarrow q_{i+1}$. Where, q_i, q_{i+1} are the current and the next configuration of the soft robot, respectively. Correspondingly, this implies that a closed-loop dynamic controller would require only the zero-order state feedback (q) for control.

2.1. Controller Design

The obtained first-order dynamical Equation (3), the first-order configuration-space term can be replaced using the well known inverse kinematics mapping:

$$\dot{x} = J(q)\dot{q}$$

$$\dot{q} = J(q)^{\dagger}\dot{x} \tag{4}$$

Where, $J(q)$ is Jacobian matrix and $J(q)^{\dagger}$ is any generalized inverse matrix. Note that we have ignored the null-space terms for brevity. Now the first-order dynamic equation can be reformulated as function of the task-space variables:

$$\tau = F(q,\dot{x}) \tag{5}$$

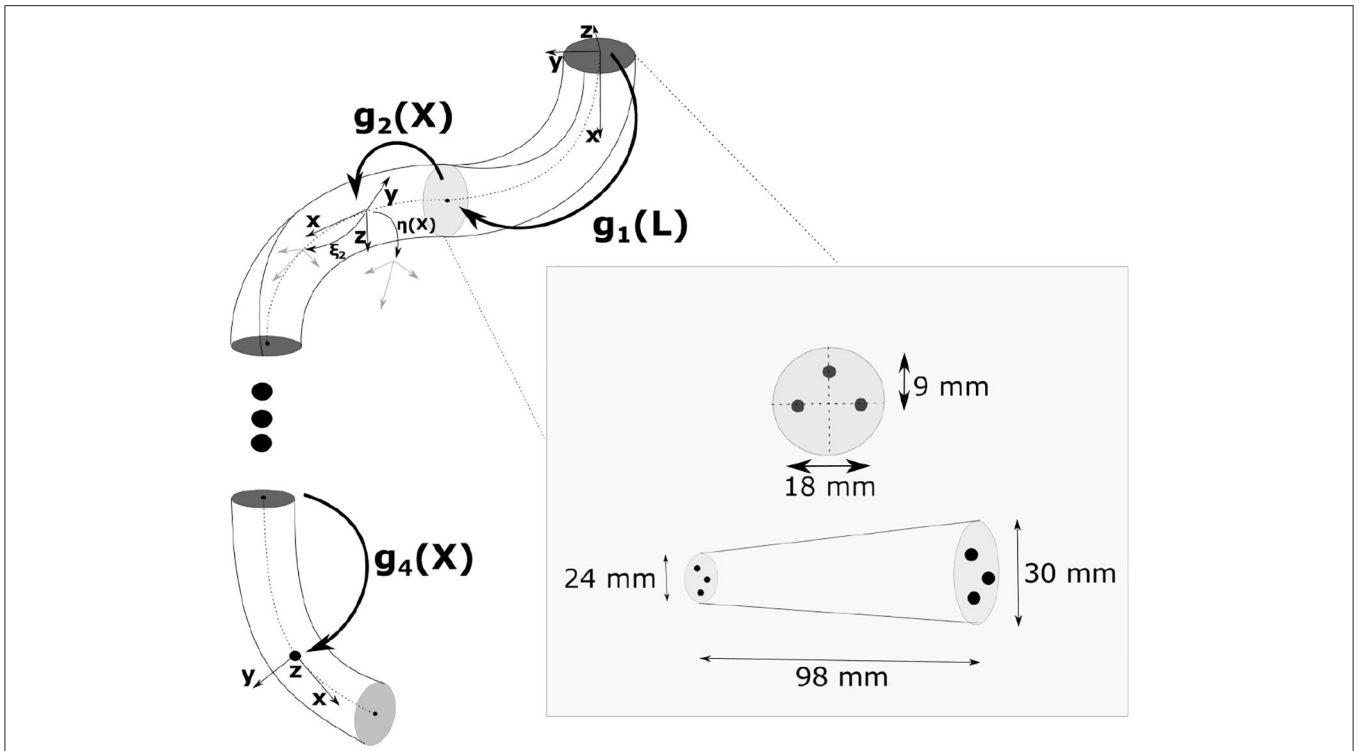


FIGURE 1 | Schematic of the soft manipulator used for our studies. The manipulator is driven by three tendons arranged in the configuration shown above. Unless stated otherwise, the single section manipulator is used for the study.

This functional mapping can now be directly learning by a machine learning architecture. We can now transform the mapping and introduce the target task-space variable x^d as:

$$\dot{x} = x^d - x^c \tag{6}$$

The operational space controller mapping now becomes:

$$\tau = F(q^c, x^d) \tag{7}$$

Here, (q^c, x^c) are the current configuration-space coordinates and the current task-space coordinates, respectively. For the special case when the cardinality of the configuration-space coordinate is the same as the cardinality of the task-space coordinate (mapping from $q \rightarrow x$ is bijective), the operational-space controller can be further simplified to a mapping: $(x^c, x^d) \rightarrow \tau$. A simple feedforward neural network can be used to learn this mapping (**Figure 2**). In this paper, we restrict our studies to this condition for simplicity. This allows us to test the learned controller by providing trajectories in the task-space without the need to solve the inverse kinematics problem. For redundant task-space controllers (non-bijective mappings), additional planning stages might be required to obtain the configuration-space trajectories. In other words, the control trajectory cannot be represented only in the task-space variables. In such a case, an augmented trajectory can be defined with the task-space variable along with a optimization routine to check for kinematic constraints as shown before in Thuruthel et al. (2017). Note that the trajectories are only zero-order task space variables and the feedback required for closed-loop control is also zero-order. This greatly reduces the requirement on the sensors, the effect of noise and the complexity of the trajectory planner.

3. SIMULATION MODEL

The dynamic model is based on the Piece-wise Constant Strain (PCS) approach for soft-rigid multibody system of Renda and Seneviratne (2018) (see **Figure 1**). In the following, all the quantities are expressed in the local (body) coordinate frame if not specified otherwise. The superscript $'$ and $\dot{}$ represent partial differentiation with respect to the space variable and time variables, respectively. The accent $\tilde{}$ represents the usual isomorphism between a vector in $\mathbb{R}^3$ and its corresponding skew-symmetric matrix in $\mathfrak{so}(3)$.

3.1. Kinematics

The relative position and orientation of a soft body i with respect to its predecessor in the chain is defined as a curve $\boldsymbol{g}_i(\cdot) : X \in [0, L_i] \mapsto \boldsymbol{g}_i(X) \in SE(3)$ with

$$\boldsymbol{g}_i(X) = \begin{pmatrix} \mathbf{R}_i & \mathbf{u}_i \\ \mathbf{0}^T & 1 \end{pmatrix}.$$

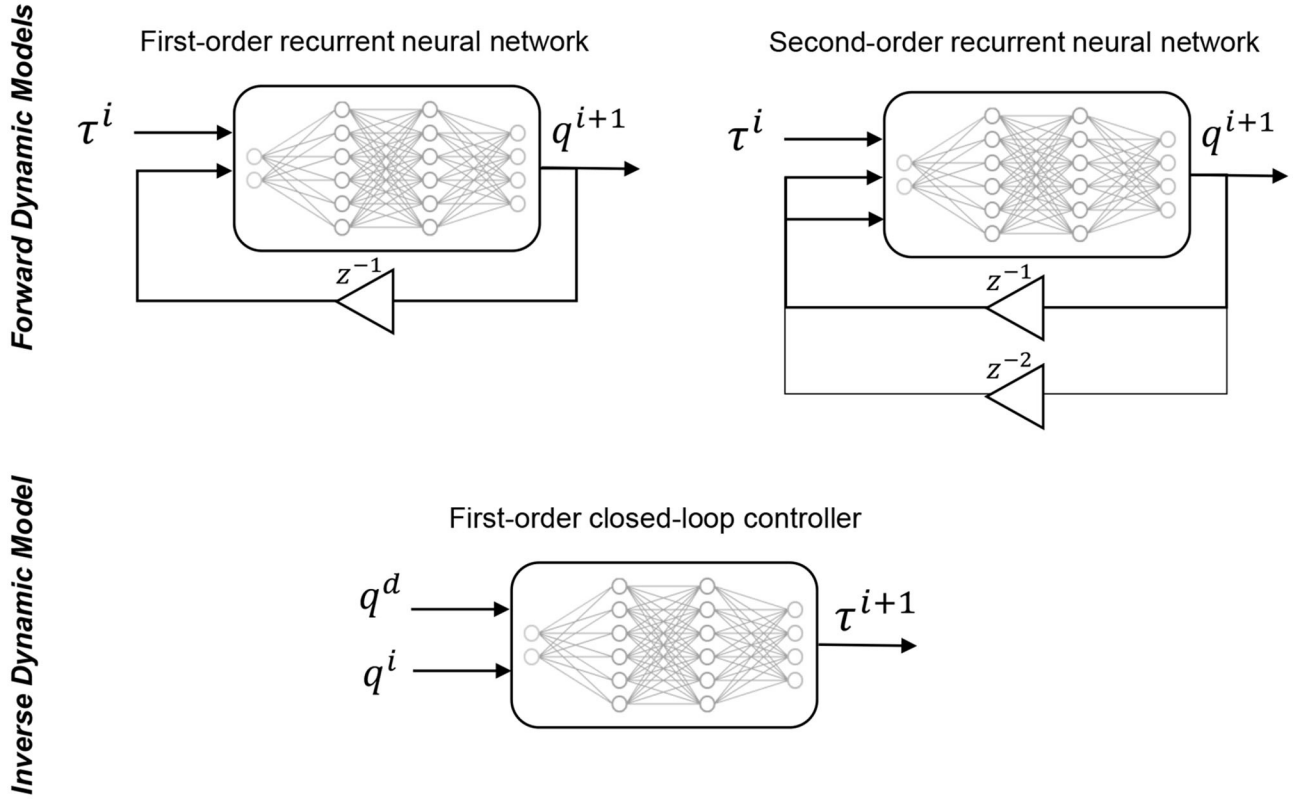


FIGURE 2 | Modeling of the forward dynamics and the inverse dynamics controller. Note that for the non-redundant cases, the configuration-space (q) can be interchanged with the task-space variable (x), as done in this paper.

TABLE 1 | Parameters of the learned forward dynamic model.

Parameter	Value
Type	NARX network
Hidden layer size	40
No. of samples	7000
Training algorithm	Levenberg-Marquardt backpropagation
Training:Testing:Validation ratio	70:15:15
Stopping criterion	Validation set error
Maximum no. of epochs	100

The continuous models of the position, velocity and acceleration of a soft body can be derived from the Cosserat rod theory, which gives (Boyer and Renda, 2016):

$$\boldsymbol{g}_i' = \boldsymbol{g}_i \widehat{\boldsymbol{\xi}}_i \,, \tag{8a}$$

$$\boldsymbol{\eta}_i' = \dot{\boldsymbol{\xi}}_i - \mathrm{ad}_{\boldsymbol{\xi}_i} \boldsymbol{\eta}_i \,, \tag{8b}$$

$$\dot{\boldsymbol{\eta}}_i' = \ddot{\boldsymbol{\xi}}_i - \mathrm{ad}_{\dot{\boldsymbol{\xi}}_i} \boldsymbol{\eta}_i - \mathrm{ad}_{\boldsymbol{\xi}_i} \dot{\boldsymbol{\eta}}_i \,, \tag{8c}$$

where $\boldsymbol{\xi}_i(X) \in \mathbb{R}^6$ defines the strain state, $\boldsymbol{\eta}_i(X) \in \mathbb{R}^6$ is the cross-section velocity twist and $\mathrm{ad}_{(.)} \in \mathbb{R}^{6\times 6}$ is the adjoint operator of the Lie algebra (see Nomenclature). Going further into detail, we have

$$\widehat{\boldsymbol{\xi}}_i(X) = \begin{pmatrix} \tilde{\mathbf{k}}_i & \mathbf{p}_i \\ \mathbf{0}^T & 0 \end{pmatrix} \in \mathfrak{se}(3) \,,\; \boldsymbol{\xi}_i(X) = \left(\mathbf{k}_i^T, \mathbf{p}_i{}^T\right)^T \in \mathbb{R}^6 \,,$$

$$\widehat{\boldsymbol{\eta}}_i(X) = \begin{pmatrix} \tilde{\mathbf{w}}_i & \mathbf{v}_i \\ \mathbf{0}^T & 0 \end{pmatrix} \in \mathfrak{se}(3) \,,\; \boldsymbol{\eta}_i(X) = \left(\mathbf{w}_i^T, \mathbf{v}_i^T\right)^T \in \mathbb{R}^6 \,,$$

with $\mathbf{k}_i(X) \in \mathbb{R}^3$ and $\mathbf{p}_i(X) \in \mathbb{R}^3$ the angular and linear strain; and $\mathbf{w}_i(X) \in \mathbb{R}^3$ and $\mathbf{v}_i(X) \in \mathbb{R}^3$ the angular and linear velocity, respectively.

To model constrained rod, such as the Kirchhoff-Love case with angular strain only, the strain field is specified as:

$$\boldsymbol{\xi}_i = \boldsymbol{B}_i \boldsymbol{q}_i + \boldsymbol{\xi}_i^* \,,$$

where $\boldsymbol{B}_i \in \mathbb{R}^{6\times n_i}$ forms a basis for the allowed motion subspace, $\boldsymbol{q}_i \in \mathbb{R}^{n_i}$ contains the values of the allowed strains and, $\boldsymbol{\xi}_i^* \in \mathbb{R}^6$ is the reference twist modeling the reference shape.

Assuming piece-wise constant strains (Renda et al., 2016), Equations (8) can be analytically integrated using the matrix exponential method, leading to:

$$\boldsymbol{g}_i(X) = e^{X\widehat{\boldsymbol{\xi}}_i}\,, \tag{9a}$$

$$\boldsymbol{\eta}_i(X) = \mathrm{Ad}_{\boldsymbol{g}_i}^{-1}\boldsymbol{\eta}_h + \mathrm{Ad}_{\boldsymbol{g}_i}^{-1}\mathrm{T}_{\boldsymbol{g}_i}\boldsymbol{B}_i\dot{\boldsymbol{q}}_i\,, \tag{9b}$$

$$\dot{\boldsymbol{\eta}}_i(X) = \mathrm{Ad}_{\boldsymbol{g}_i}^{-1}\dot{\boldsymbol{\eta}}_h + \mathrm{Ad}_{\boldsymbol{g}_i}^{-1}\int_0^X \mathrm{Ad}_{\boldsymbol{g}_i(s)}\mathrm{ad}_{\boldsymbol{\eta}_i(s)}ds\boldsymbol{B}_i\dot{\boldsymbol{q}}_i + \mathrm{Ad}_{\boldsymbol{g}_i}^{-1}\mathrm{T}_{\boldsymbol{g}_i}\boldsymbol{B}_i\ddot{\boldsymbol{q}}_i\,, \tag{9c}$$

where $\mathrm{Ad}_{\boldsymbol{g}_i}(X) \in \mathbb{R}^{6\times6}$ is the Adjoint operator of $SE(3)$, and $\mathrm{T}_{\boldsymbol{g}_i}(X) = \int_0^X e^{s\mathrm{ad}_{\boldsymbol{\xi}_i}}ds$ is the tangent operator of the exponential map, of which an analytic expression, derived from (Selig, 2007), is given in the Nomenclature.

Successive applications of the kinematics (Equation 9) for all the bodies of the system, yields to the definition of the geometric Jacobian $\boldsymbol{J}_i(\boldsymbol{q}, X) \in \mathbb{R}^{6\times n}$ and its derivative $\dot{\boldsymbol{J}}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, X) \in \mathbb{R}^{6\times n}$ (n being the total number of DOFs), which relates the generalized coordinate vector $\boldsymbol{q} = \left[\boldsymbol{q}_1^T\ \boldsymbol{q}_1^T\cdots\ \boldsymbol{q}_N^T\right]^T \in \mathbb{R}^n$ (N being the total number of bodies) and the velocity twist $\boldsymbol{\eta}_i(X)$, for each soft body i, as shown below.

$$\boldsymbol{\eta}_i(X) = \sum_{h=0}^{i}\mathrm{Ad}_{\boldsymbol{g}_h\cdots\boldsymbol{g}_i}^{-1}\mathrm{T}_{\boldsymbol{g}_h}\boldsymbol{B}_h\dot{\boldsymbol{q}}_h = \sum_{h=0}^{i}{}^{i}\boldsymbol{S}_h\dot{\boldsymbol{q}}_h = \boldsymbol{J}_i(\boldsymbol{q}, X)\dot{\boldsymbol{q}}\,, \tag{10a}$$

$$\begin{aligned}\dot{\boldsymbol{\eta}}_i(X) &= \sum_{h=0}^{i}{}^{i}\boldsymbol{S}_h\ddot{\boldsymbol{q}}_h + \mathrm{Ad}_{\boldsymbol{g}_h\cdots\boldsymbol{g}_i}^{-1}\int_0^X \mathrm{Ad}_{\boldsymbol{g}_h(s)}\mathrm{ad}_{\boldsymbol{\eta}_h(s)}ds\boldsymbol{B}_h\dot{\boldsymbol{q}}_h \\ &= \sum_{h=0}^{i}{}^{i}\boldsymbol{S}_h\ddot{\boldsymbol{q}}_h + {}^{i}\dot{\boldsymbol{S}}_h\dot{\boldsymbol{q}}_h = \boldsymbol{J}_i(\boldsymbol{q}, X)\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, X)\dot{\boldsymbol{q}}\,,\end{aligned} \tag{10b}$$

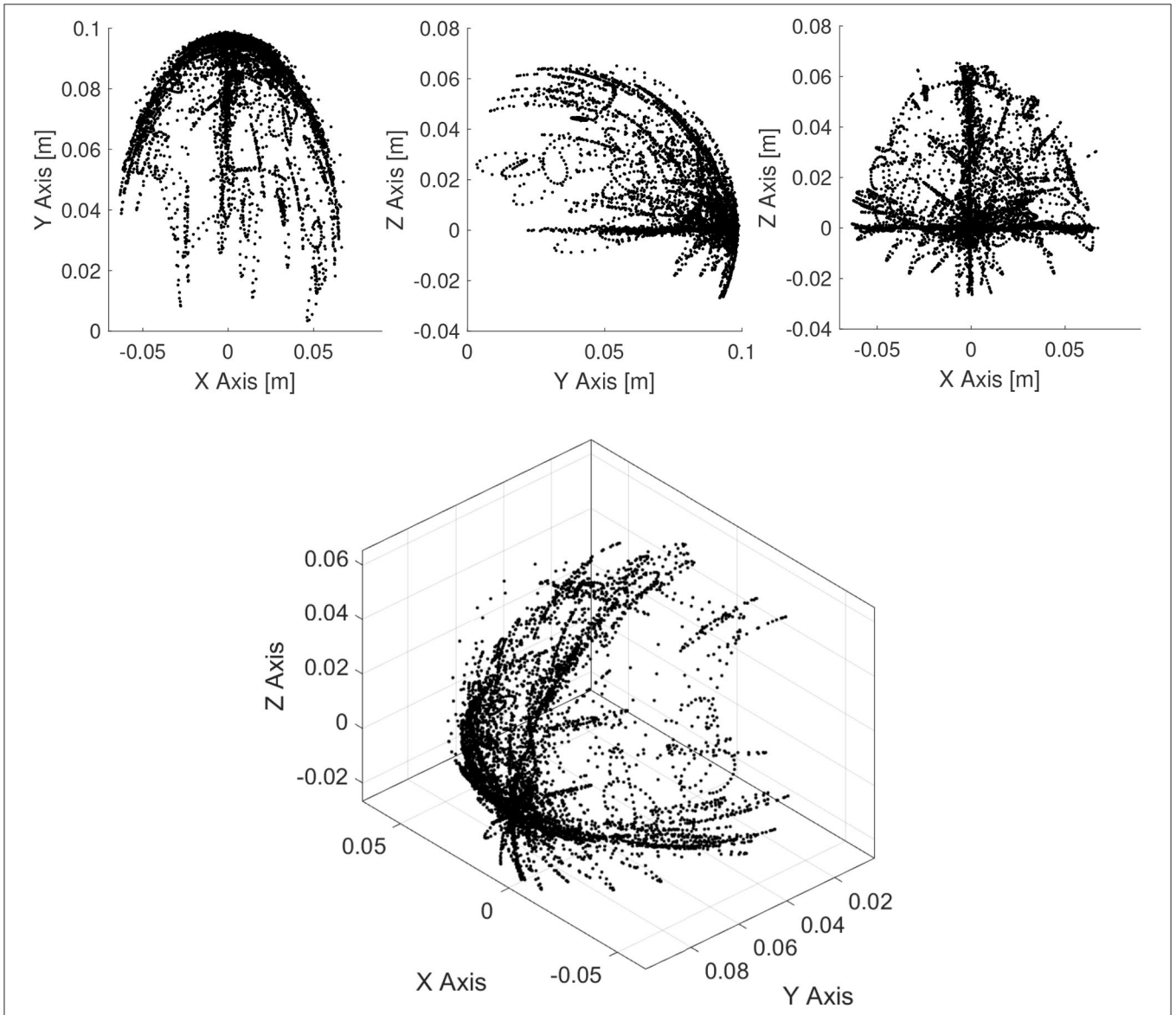


FIGURE 3 | Dynamic workspace of the manipulator obtained by motor babbling. This is obtained by recording end-effector position of the manipulator when actuated by random continuous actuation signals.

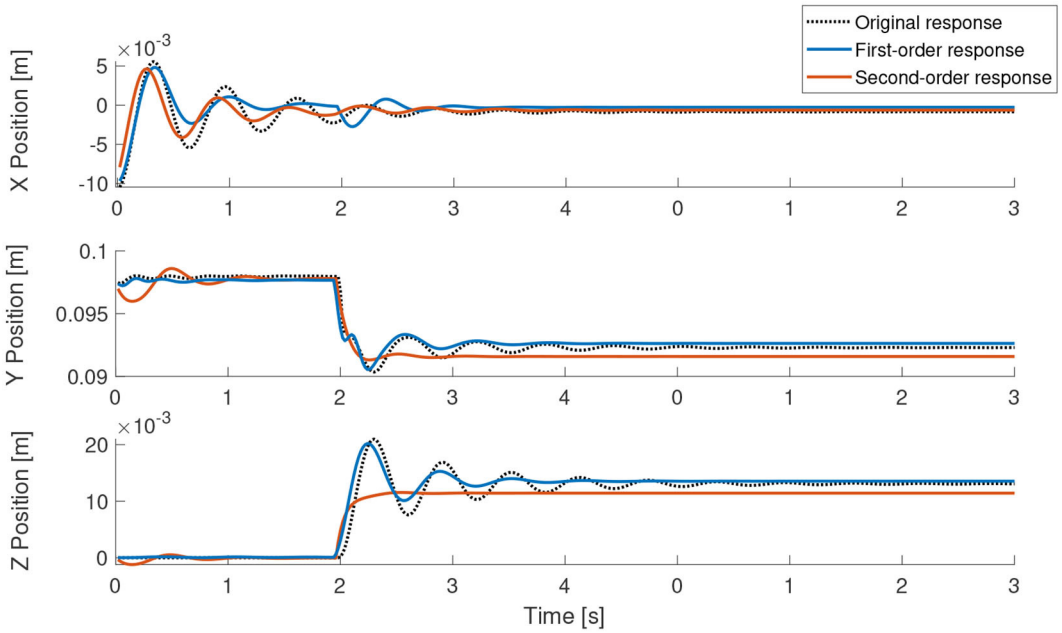


FIGURE 4 | Step response of the learned models in comparison to the actual analytical model. The step signal is given to single actuator at time = 2 s.

where the block elements of the i^{th} Jacobian ${}^{i}\boldsymbol{S}_{(\cdot)} \in \mathbb{R}^{6\times n_{(\cdot)}}$ and its derivative ${}^{i}\dot{\boldsymbol{S}}_{(\cdot)} \in \mathbb{R}^{6\times n_{(\cdot)}}$ have been defined. Note that the last three rows of Equation (10a) provide an analytical expression of the kinematics map required by Equation (4).

3.2. Dynamics

Once a Jacobian is found, the generalized dynamics of the system can be obtained by projecting the free dynamics of each soft body by virtue of the D'Alembert's principle. The free dynamic equation, with its boundary conditions, of a soft body is given by (Renda et al., 2018):

$$\begin{aligned}&\mathcal{M}_i\dot{\boldsymbol{\eta}}_i + \mathrm{ad}^*_{\boldsymbol{\eta}_i}\mathcal{M}_i\boldsymbol{\eta}_i = \left(\mathcal{F}_{i_i} - \mathcal{F}_{a_i}\right)' + \mathrm{ad}^*_{\boldsymbol{\xi}_i}\left(\mathcal{F}_{i_i} - \mathcal{F}_{a_i}\right) + \bar{\mathcal{F}}_{e_i}\,,\\ &\left(\mathcal{F}_{i_i} - \mathcal{F}_{a_i}\right)(0) = -\mathcal{F}_{J_i}\,,\ \left(\mathcal{F}_{i_i} - \mathcal{F}_{a_i}\right)(L_i) = -\mathrm{Ad}^*_{\boldsymbol{g}_{ij}}\mathcal{F}_{J_j}\,;\end{aligned} \tag{11}$$

where $\mathcal{M}_i(X) = \mathrm{diag}(J_{x_i}, J_{y_i}, J_{z_i}, A_i, A_i, A_i)\rho_i \in \mathbb{R}^{6\times 6}$ is the screw inertia matrix of the cross-section ($J_{\cdot_i}(X)$ being the second moment of area about the axis $\cdot$ and $A_i(X)$ the area of the cross-section); $\bar{\mathcal{F}}_{e_i}(X) \in \mathbb{R}^6$ is the distributed external load; $\mathcal{F}_{a_i}(X) \in \mathbb{R}^6$ is the internal wrench due to the distributed actuation (Renda et al., 2017); $\mathcal{F}_{i_i}(X) \in \mathbb{R}^6$ is the internal wrench due to the elasticity of the soft body; $\mathcal{F}_{J_{(\cdot)}} \in \mathbb{R}^6$ is the wrench transmitted across joint $(\cdot)$ and $\mathrm{ad}^*_{(\cdot)}$ (respectively $\mathrm{Ad}^*_{(\cdot)}$) $\in \mathbb{R}^{6\times 6}$ is the co-adjoint (respectively co-Adjoint) map of the Lie algebra (respectively Lie group) defined in Nomenclature. Regarding the internal elastic force, a linear viscoelastic constitutive model is usually chosen:

$$\mathcal{F}_{i_i}(X) = \boldsymbol{\Sigma}_i\left(\boldsymbol{\xi}_i - \boldsymbol{\xi}^*\right) + \boldsymbol{\Upsilon}_i\dot{\boldsymbol{\xi}}_i = \boldsymbol{\Sigma}_i\boldsymbol{B}_i\boldsymbol{q}_i + \boldsymbol{\Upsilon}_i\boldsymbol{B}_i\dot{\boldsymbol{q}}_i\,, \tag{12}$$

where

$$\boldsymbol{\Sigma}_i(X) = \mathrm{diag}(G_iJ_{x_i}, E_iJ_{y_i}, E_iJ_{z_i}, E_iA_i, G_iA_i, G_iA_i) \in \mathbb{R}^{6\times 6} \text{ and}$$

$$\boldsymbol{\Upsilon}_i(X) = \mathrm{diag}(J_{x_i}, 3J_{y_i}, 3J_{z_i}, 3A_i, A_i, A_i)\nu_i \in \mathbb{R}^{6\times 6}$$

are the screw stiffness and viscosity matrix (E_i being the young modulus, G_i the shear modulus and ν_i the shear viscosity).

By Jacobian projection of the free dynamics (Equation 11), we obtain the generalized dynamics in its classical configuration-space form:

$$\boldsymbol{M}\left(\boldsymbol{q}\right)\ddot{\boldsymbol{q}} + \left(\boldsymbol{C}\left(\boldsymbol{q},\dot{\boldsymbol{q}}\right) + \boldsymbol{D}\right)\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{\tau}\left(\boldsymbol{q}\right) + \boldsymbol{F}\left(\boldsymbol{q}\right)\,, \tag{13}$$

where $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is the generalized mass matrix, $\boldsymbol{C} \in \mathbb{R}^{n\times n}$ is the generalized Coriolis matrix, $\boldsymbol{D} \in \mathbb{R}^{n\times n}$ is the block-diagonal generalized damping matrix, $\boldsymbol{K} \in \mathbb{R}^{n\times n}$ is the block-diagonal generalized stiffness matrix, $\boldsymbol{F} \in \mathbb{R}^n$ is the vector of generalized position-dependent external forces and $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of applied actuation forces. Note that the dynamic Equation (13) can be written in the form required by Equation (2), with

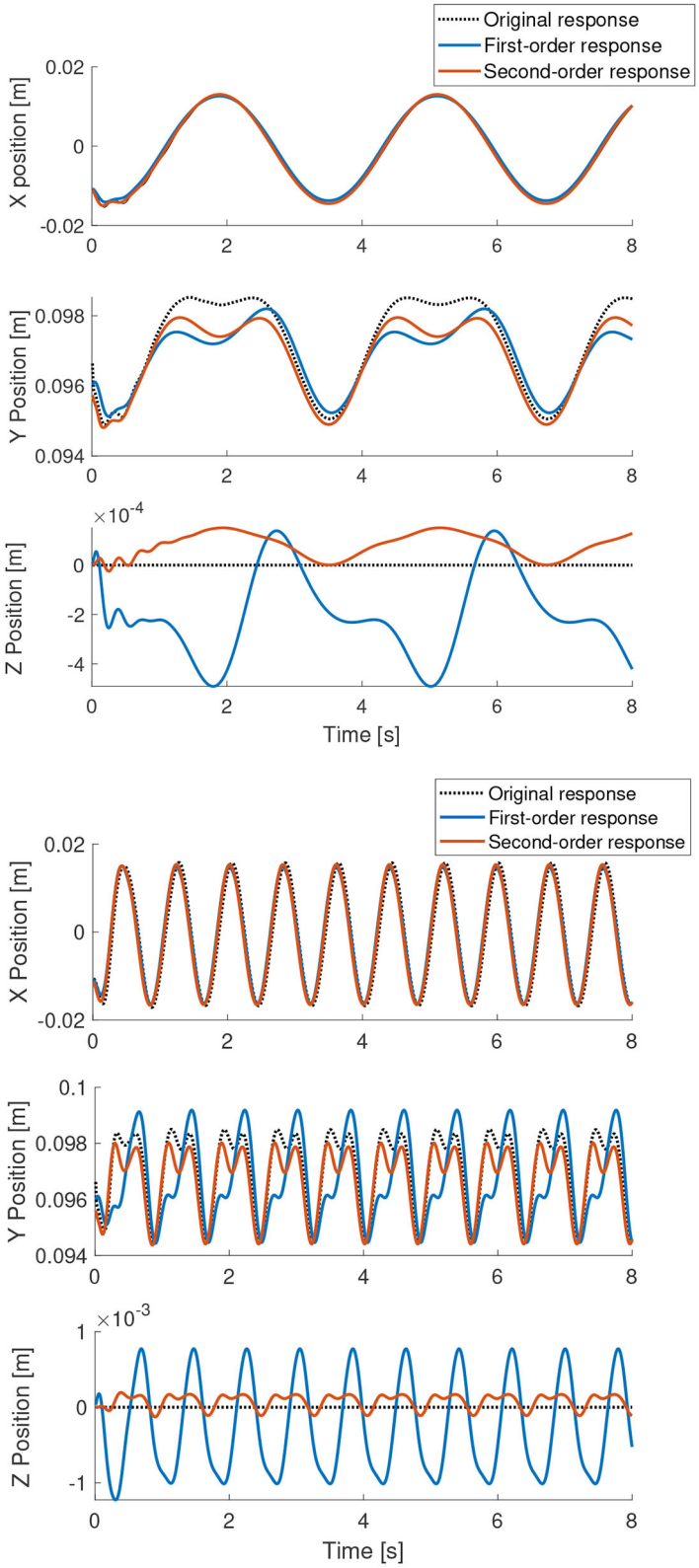


FIGURE 5 | Periodic response of the learned models in comparison to the actual analytical model.

$C(q,\dot{q}) = \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{D}$ and $G(q) = \boldsymbol{K}\boldsymbol{q} - \boldsymbol{F}(\boldsymbol{q})$. Going further into details, the coefficient matrices take the form:

$$\boldsymbol{M}(\boldsymbol{q}) = \sum_{i=1}^{N} \int_0^{L_i} \boldsymbol{J}_i^T \mathcal{M}_i \boldsymbol{J}_i dX \,, \tag{14a}$$

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \sum_{i=1}^{N} \int_0^{L_i} \boldsymbol{J}_i^T \left(\mathrm{ad}^*_{\boldsymbol{J}_i\dot{\boldsymbol{q}}} \mathcal{M}_i \boldsymbol{J}_i + \mathcal{M}_i \dot{\boldsymbol{J}}_i \right) dX \tag{14b}$$

$$\boldsymbol{D} = \mathrm{diag}\left(\boldsymbol{B}_1^T \int_0^{L_1} \boldsymbol{\Upsilon}_1 dX \boldsymbol{B}_1, \boldsymbol{B}_2^T \int_0^{L_2} \boldsymbol{\Upsilon}_2 dX \boldsymbol{B}_2, \cdots, \boldsymbol{B}_N^T \int_0^{L_N} \boldsymbol{\Upsilon}_N dX \boldsymbol{B}_N \right) , \tag{14c}$$

$$\boldsymbol{K} = \mathrm{diag}\left(\boldsymbol{B}_1^T \int_0^{L_1} \boldsymbol{\Sigma}_1 dX \boldsymbol{B}_1, \boldsymbol{B}_2^T \int_0^{L_2} \boldsymbol{\Sigma}_2 dX \boldsymbol{B}_2, \cdots, \boldsymbol{B}_N^T \int_0^{L_N} \boldsymbol{\Sigma}_N dX \boldsymbol{B}_N \right) , \tag{14d}$$

$$\boldsymbol{\tau}(\boldsymbol{q}) = \left[\left(\boldsymbol{B}_1^T \int_0^{L_1} \mathcal{F}_{a_1} dX \right)^T \left(\boldsymbol{B}_2^T \int_0^{L_2} \mathcal{F}_{a_2} dX \right)^T \cdots \left(\boldsymbol{B}_N^T \int_0^{L_N} \mathcal{F}_{a_N} dX \right)^T \right]^T , \tag{14e}$$

$$\boldsymbol{F}(\boldsymbol{q}) = \sum_{i=1}^{N} \int_0^{L_i} \boldsymbol{J}_i^T \bar{\mathcal{F}}_{e_i} \,. \tag{14f}$$

It is worth noting here the different structure of the components of the generalized dynamics Equation (13). Similarly to the minimal Lagrangian models of traditional rigid robots, inertial loads are characterized by full coefficient matrices, as can be see from Equations (14a) and (14b), while damping and stiffness loads are characterized by block-diagonal coefficient matrices, as for Equations (14c) and (14d). This is in contrast with other modeling approaches that use absolute coordinates, such as Finite Elements, for which the opposite holds. Inertial coefficient matrices are block-diagonal while damping and stiffness coefficient matrices are full. Thus, neglecting inertial terms will be well suited for minimal Lagrangian models for soft robotic manipulator, such as the PCS approach.

4. METHODS AND RESULTS

This section investigates two studies. First, we validate the accuracy of the learned first-order model with respect to the learned second-order model. Second, we perform simulated experiments to validate the accuracy of the proposed controller. All the tests are performed on the fully dynamic model described in section 3.

4.1. Dynamic Modeling

The learned models are derived using a kind of recurrent neural network called a nonlinear autoregressive exogenous (NARX) model (Billings, 2013). The NARX network is particularly suited

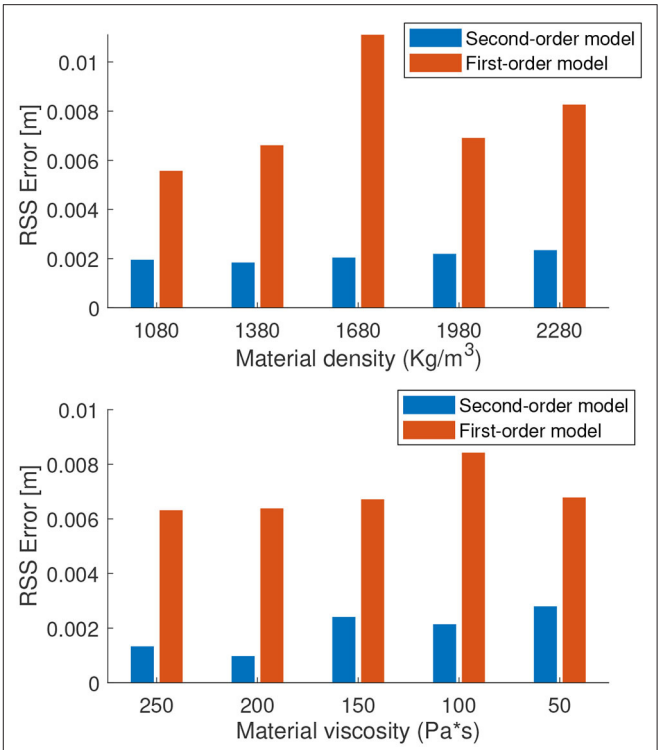


FIGURE 6 | Root sum squared error of the learned model for varying dynamic properties of the manipulator.

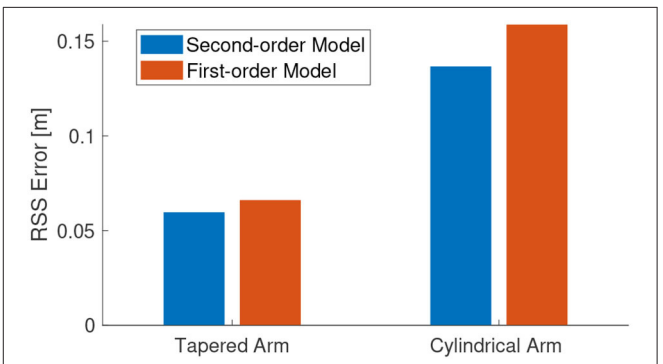


FIGURE 7 | Root sum squared error of the learned model for the 4-section underactuated manipulator.

TABLE 2 | Parameters of the first-order inverse dynamics controller.

Parameter	Value
Type	Feedforward neural network
Hidden layer size	30
No. of samples	7,000
Training algorithm	Levenberg-Marquardt backpropagation
Training:Testing:Validation ratio	70:15:15
Stopping criterion	Validation set error
Maximum no. of epochs	100

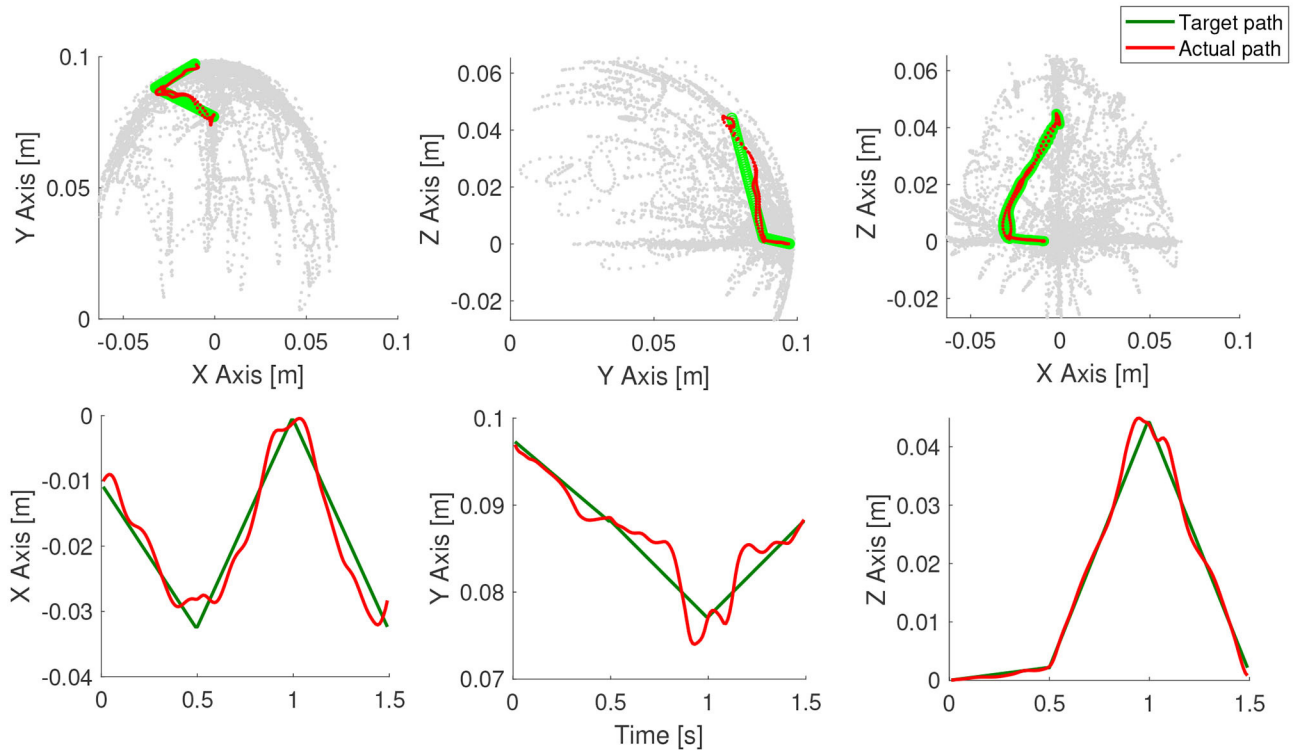


FIGURE 8 | Controller results for a continuous path. The motion of the end-effector is shown here.

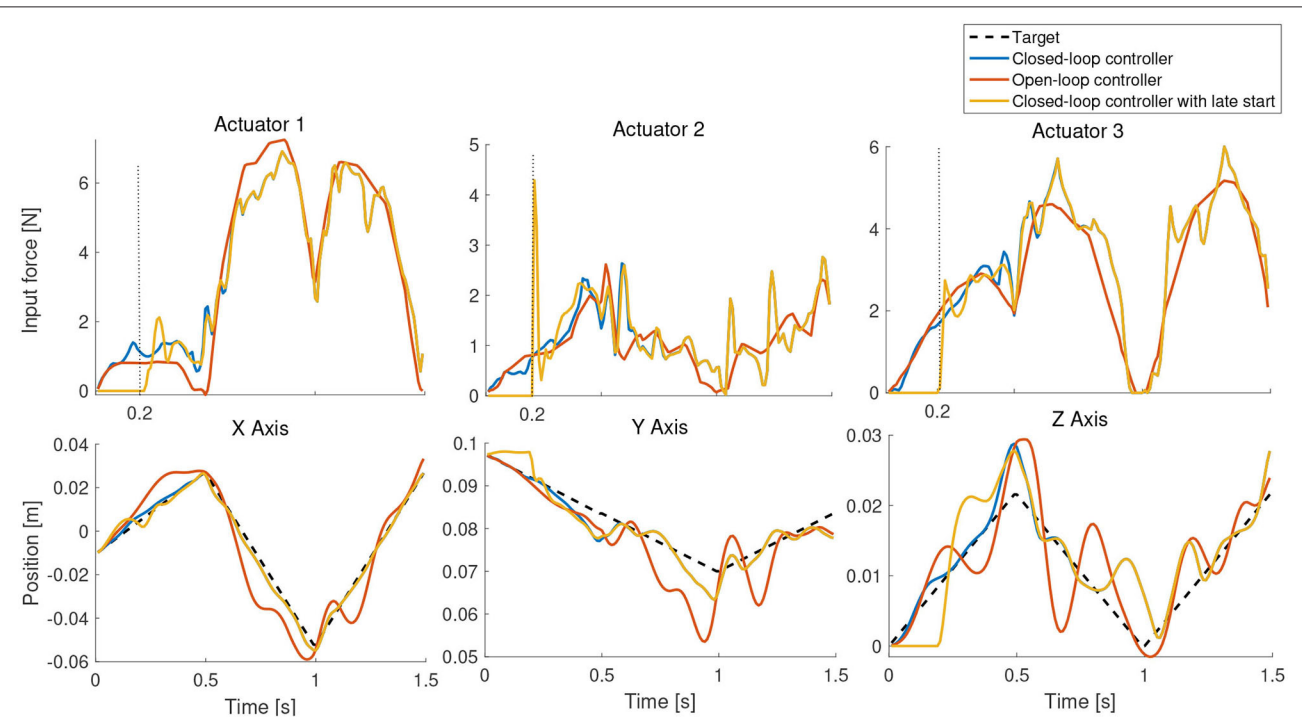


FIGURE 9 | Testing the first-order controller in open-loop, closed-loop, and with a late start.

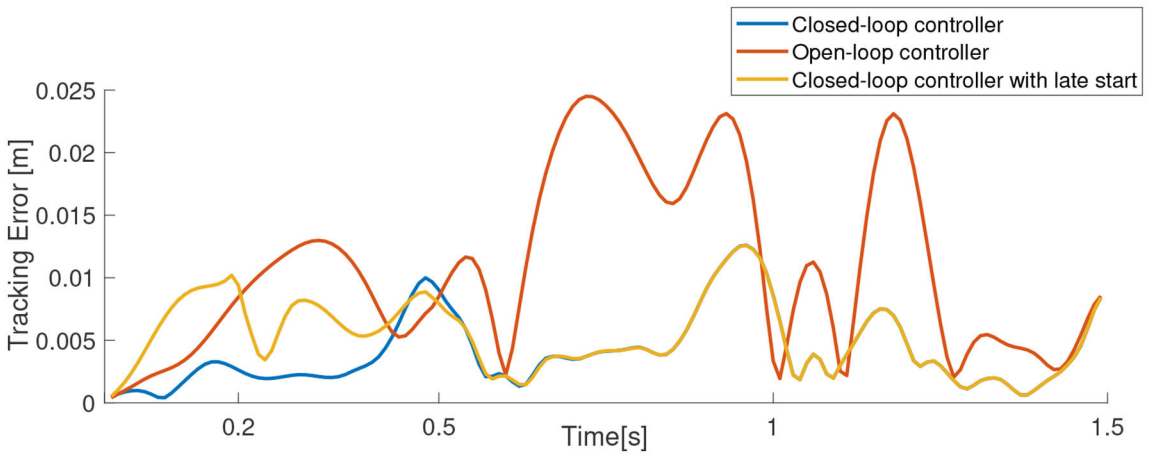


FIGURE 10 | Tracking error for the scenario shown in **Figure 9**.

for our study as it allows us to define the feedback horizon of the recurrent connection explicitly (**Figure 2**). In other words, we can ensure that the neural network receives only zeroth-order feedback for the first-order model and the appended first-order feedback for the second-order model. For the single section soft manipulator, the configuration-space (q) is equivalent to the task-space variable (x), which is defined as the three-dimensional position of the end-effector. We use a recurrent network with one hidden layer with a size of 40 for both the first-order and second-order model, for a fair comparison. The training parameters of the NARX network is given in **Table 1**.

Random actuation of the tendons are performed (motor babbling) for 70 s to obtain the samples for learning the forward model (**Figure 3**). Specifically, random actuation inputs are used to drive the manipulator and the corresponding actuator inputs and end-effector position is recorded over time. The same data samples can also be used for developing the closed-loop controller. Note that the dynamic workspace is concentrated along the direction of the three actuators. This is because tendons in tension have a strong attractor behavior. Hence, it will be easier to move along this direction. When testing our controller, careful measure is taken to ensure that our trajectories pass across this workspace regions.

Figures 4, **5** show the performance of the learned model in comparison to the original cosserat model for a step and periodic response, respectively. Note the higher errors in the first-order model in the beginning of motion for the step response. Since the inertial effects are ignored, it is also visible that oscillations caused by overshoot is not found in the first-order model. However, the steady state error, with respect to the second-order model is relatively small. For the periodic excitation case, the difference between the first-order model and the second-order model is almost non-existent in the relevant coordinates. This is as expected since our approximations are more valid when the manipulator is in a non-stationary state.

In order to analyze the effects of the viscosity and the inertial effects on our modeling assumption, we further perform studies on the accuracy of the first-order model for varying material viscosity and density. As material density increases and the material viscosity decreases, the inertial effects become more and more dominant. Hence, one would expect the accuracy of the first-order model to decrease and the second-order model to remain constant. However, this is not necessarily the case as the training of second-order recurrent neural networks is more prone to instabilities (Pascanu et al., 2013).

Figure 6 shows how the root sum squared (RSS) error of the first-order model is affected when the material properties of the soft arm is changed in a way that weakens our main assumption. The material density is increased up to a factor of 2 and the material viscosity is reduced by a factor of 6. The motor babbling inputs and the neural network parameters are kept the same for the tests. It is clear from the prediction errors that the second-order model always performs better than the first-order model. However, the change in accuracy of both the models are not affected significantly by the change in material properties. Note that the initial parameters of the simulated cosserat model soft arm were obtained from real experiments on an Octopus-inspired soft manipulator, which was manufactured with silicone and driven by tendons (Renda et al., 2014).

Increasing the length of the manipulator is another way to increase the inertial properties and weaken the first-order assumption. For this, we test the same methodology on a 4 section manipulator, however, actuated only on the first section. Each section has approximately the same length, with the total length of the manipulator adding to 418 mm. We test two designs, one with a tapered morphology and the other with a cylindrical morphology (higher inertial properties). The radius of the tapered morphology linearly reduces from 30 to 10 mm while the cylindrical morphology has a fixed diameter of 30 mm. For this test and the following controller results, the default parameters of the manipulator is used (i.e., with the material density of 1,080 kg/m^3 and viscosity of 300 Pas; Renda et al., 2014). The results of our forward dynamics prediction on both the systems are shown in **Figure 7**. Contrary to our

expectation, increasing the length of the manipulator did not affect the performance of the first-order model when compared to the second-order model. We believe this is because the inertial effects are still compensated by the medium (water) in which the manipulator is surrounded in. Our previous studies have shown that the dynamics of the manipulator becomes chaotic and hence unpredictable without a surrounding medium and when the length increases (Thuruthel et al., 2019). Based on our results, it can be deduced that this is more because of the first-order terms rather than the second-order terms. It could be because of the increased length, gravity, centripetal/centrifugal forces, higher DoFs etc. Note that this is a limitation of learning-based approaches as increased sensitivity to initial conditions decrease the stability of the training process and hence the performance of the model. The same applies to the parameter tuning process for analytical models. It could be concluded that it is a good practice to reduce the inertial effects of a soft-bodied system to improve the predictability of its dynamics, irrespective of the order of the model.

4.2. First-Order Dynamic Controller

The closed-loop task-space controller is derived by learning the operational-space dynamics mapping as described in section 2.1. As the mapping is not recursive, it can be learned using a simple feedforward neural network, as shown in **Figure 2**. For our non-redundant case, we can replace the configuration-space variable, q, with the task-space variable, x. The samples for learning the mapping is obtained through the same motor babbling process as described in the previous section. For training the controller the mapping is defined as: $(x^i, x^{i+1}) \rightarrow \tau$. When testing the controller, the next task-space coordinate, x^{i+1}, is replaced by the desired task-space variable x^d. The parameters of the neural network used for learning the first-order inverse dynamics mapping is given in **Table 2**.

Path planning is usually a complex problem in inverse dynamics based controllers, but as our inverse model is developed with only zeroth order state feedback, the development of the task-space trajectory is greatly simplified. Acceptable paths can easily be generated using the data points obtained from the workspace of the manipulator, which is obtained during the motor babbling phase. The desired paths can be generated by picking reachable points from the workspace and routing a path through them, ensuing that there is sufficient time for the manipulator to reach adjacent points. This can be easily done by fixing a cap on the maximum distance between adjacent task-space variables.

To test our controller, we generate randomized linear paths for the end-effector of the manipulator to follow. This is done by picking two random points from the robot workspace and linearly interpolating a trajectory between them and from the initial position of the end-effector. If the initial position of the end-effector is p^0 and the two random points are p^1 and p^2, the generated path is from $p^0 \rightarrow p^1 \rightarrow p^2 \rightarrow p^1$. Note that the intermediate points are not necessarily reachable by this naive approach. Accurate trajectories can be generated by searching for adjacent points in the workspace or by projecting the trajectory onto the workspace surface.

The results of the trajectory tracking is shown in **Figure 8**. The performance of the controller is excellent considering the fact that it is myopic with no step-ahead planning and the task-space trajectory being non-optimal. Due to the low computational cost in running the inverse-dynamic controller, we are able to run closed-loop task-space controller at a very high control frequency of 100 Hz. This will also allow the controller to compensate for any modeling errors incurred by the approximation.

The *quality* of the task-space trajectory can be analyzed by running the controller in open-loop. This can be done by assuming the manipulator is able to reach each trajectory point perfectly and obtaining the best control action at each time step. The results of such a scenario is shown in **Figure 9**, along with the corresponding control inputs. As expected, the open-loop controller performs worse than the closed-loop controller even though there are no external disturbances in the simulation. The same scenario is repeated with the controller inactive for the first 0.2 s in **Figure 9**. As the desired targets are now farther from the current position of the manipulator, it is not necessary that the controller is able to follow the trajectory accurately. However, as evident from the results, the controller is able to recover and remarkably converge to the same solution as the original closed-loop controller. This also shows how important it is to close-the-loop, even at the cost of reduced model accuracy. The tracking error of the three tests described in this scenario is shown in **Figure 10**.

5. CONCLUSION

This paper presents and verifies a model simplifying assumption for soft robots. The core idea of the assumption is that soft robots, by definition, tend to have low inertial and high viscoelastic properties. This leads to dynamic behaviors which are well approximated by a first-order system, as typically observed in nature. We verify this assumption using a simulated fully dynamic model of an Octopus-like manipulator and a type of recurrent neural network called NARX network. Finally, we develop easy-to-develop closed-loop task-space dynamic controllers based on this assumption. Our results indicate that controllers developed on this assumption can compensate the errors in modeling accuracy with the increased control frequency. Our method makes path planning simpler for non-redundant cases. Moreover, the sensory requirements for closed-loop dynamic control is reduced because of the simplification. This is because the state feedback required for the controller is only the zeroth-order component. Our work also indicates that the additional modeling complexity that soft elements introduce can, to some extent, be reduced by designing low inertial highly visco-elastic soft robot designs.

Although we use machine learning tools to test our modeling assumption and develop our controller, the approach is equally suited for analytical approaches. In fact, ignoring the inertial elements would greatly simply the modeling and parameter estimation process involved in model-based control of soft robots. Not only can we reduce the states of the dynamical system, but we also avoid the problem of estimating and inverting

the full mass matrix (see section 3.2). However, it must be kept in mind that first-order systems present numerical challenges in their implementation. Such problems are not found in a learning based approach and hence desirable in that respect. Typically, the first-order model leads to a stiff differential equation and requires specialized techniques for solving them. Interested readers are suggested to refer to Strogatz (2001) for further information. Future work involve extending the work to redundant systems.

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 author/s.

AUTHOR CONTRIBUTIONS

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

FUNDING

This publication was partially supported by Khalifa University of Science and Technology under Award Nos. FSU-2018-08, RC1-2018-KUCARS and Research Excellence (AARE) 2018-05 and by the SHERO project, a Future and Emerging Technologies (FET) programme of the European Commission (grant agreement ID 828818).

REFERENCES

Billings, S. A. (2013). *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Chichester: John Wiley & Sons.

Boyer, F., and Renda, F. (2016). Poincare's equations for cosserat media: application to shells. *J. Nonlin. Sci.* 27, 1–44. doi: 10.1007/s00332-016-9324-7

Della Santina, C., Katzschmann, R. K., Bicchi, A., and Rus, D. (2019). Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment. *Int. J. Robot. Res.* 39, 490–513. doi: 10.1177/0278364919897292

Della Santina, C., Katzschmann, R. K., Biechi, A., and Rus, D. (2018). "Dynamic control of soft robots interacting with the environment," in *2018 IEEE International Conference on Soft Robotics* (Livorno: RoboSoft), 46–53.

Della Santina, C., and Rus, D. (2019). Control oriented modeling of soft robots: the polynomial curvature case. *IEEE Robot. Autom. Lett.* 5, 290–298. doi: 10.1109/LRA.2019.2955936

Duriez, C. (2013). "Control of elastic soft robots based on real-time finite element method," in *2013 IEEE International Conference on Robotics and Automation* (Karlsruhe), 3982–3987.

George Thuruthel, T., Ansari, Y., Falotico, E., and Laschi, C. (2018). Control strategies for soft robotic manipulators: a survey. *Soft Robot.* 5, 149–163. doi: 10.1089/soro.2017.0007

George Thuruthel, T., Falotico, E., Manti, M., Pratesi, A., Cianchetti, M., and Laschi, C. (2017). Learning closed loop kinematic controllers for continuum manipulators in unstructured environments. *Soft Robot.* 4, 285–296. doi: 10.1089/soro.2016.0051

Gillespie, M. T., Best, C. M., Townsend, E. C., Wingate, D., and Killpack, M. D. (2018). "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *2018 IEEE International Conference on Soft Robotics* (Livorno: RoboSoft), 39–45.

Giorelli, M., Renda, F., Ferri, G., and Laschi, C. (2013). "A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Tokyo), 5033–5039.

Gong, Z., Cheng, J., Hu, K., Wang, T., and Wen, L. (2018). "An inverse kinematics method of a soft robotic arm with three-dimensional locomotion for underwater manipulation," in *2018 IEEE International Conference on Soft Robotics* (Livorno: RoboSoft), 516–521.

Jiang, H., Wang, Z., Liu, X., Chen, X., Jin, Y., You, X., et al. (2017). "A two-level approach for solving the inverse kinematics of an extensible soft arm considering viscoelastic behavior," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Singapore), 6127–6133.

Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends Biotechnol.* 31, 287–294. doi: 10.1016/j.tibtech.2013.03.002

Marchese, A. D., Tedrake, R., and Rus, D. (2016). Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. *Int. J. Robot. Res.* 35, 1000–1019. doi: 10.1109/ICRA.2015.7139538

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning* (Atlanta, GA), 1310–1318.

Renda, F., Boyer, F., Dias, J., and Seneviratne, L. (2018). Discrete cosserat approach for multisection soft manipulator dynamics. *IEEE Trans. Robot.* 34, 1518–1533. doi: 10.1109/TRO.2018.2868815

Renda, F., Cacucciolo, V., Dias, J., and Seneviratne, L. (2016). "Discrete cosserat approach for soft robot dynamics: a new piece-wise constant strain model with torsion and shears," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 5495–5502.

Renda, F., Cianchetti, M., Abidi, H., Dias, J., and Seneviratne, L. (2017). Screw-Based Modeling of Soft Manipulators With Tendon and Fluidic Actuation. *J. Mechanisms Robotics.* 9(4):041012. doi: 10.1115/1.4036579

Renda, F., Cianchetti, M., Giorelli, M., Arienti, A., and Laschi, C. (2012). A 3D steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm. *Bioinspir. Biomimet.* 7:025006. doi: 10.1088/1748-3182/7/2/025006

Renda, F., Giorelli, M., Calisti, M., Cianchetti, M., and Laschi, C. (2014). Dynamic model of a multibending soft robot arm driven by cables. *IEEE Trans. Robot.* 30, 1109–1122.

Renda, F., and Seneviratne, L. (2018). "A geometric and unified approach for modeling soft-rigid multi-body systems with lumped and distributed degrees of freedom," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane, QLD), 1567–1574. doi: 10.1109/ICRA.2018.846118

Rucker, D. C., and Webster, R. J. III. (2011). Statics and dynamics of continuum robots with general tendon routing and external loading. *IEEE Trans. Robot.* 27, 1033–1044.

Sadati, S. H., Naghibi, S. E., Shiva, A., Michael, B., Renson, L., Howard, M., et al. (2019). TMTDyn: A matlab package for modeling and control of hybrid rigid–continuum robots based on discretized lumped systems and reduced-order models. *Int. J. Robot. Res.* doi: 10.1177/0278364919881685. [Epub ahead of print].

Sadati, S. H., Naghibi, S. E., Shiva, A., Walker, I. D., Althoefer, K., and Nanayakkara, T. (2017). "Mechanics of continuum manipulators, a comparative study of five methods with experiments," in *Annual Conference Towards Autonomous Robotic Systems* (Guildford, UK: Springer), 686–702.

Selig, J. (2007). *Geometric Fundamentals of Robotics*. Monographs in Computer Science. New York, NY: Springer.

Strogatz, S. (2001). *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering (Studies in Nonlinearity)*. Boca Raton, FL: CRC press.

Thieffry, M., Kruszewski, A., Duriez, C., and Guerra, T.-M. (2018). Control design for soft robots based on reduced-order model. *IEEE Robot. Autom. Lett.* 4, 25–32. doi: 10.1109/LRA.2018.2876734

Thuruthel, T. G., Falotico, E., Renda, F., Flash, T., and Laschi, C. (2019). Emergence of behavior through morphology: a case study on an octopus inspired manipulator. *Bioinspir. Biomimet.* 14:034001. doi: 10.1088/1748-3190/ab1621

Thuruthel, T. G., Falotico, E., Renda, F., and Laschi, C. (2017). Learning dynamic models for open loop predictive control of soft robotic manipulators. *Bioinspir. Biomimet.* 12:066003. doi: 10.1088/1748-3190/aa839f

Thuruthel, T. G., Falotico, E., Renda, F., and Laschi, C. (2018). Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Trans. Robot.* 35, 124–134. doi: 10.1109/TRO.2018.2878318

Trivedi, D., Lotfi, A., and Rahn, C. D. (2008). Geometrically exact models for soft robotic manipulators. *IEEE Trans. Robot.* 24, 773–780. doi: 10.1109/TRO.2008.924923

Webster, R. J. III, and Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots: a review. *Int. J. Robot. Res.* 29, 1661–1683. doi: 10.1177/0278364910368147

Zajac, F. E. (1989). Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Crit. Rev. Biomed. Eng.* 17, 359–411.

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.

NOMENCLATURE

$[\mathrm{ad}_{\boldsymbol{\xi},\boldsymbol{\eta}}]\ (X) = \begin{pmatrix} \widetilde{\mathrm{k}}, \widetilde{\mathrm{w}} & \mathbf{0}_{3\times3} \\ \widetilde{\mathrm{q}}, \widetilde{\mathrm{v}} & \widetilde{\mathrm{k}}, \widetilde{\mathrm{w}} \end{pmatrix} \in \mathbb{R}^{6\times6}$. adjoint representation of the strain twist vector in $SE(3)$.

$[\mathrm{ad}^*_{\boldsymbol{\xi},\boldsymbol{\eta}}]\ (X) = \begin{pmatrix} \widetilde{\mathrm{k}}, \widetilde{\mathrm{w}} & \widetilde{\mathrm{q}}, \widetilde{\mathrm{v}} \\ \mathbf{0}_{3\times3} & \widetilde{\mathrm{k}}, \widetilde{\mathrm{w}} \end{pmatrix} \in \mathbb{R}^{6\times6}$. coadjoint representation of the strain twist vector in $SE(3)$.

$[\mathrm{Ad}_{\boldsymbol{g}}]\ (X) = \begin{pmatrix} \boldsymbol{R} & \mathbf{0}_{3\times3} \\ \widetilde{\mathrm{u}}\boldsymbol{R} & \boldsymbol{R} \end{pmatrix} \in \mathbb{R}^{6\times6}$. Adjoint representation of $SE(3)$.

$[\mathrm{Ad}^*_{\boldsymbol{g}}]\ (X) = \begin{pmatrix} \boldsymbol{R} & \widetilde{\mathrm{u}}\boldsymbol{R} \\ \mathbf{0}_{3\times3} & \boldsymbol{R} \end{pmatrix} \in \mathbb{R}^{6\times6}$. coAdjoint representation of $SE(3)$.

$[e^{X\widehat{\boldsymbol{\xi}}}] = \boldsymbol{I}_4 + X\widehat{\boldsymbol{\xi}} + \frac{1}{\theta^2}(1-\cos(X\theta))\,\widehat{\boldsymbol{\xi}}^2 + \frac{1}{\theta^3}(X\theta - \sin(X\theta))\,\widehat{\boldsymbol{\xi}}^3$. Exponential map in $SE(3)$ with $\theta^2 = \mathrm{k}^T\mathrm{k}$.

$[\mathrm{T}_{\boldsymbol{g}}]\ (X) = X\boldsymbol{I}_6 + \frac{1}{2\theta^2}(4 - 4\cos(X\theta) - X\theta\sin(X\theta))\,\mathrm{ad}_{\boldsymbol{\xi}} + \frac{1}{2\theta^3}(4X\theta - 5\sin(X\theta) + X\theta\cos(X\theta))\,\mathrm{ad}^2_{\boldsymbol{\xi}} + \frac{1}{2\theta^4}(2 - 2\cos(X\theta) - X\theta\sin(X\theta))\,\mathrm{ad}^3_{\boldsymbol{\xi}} + \frac{1}{2\theta^5}(2X\theta - 3\sin(X\theta) + X\theta\cos(X\theta))\,\mathrm{ad}^4_{\boldsymbol{\xi}}$. Analytic expression of the tangent operator of the exponential map in $SE(3)$.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 11 September 2020
doi: 10.3389/frobt.2020.00117

Control Architecture for Human-Like Motion With Applications to Articulated Soft Robots

Franco Angelini[1,2,3]*, *Cosimo Della Santina*[4,5,6], *Manolo Garabini*[1,3], *Matteo Bianchi*[1,3] and *Antonio Bicchi*[1,2,3]

[1] Centro di Ricerca "Enrico Piaggio", Università di Pisa, Pisa, Italy, [2] Soft Robotics for Human Cooperation and Rehabilitation, Fondazione Istituto Italiano di Tecnologia, Genoa, Italy, [3] Dipartimento di Ingegneria dell'Informazione, Università di Pisa, Pisa, Italy, [4] Robotic Mechatronic Center, German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Cologne, Germany, [5] Department of Informatics, Technical University Munich, Munich, Germany, [6] Cognitive Robotics Department, Delft University of Technology, Delft, Netherlands

Human beings can achieve a high level of motor performance that is still unmatched in robotic systems. These capabilities can be ascribed to two main enabling factors: (i) the physical proprieties of human musculoskeletal system, and (ii) the effectiveness of the control operated by the central nervous system. Regarding point (i), the introduction of compliant elements in the robotic structure can be regarded as an attempt to bridge the gap between the animal body and the robot one. Soft articulated robots aim at replicating the musculoskeletal characteristics of vertebrates. Yet, substantial advancements are still needed under a control point of view, to fully exploit the new possibilities provided by soft robotic bodies. This paper introduces a control framework that ensures natural movements in articulated soft robots, implementing specific functionalities of the human central nervous system, i.e., learning by repetition, after-effect on known and unknown trajectories, anticipatory behavior, its reactive re-planning, and state covariation in precise task execution. The control architecture we propose has a hierarchical structure composed of two levels. The low level deals with dynamic inversion and focuses on trajectory tracking problems. The high level manages the degree of freedom redundancy, and it allows to control the system through a reduced set of variables. The building blocks of this novel control architecture are well-rooted in the control theory, which can furnish an established vocabulary to describe the functional mechanisms underlying the motor control system. The proposed control architecture is validated through simulations and experiments on a bio-mimetic articulated soft robot.

Keywords: motion control algorithm, motor control, natural machine motion, articulated soft robots, human-inspired control, compliant actuation

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Barkan Ugurlu, Özyeğin University, Turkey
Kosta Jovanovic, University of Belgrade, Serbia

***Correspondence:**
Franco Angelini
frncangelini@gmail.com

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *30 April 2020*
Accepted: *28 July 2020*
Published: *11 September 2020*

Citation:
Angelini F, Della Santina C, Garabini M, Bianchi M and Bicchi A (2020) Control Architecture for Human-Like Motion With Applications to Articulated Soft Robots. Front. Robot. AI 7:117. doi: 10.3389/frobt.2020.00117

1. INTRODUCTION

Daily activities of human beings are a clear example of the exceptional versatility of their motor control system. Tasks that are still challenging for robots are indeed easily executed by people. Responsible for such a high level of performance are the musculoskeletal system and the Central Nervous System (CNS). The musculoskeletal system allows to exert forces and to percept the external world through a multitude of receptors. One of the main characteristics of this system is its compliant nature. Indeed, body flexibility provided by muscles and tendons enables features like energy efficiency, power amplification and shock absorption (Roberts and Azizi, 2011).

The same feature are usually hard to be achieved by traditional rigid robots. Inspired by the effectiveness of the biological example, researchers developed robots with compliant elements to mimic the animal body. This novel generation of systems, namely soft robots, can be categorized as invertebrate-inspired or vertebrate-inspired (Della Santina et al., 2020). The latter class includes *articulated soft robots*, which are systems with rigid links and elasticity lumped at the joints (Albu-Schaffer et al., 2008). In this paper, we focus on the latter category, i.e., robots actuated by series elastic actuators (SEA) (Pratt and Williamson, 1995) or variable stiffness actuators (VSA) (Vanderborght et al., 2013). The musculoskeletal system of vertebrates allows to adjust its dynamics, for instance, it allows to vary joint stiffness via co-contraction of antagonistic muscles. Agonistic-antagonist VSAs mimic this mechanism as described in Garabini et al. (2017), thus they try to replicate the working principle of the human musculoskeletal system.

Several works in literature describe how the features of a flexible body can be conferred also to a robot through different solutions (Landkammer et al., 2016; Zhang et al., 2019; Pfeil et al., 2020). Particularly relevant are the solutions that completely replicate the whole structure of the human musculoskeletal system. For examples, Kenshiro (Asano et al., 2016) is a humanoid robot reproducing the human skeleton and muscle arrangement. Marques et al. (2010) presents ECCE, an anthropomimetic humanoid upper torso. Jäntsch et al. (2013) proposes Anthrob, a robot mimicking a human upper limb.

Yet, controlling soft robots still remains a very challenging task. The reason is that articulated soft robots have highly non-linear dynamics, presenting also hysteresis, bandwidth limitation and delays. Therefore, obtaining an accurate and reliable dynamic model is not a trivial task that could directly affect the performance of model-based control techniques. Moreover, articulates soft robots present anatomical degrees of freedom (DoFs) redundancy, because they typically have more than one motor per joint, and they may have kinematic DoFs redundancy, depending on the platform. The majority of existing model-based control approaches has the strong drawback of requiring an accurate model identification process, which is hard to be accomplished and time-consuming. In Buondonno and De Luca (2016) feedback linearization of VSA is faced. In Zhakatayev et al. (2017) an optimization framework to minimize time performance is proposed. In Keppler et al. (2018) the Authors propose a controller to achieve motion tracking while preserving the elastic structure of the system and reducing the link oscillations. On the other hand, model-free algorithms are promising, but usually require long-lasting learning procedures and face generality issues (Angelini et al., 2018; Hofer et al., 2019).

However, the complexity of the articulated soft robot body is analogous to that of their source of inspiration. Indeed, the human body is a complex system that presents an unknown non-linear dynamics and redundancy of degrees of freedom (DoFs). Despite that, the CNS is able to cope with these issues, fully exploiting the potential of the musculoskeletal system. For this reason, in this work, we analyze the effectiveness of a bio-inspired algorithm to control bio-mimetic robots.

To the authors best knowledge, despite the variety of approaches in the motor control field, an architecture based on control theory able to present at the same time various CNS behavior is still lacking for articulated soft robots (Cao et al., 2018; Ansari et al., 2019). The study of the human CNS has been already exploited to enhance robot capability. For instance, in Medina et al. (2019) the Authors propose a method for modeling human motor behavior in physical and non-physical human-robot interactions. Based on previous observations, the developed model is able to predict the force exerted during the interaction. Capolei et al. (2019) presents a cerebellar-inspired controller for humanoid robot moving in unstructured environment. The controller is based on machine learning, artificial neural network, and computational neuroscience. In Kuppuswamy et al. (2012) the Authors propose a motor primitive inspired architecture for redundant and compliant robots. Lee et al. (2018) proposes a model of human balancing with the goal of designing a controller for exoskeleton.

In this work, our goal is to make a step further toward the development of human-inspired controllers for articulated soft robots: taking inspiration from motor control theories, we implemented a hierarchical control architecture exhibiting well-known characteristics of human motor control system (i.e., learning by repetition, anticipatory behavior, synergistic behavior). Such a control framework is a proper combination of feedback control, feedforward, Iterative Learning Control, and Model Predictive Control. The goal is to design a bio-mimetic control architecture for bio-inspired robots, focusing on trajectory planning and tracking tasks.

A major contribution of this work is to show how well-established paradigms belonging to the control theory can be used to approach the motor control problem. Finally, the authors want to clearly state that is beyond the scope of this work to infer possible neurophysiological implications based on the presented control framework.

Our belief is that a control system able to work like the CNS, such the one proposed here, can successfully manage a soft robotic system. We test here this hypothesis, among with the human-like behaviors, both in simulation and in experiments, using as testbed robots actuated by VSAs.

2. THE BIOLOGICAL INSPIRATION

The unparalleled performance of the animal CNS are an ambitious goal for the robotic community, especially because the issues faced by the CNS are very similar to the ones occurring in robots, i.e., unknown non-linear dynamics and redundancy of degrees of freedom. These are (Latash, 2012):

- *Unknown non-linear dynamics*. The human body is a complex system, with strong non-linearities at every level. Moreover, environmental force fields can not be known *a priori*.

- Degree of freedom (DoF) redundancy. The human body presents three types of redundancy. *Anatomical*—human body is characterized by a complex highly redundant

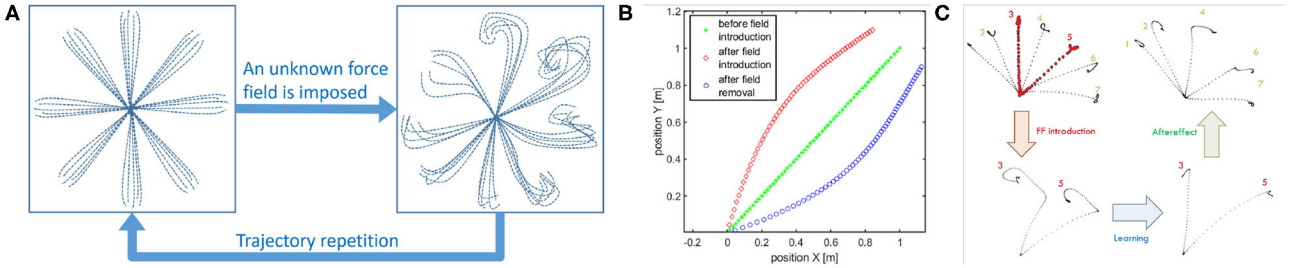


FIGURE 1 | Representation of some human behaviors considered in this work. Learning by repetition **(A)**: a subject is able to reach a series of point in space with its end effector, when a force field is imposed the trajectories result deformed, repeating the reaching trials many times the subject results able to restore the initial behavior. Aftereffect in known trajectories: **(B)** Hand trajectories of a typical point to point movement. The typical movement is a strict line. If a force field is introduced the trajectory is firstly deformed. After some repetitions the strict movement is recovered. If the force field is then removed the hand trajectory is deformed in a way specular to the first deformation. This is called aftereffect. Aftereffect in unknown trajectories: **(C)** Hand trajectories of typical point to point movements. When the force field is introduced the subject make experience through learning by repetition of just trajectories 3 and 5. When the force field is removed aftereffect is present on trajectories not experienced closer to trajectories 3 and 5: trajectory 4 presents maximum aftereffect, trajectories 1 and 7 presents negligible aftereffect (image obtained from an elaboration of images in Gandolfo et al., 1996).

structure. The number of joints is greater than the number of DoFs necessary to accomplish a generic task, and the number of muscles is greater than the number of joints. *Kinematic*—infinite joints trajectories can achieve the same task, or simply perform the same end effector point to point movement. *Neurophysiological*—each muscle consists of hundreds of motor units, and they are activated by moto-neurons that can spike with different frequency (hundreds of variables).

For this reason, we use the motor control theory as a source of inspiration for our controller.

2.1. Hierarchical Nature of the Central Nervous System

There are several evidences that the Central Nervous System can cope with the incredible complexity of the musculoskeletal apparatus by relying on a hierarchical organization of subsequent simplifications of the control problem (Swanson, 2012; Hordacre and McCambridge, 2018). For example, the Bernstein classification (Bemstein, 1967) categorizes the construction of movement in six levels, from symbolic reasoning to muscle tone activation. Level A is called *rubro-spinal level* or *paleokinetic level*, and it provides reflex function and manages muscle tone. Level B, i.e., *thalamo-pallidal level*, is the level of synergies and patterns and produces coordinate movement patterns. Finally, level C1, is the *striatal* or *extrapyramidal level*. This is one of the two levels of the *spatial field* level, and it specifies a way to reach performance defined by higher levels. The other three levels, C2, D, and E, describe higher level of abstractions, as meaningful actions and information transmission. Therefore, they will not be treated in by the proposed control architecture.

2.2. Some Salient Characteristics of the Human Motor Control

In this section we list a few of salient characteristics of the neural control architecture that we consider of paramount importance for the human motion performance, and that we aim at replicating on the considered bio-mimetic robots. In the remainder of the article we will often refer to them as (i)–(v). These peculiar characteristics of the CNS are:

(i) Learning by repetition (Shadmehr and Mussa-Ivaldi, 1994): CNS inverts an unknown dynamic over a trajectory, repeating it several times. **Figure 1A** represents a classical experiment. It is possible to notice that the subject is asked to reach some points in the workspace. Then a force field is introduced. Initially, trajectories are strongly deformed. After repetitions of the same movements, performances obtained before the introduction of the force field are achieved again. The same behavior can be found in the development, where the CNS needs to adapt to its own dynamics.

(ii) Anticipatory behavior (Hoffmann, 2003): ability of CNS to usually anticipate the necessary control action relying on sensory-motor memory. The acquired previous experiences cause a shift in the control action from closed loop to open loop. Anticipatory behavior is fundamental in many human activities, such as manipulation (Fu et al., 2010), coordinated (Flanagan and Wing, 1993), and fast movements (Haith et al., 1988).

(iii) Aftereffect over a learned trajectory (Lackner and Dizio, 1998) and aftereffect over unknown trajectories (Gandolfo et al., 1996). After recovering the performance loss due to the introduction of the external force field, by removing the force field, subjects exhibit deformations of the trajectory specular to the initial deformation due to the force field introduction. This behavior is called mirror-image aftereffect **Figure 1B**. This effect arises also in novel trajectories as depicted in **Figure 1C**.

(iv) Synergistic behavior (Latash, 2010): synergy can be defined as "*[. . .] a hypothetical neural mechanism that ensures task-specific co-variation of elemental variables providing for desired stability properties of an important output*

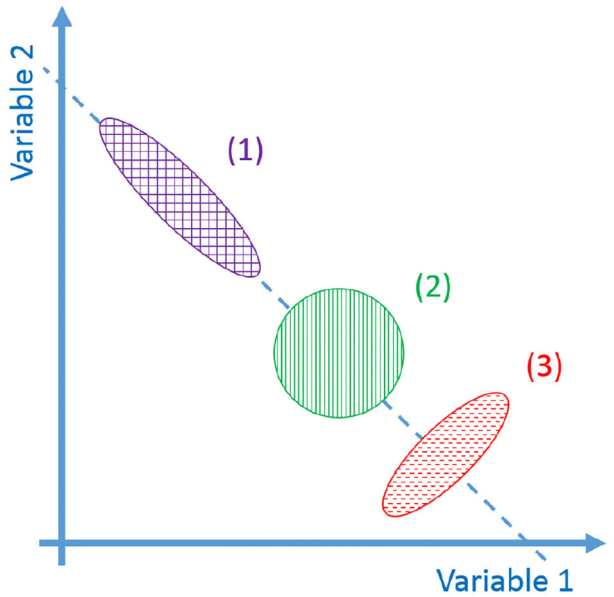


FIGURE 2 | Representations of the synergistic behavior. In the figure there are different possible distributions of task configuration in task space. The dashed line is locus of configurations that meets the task. V_{good} is the variance of the distribution along the dashed line, V_{bad} is the variance in the orthogonal directions. The fact that $V_{good} > V_{bad}$ indicates that a task synergy exists (1). If $V_{good} \backsimeq V_{bad}$ no synergy exists (2). If $V_{good} << V_{bad}$ a destabilizing synergy exists (3).

(performance) variable." Given an "important output variable" we can define two variables V_{good} and V_{bad}. V_{good} is the variance through the directions where output is constant and the constraints are verified (named *uncontrolled manifold*), while V_{bad} is the variance in the other directions (Scholz and Schöner, 1999). The system presents a synergistic behavior when $V_{good} > V_{bad}$. **Figure 2** visually explains this point.

(v) Re-plan of anticipatory action: CNS modifies the anticipatory motor actions on-line if the goal changes (e.g., Soechting and Lacquaniti, 1983), or if the sensory outcome is different from the expected one (e.g., Engel et al., 1997). Note that this is fundamentally different from feedback. Indeed, feedback actions are proportional to the instantaneous error, while re-plan of anticipatory action depends on the outcome of the task.

3. PROBLEM STATEMENT

Inspired by the biological example, we design the control architecture with a hierarchic structure similar to the one of CNS. In particular we reproduce the first three levels of the Bernstein classification (Bemstein, 1967) (briefly summarized in section 2.1) with the goal of executing a task reference ν generated by the three higher abstraction levels. Furthermore, the controller has to reproduce the peculiar behaviors of the human CNS described in section 2.2.

We refer to a generic dynamic system, which may represent both articulated soft robots and biological models (**Figures 3A,B**), i.e., $\dot{x}(t) = f(x(t), u(t))$, $y(t) = h(x(t))$, where f is the dynamic function, $x = [q^{\mathrm{T}}, \dot{q}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{2n}$ is the state vector, $q \in \mathbb{R}^n$ are the Lagrangian variables, $y \in \mathbb{R}^l$ is the output variable, and $h(x)$ is the output function. It is worth mentioning here that human muscles and agonistic antagonistic variable stiffness actuators share similar characteristics as depicted in **Figures 3C,D** (Garabini et al., 2017). We propose a bio-mimetic control architecture for bio-inspired robots. The architecture is divided into two layers and summarized in **Figure 4**. The whole controlled system is organized in four building blocks: the two control levels, the dynamic system, and the output function $h(x)$ selecting the portion of the state from which depends the task to be accomplished.

The low level features characteristics similar to level A of the Bernstein classification, i.e., it provides low level feedback and dynamic inversion. Thus, it generates as output the efferent action u depending on afferent proprioceptive inputs, i.e., $q, \dot{q}$, and higher level reference $\rho \in \mathbb{R}^p$, generated by the high level control, relying on q and y. Thus, given a desired output trajectory $\hat{y}:[0, t_f) \rightarrow \mathbb{R}^l$, where t_f is the terminal time, the low level control is an appropriate controller able to track that trajectory. On the other hand, the high level control is inspired by level B and level C1 and provides task management.

The low level controller has to present three behaviors: learning by repetition (i), anticipatory behavior (ii), and aftereffect over known and unknown trajectories (iii). The high level control will present synergistic behavior (iv) and ability of re-plan the anticipatory action (v).

To design the control architecture we assume the desired robot impedance behavior as given. Future extension of this work will also consider a direct learning of the optimal impedance depending on the task.

4. FROM MOTOR CONTROL TO MOTION CONTROL

In this section we describe the proposed control architecture and its components. To obtain learning by repetition (i) we will employ a learning algorithm able to cope with the non-linear dynamics of the studied class of robots. In particular, we rely on the Iterative Learning Control (ILC) framework (Bristow et al., 2006). The employed ILC method merges a low gain feedback with a feedforward action. Through repetitions the feedforward action will prevail over the feedback action leading to the desired anticipatory behavior (ii). It is worth mentioning that ILC is a local method and requires a new learning phase for every novel desired trajectory. Conversely, humans are able to generalize the motion learned through repetitions (Sternad, 2018). To obtain the same feature, we employ Gaussian Process Regression (GPR) (Williams and Rasmussen, 2006) to create a map of learned trajectories. We aim at obtaining also aftereffect, i.e., behavior (iii)—to test the level of bio-mimecity of the

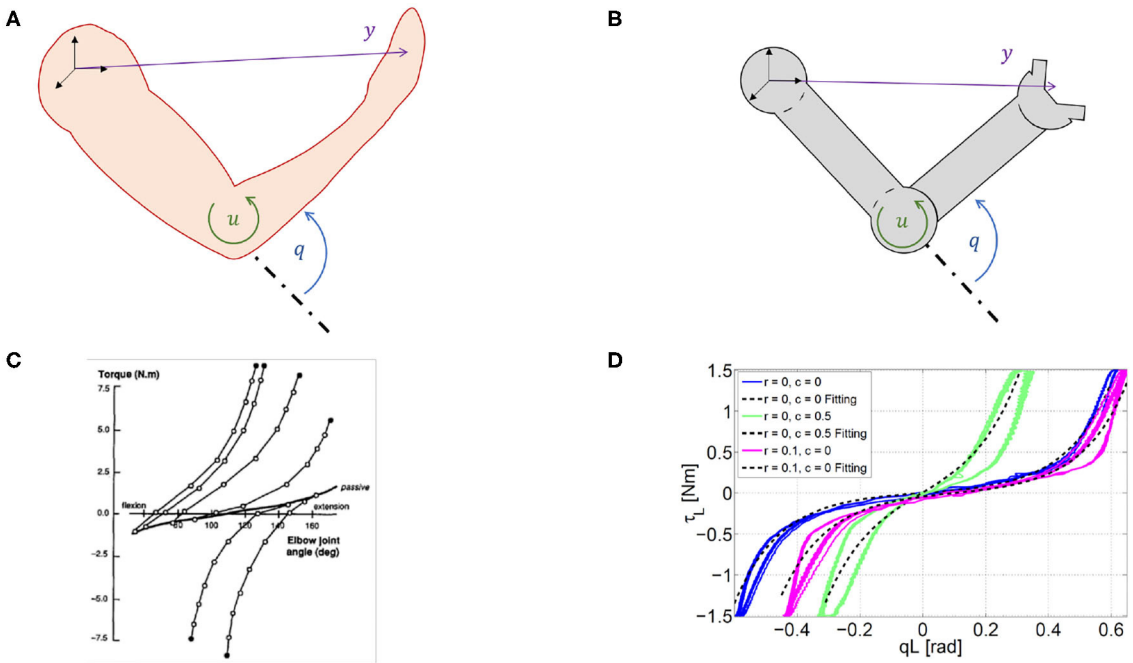


FIGURE 3 | Similarity between humans and robots. Variable definitions in humans **(A)** and robots **(B)**. $q \in \mathbb{R}^n$ are the Lagrangian variables, $x = [q^T, \dot{q}^T]^T \in \mathbb{R}^{2n}$ is the state vector, $u \in \mathbb{R}^m$ is the input and $y \in \mathbb{R}^l$ is the output. These variables are valid both for biological systems and articulated soft robots. Experimentally measured force–length characteristics in natural **(C)** and robotic **(D)** system. **(C)** Elastic characteristic of agonist and antagonist muscles acting on the elbow joint in the human, taken from Gribble et al. (1998). **(D)** Elastic characteristic of a agonist and antagonist variable stiffness actuator (Garabini et al., 2017).

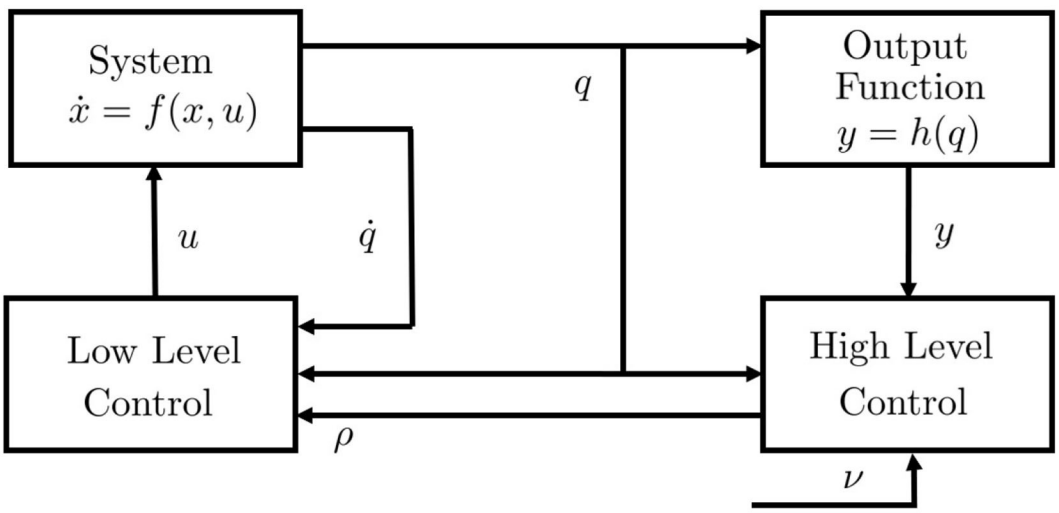


FIGURE 4 | Control structure. u is the low level control variable or efferent action, ρ is the high level control variable, ν is the reference in the task space, q is the position vector, $\dot{q}$ is the speed vector, $x = [q^{(T)}, \dot{q}^{(T)}]^{(T)}$ is the state vector, y is the output vector, $h(\cdot)$ is the output function. The control system is supposed equipped by a complete proprioception.

proposed architecture. We base the high level controller on an optimization problem to define the desired task and to solve the redundancy issue. From this optimization problem a synergistic behavior (iv) results. Finally, to re-plan an anticipatory action (v) we propose two different approaches, one based on proportional control and the other one based on Model Predictive Control (MPC). Both methods will be tested and compared. We also focus on a trade off between problem dimensionality and accuracy.

4.1. Low Level Control

Let us define the error signal as $e := \hat{x} - x$, where x is the measured state vector, while $\hat{x}$ is the desired evolution, given by higher levels of the architecture. In addition, let us define the inverse functional $W : C^1[0, t_f) \rightarrow C^0[0, t_f)$, mapping a desired state trajectory $\hat{x}$ into the input $\hat{u}$ able to track that trajectory. The purpose of the low level controller is to perform dynamic inversion of the system given any desired trajectory $\hat{x}$, thus to find a map approximating W. In addition, we aim at replicating the CNS features (i), (ii) and (iii). To this end, we propose a new algorithm combining Iterative Learning Control (ILC) and Gaussian Process Regression (GPR).

4.1.1. Learning to Track a Trajectory

The learning by repetition behavior (i) can be achieved using a learning technique. Emken et al. (2007) presents a model of learning by repetition process, derived from a statistic model of error evolution over iterations

$$u_{i+1} = \alpha\, u_i + \beta e_i \,, \tag{1}$$

where $\alpha, \beta \in \mathbb{R}^+$ are two positive constants, while u_i and e_i are the control action and the error at the i-th iteration, respectively. In this way an input sequence is iteratively computed such that the output of the system is as close as possible to the desired output. Iterative Learning Control (ILC) (Bristow et al., 2006) permits to embed this rule in a general theory, and already achieved good results when applied to VSA robots (Angelini et al., 2018). ILC exploits the whole previous iteration error evolution to update a feedforward command, according to the law

$$u_{i+1} = L(u_i) + z(e_i) \,, \tag{2}$$

where the function $z(e_i)$ identifies the iterative update, while $L(u_i)$ is a function[1] mapping the control action of the previous iteration u_i into the current one.

While in works, such as Tseng et al. (2007) is described the pure contribution of error signals, there are evidence, such as Kawato (1996), that feedback motor correction plays a crucial role in motor learning. Hence, a more general algorithm able to merge all of these contribution is needed. Thanks to the described inclusion we can design an ILC controller merging both feedback and feedforward, applying a control law, such as

$$u_{i+1} = L(u_i) + z(e_i, e_{i+1}) \,, \tag{3}$$

where the presence of the error of the current iteration e_{i+1} leads to the feedback action. The combination of feedback and feedforward actions, allows to profitably collect sensory-motor memory implementing also the described anticipatory behavior (ii). Furthermore, relying mostly on a feedforward action, ILC allows a limited stiffening of the robot (Della Santina et al., 2017a).

Among all the ILC algorithms, in order to opportunely generalize (1) maintaining its intrinsic model-free structure, in this work we use an PD-ILC law in the form of the ones proposed (e.g., in Shou et al., 2003; Ruan et al., 2007), to obtain a minimal dependence on a model of the system dynamics. The proposed approach has been already preliminarily introduced in Angelini et al. (2020a). The adopted iterative update is

$$z(t, i) = \Gamma_{\mathrm{FFp}}\, e_i(t) + \Gamma_{\mathrm{FFd}}\, \dot{e}_i(t) + \Gamma_{\mathrm{FBp}}\, e_{i+1}(t) + \Gamma_{\mathrm{FBd}}\, \dot{e}_{i+1}(t) \,, \tag{4}$$

where, e_i is the error evolution at the i-th iteration, $\Gamma_{\mathrm{FFp}} \in \mathbb{R}^{m\times 2n}$ and $\Gamma_{\mathrm{FFd}} \in \mathbb{R}^{m\times 2n}$ are the PD control gains of the iterative update while $\Gamma_{\mathrm{FBp}} \in \mathbb{R}^{m\times 2n}$ and $\Gamma_{\mathrm{FBd}} \in \mathbb{R}^{m\times 2n}$ are the PD feedback gains. We choose a decentralized structure for the ILC controller, hence, the gain matrices are block diagonal. The gains of the control algorithm can be chosen through several methods. Trial and error approaches could be adopted, but they are usually time consuming and the final performance depends on the experience of the human operator. The ILC framework proposes several techniques to guarantee the convergence of the iterative process depending on the control gains. Thus, other tuning approaches rely on these convergence condition to choose the gains. Some relevant examples of convergence conditions can be found in Arimoto et al. (1984), Ahn et al. (1993), Moore (1999), Bristow et al. (2006), and Wang et al. (2009). In Angelini et al. (2018) an algorithm to automatically tune the control gains is proposed. Finally, it is worth mentioning that the feedback gains should be set low to avoid alteration of the softness of the controlled system (Della Santina et al., 2017a; Angelini et al., 2018).

The adopted solution achieves aftereffect over known trajectories (iii). Indeed, the method is able to compensate also unmodeled potential external force field, because it is model-free and learning based. This means that the learned action depends on the external force disturbances that were present during the learning phase. Furthermore, since the method is mostly feedforward, when the external force field is removed, the system presents the desired aftereffect (iii).

4.1.2. Generalization of the Learned Trajectories

Given a desired trajectory $\hat{x}$, ILC returns an input $\hat{u}$ such that $\hat{u} = W(\hat{x})$, thus it returns a pair $(\hat{x}, W(\hat{x}))$. However, the method lacks of generality. Indeed, ILC is a local method, and it requires a novel learning phase for each novel desired trajectory $\hat{x}$. Conversely, humans are capable of effectively performing novel tasks exploiting and generalizing the previously acquired experiences (Sternad, 2018). Angelini et al. (2020b) proposes a method to generalize the control actions w.r.t. to time execution given a limited set of pairs $(\hat{x}, W(\hat{x}))$. Given a desired trajectory $\hat{x}$, the method allows to track $\hat{x}$ with any desired velocity, without any knowledge of the robot model. In this paper, we are interested in generalizing the learning control action w.r.t. the joint evolution, replicating the feature of human beings. To this end, we apply GPR on a set of learned pairs $(\hat{x}, W(\hat{x}))$, in order to regress a map—approximating W—able to track any novel desired trajectory $\hat{x}$. Then, the system will present also the desired behavior aftereffect over unknown trajectories (iii). This is achieved because the regressed map will be based on the learned feedforward control actions.

[1] $L(u_i)$ is typically a smoothing function.

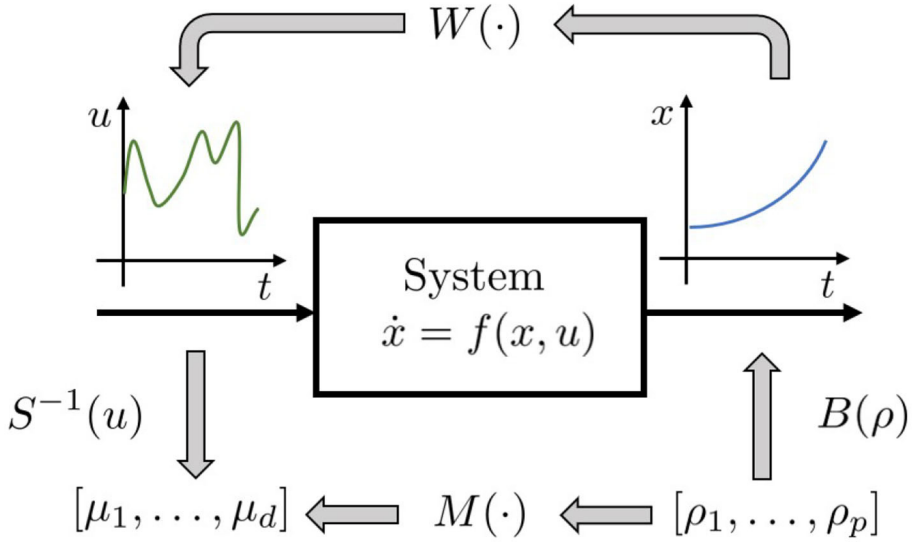


FIGURE 5 | Proposed regression approach: instead of trying to regress the whole inverse functional $W(\cdot)$, the idea is to regress the function $M(\cdot)$, which provides an approximation [defined by $S(\cdot)$] of control action needed to induce a reduced set of evolution [defined by $B(\cdot)$].

Several approaches can be applied to compute the inverse functional W. Some methods contemplate the independent estimation of a complete model of the system (e.g., Arif et al., 2001; Purwin and D'Andrea, 2009). The limitations of complete model estimation (Nguyen-Tuong et al., 2008) approaches are well-known (e.g., computational onerous). Conversely, in our approach we will focus on a reduced space of control actions and trajectories, in order to limit the computational burden.

W is the functional mapping the functional space of the state trajectories into the functional space of the input signals. Computing the regressor of a functional is not a trivial task. For this reason, we reduce the problem complexity limiting our analysis to an approximated solution. In particular we transform the functional W into a function through the introduction of two parameterization functions. Then, we focus on the regressor of this approximated solution.

Let us define:

- a parameterization B of a subspace of the trajectories space $\mathbb{F} \subseteq C^1[0, t_f)$, with dimension p, $B : \mathbb{R}^p \to \mathbb{F}$.
- a parameterization S of a subspace of the input space $\mathbb{V} \subseteq C^0[0, t_f)$, with dimension d, $S : \mathbb{R}^d \to \mathbb{V}$.

The trajectory parameterization B constraints low level controller to manage only a sub-set $\mathbb{F}$ of the possible evolutions. The parameterization S defines an approximation of control actions, reducing them to the ones included in $\mathbb{V}$. Hence, with an abuse of notation, we indicate with S^{-1} the application that, given a control action u, returns the set of parameters that identifies its approximation, and such that $S^{-1}(S(\mu)) = \mu \quad \forall \mu \in \mathbb{R}^d$. Hence $M(\rho) : \mathbb{R}^p \to \mathbb{R}^d$ is so defined

$$M(\rho) : \rho \mapsto S^{-1}(W(B(\rho))) \,. \tag{5}$$

$M(\cdot)$ is the map we are interested for (**Figure 5**). ρ is the array of parameters defining the desired trajectory. The map can then be approximated using a non-linear regression technique. We can then use the approximated map to estimate the control action needed to track a new trajectory. We employ here *Gaussian Process Regression* (GPR), because it achieves good performance, while maintaining low the computational cost. In particular, in the GPR algorithm implementation, we employ the squared exponential as covariance function (Williams and Rasmussen, 2006) described as $k_c(x_1, x_2) = \sigma_f^2 \, e^{\frac{-(x_1 - x_2)^2}{2\gamma^2}} + \sigma_n \delta(x_1 - x_2)$,where $\delta(\cdot)$ is the Kronecker delta, and σ_f, σ_n, and γ are free parameters.

Each novel control action will update the map used for generalization. However, to further limit the number of regressed points, for each pair $(\bar{\rho}, S^{-1}(W(B(\bar{\rho}))))$, we remove all the stored points from the map which are in a sphere of radius δ_{err}, centered in $\bar{\rho}$.

The parametrization of the sub-spaces $\mathbb{F}$ and $\mathbb{V}$ can be chosen freely, with the primary goal of keeping low the method complexity without compromising its generality. Several solutions could be implemented and tested. For instance, $\mathbb{F}$ can be set as a space of polynomial with a fixed order, or as a space of sums of sinusoidal signals. On the other hand, $\mathbb{V}$ can be approximated as a Gaussian space, or simply a discretization of the signal (Herreros et al., 2016).

Regarding the choice of the sub-space $\mathbb{F}$, we would like to adopt trajectories that mimic the human motions. Which are the main characteristics of a motion that make it human-like is still an ongoing debate in literature. In Mombaur et al. (2010), the Authors apply inverse optimal control to define a model of human locomotion path and to exploit it for humanoid robot motion generation. In Tomić et al. (2018) it is studied the problem of human dual-arm motion in presence of

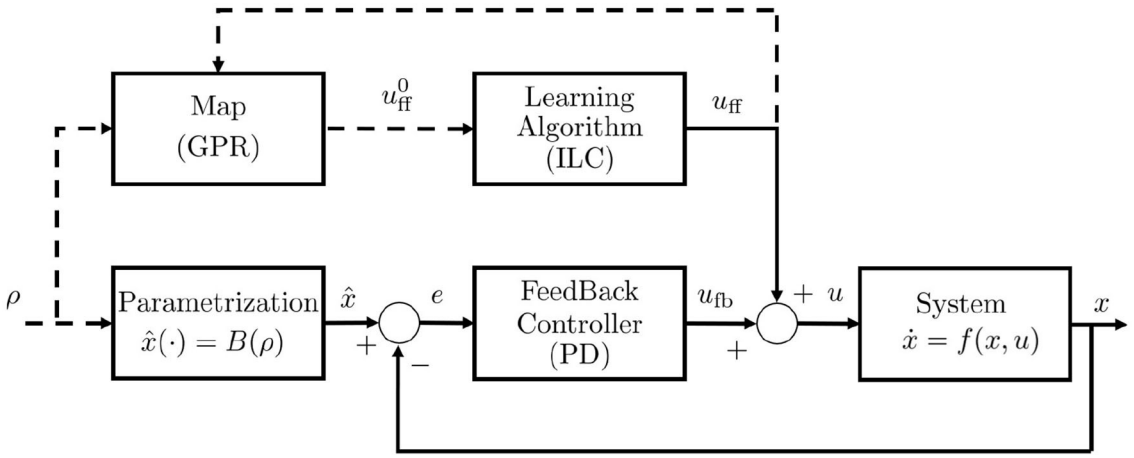


FIGURE 6 | Low level control scheme. $u = u_{\text{fb}} + u_{\text{ff}}$ is the resulting afferent action, and u_{fb} and u_{ff} are respectively the closed loop and the open loop control components, u_{ff}^0 is the a-priori feedforward estimation returned by the map $S^{-1}(W(B(\cdot)))$, ρ is the parameter array, x is the configuration vector. The *Feedback Controller* is a PD controller, the *Learning Algorithm* is the ILC algorithm, the block *Parametrization* implements the $B(\cdot)$ function. Dashed lines indicates flux of information.

contacts with the environment, and it is proposed an algorithm merging inverse optimal control and inverse kinematics to map human motion to humanoid robot motion. An additional method to characterize the human-likeness of robot motion is the adoption of functional synergies directly extracted from human examples as base space (Averta et al., 2017). Without any claim about the solution of this debate, in this work, we adopt the hypothesis formulated in Flash and Hogan (1985) and Friedman and Flash (2009), which states that human movements minimize the jerk. Minimum jerk trajectories are fifth order polynomial (Flash and Hogan, 1985), thus—without any claim of exhaustiveness—we set the vector ρ as the coefficients of the polynomial.

For what concerns the input space parametrization, in this work we focus on piece-wise constant functions with a fixed number d of constant length segments, and we implement S^{-1} as a time discretization, since it is one of the more natural signal approximation in control. Future work will analyze different choices of parametrization of the input and output spaces.

In **Figure 6** we report the resulting low level control scheme. The input ρ is used in the form of $B(\rho)$ as efferent copy for feedback compensation, and through $M(\rho) = u_{\text{ff}}^0$ as estimated anticipatory action. Then, this action can be refined through the learning algorithm. It is worth to be noticed that the proposed low level controller combines learned anticipatory actions and feedback control, working mainly in feedforward when the map reaches the convergence.

It is worth remarking that the adopted solution achieves aftereffect over unknown trajectories (iii). Indeed, the regressed map depends on the learned actions. These actions depend on the external force disturbances that were present during the learning phase. Therefore, when the external force field is removed, the system presents the desired aftereffect (iii).

The acquired control inputs and, more in general, the regressed map depends on the impedance behavior. This was assumed as provided by an higher level of control in this article (section 3). However, future extension of this work will aim at learning the optimal impedance behavior too, imitating the human capabilities (Burdet et al., 2001). In Mengacci et al. (2020) it is presented a method to decouple the control input to track a trajectory and the control input to regulate the robot impedance, removing the dependency between learned control input and desired stiffness profile. This, in combination with GPR, could be used to generalize the acquired control input w.r.t. the desired stiffness profile and the desired task.

4.2. High Level Control

The role of the high level controller is to perform DoFs management in task execution. In particular we are interested in reproducing two of the characteristics of the CNS: synergistic behavior (iv) [i.e., given the desired output $h(x)$, $V_{\text{good}} > V_{\text{bad}}$ in the configuration space] and re-plan of anticipatory action (v).

The degrees of freedom redundancy in humans is classified as *anatomical*, kinematic or *neurophysiological* (section 2). Here we focus on the kinematic redundancy, and the proposed high level control produces a synergistic behavior for this class of synergies. However, we believe that it could be extended also to the anatomical redundancy. Future work will focus on this point. The neurophysiological redundancy does not have a counterpart in robotics, so it is the Authors' opinion that it is not required to deal with it.

Several works report evidences of the discrete nature of the higher levels of the neural control of movements (e.g., Morasso and Ivaldi, 1982; Loram et al., 2011). In particular, in Neilson et al. (1988) is postulated that the CNS does not plan a new movement until the previous one is finished. This happens because the CNS plan a new motion after receiving the desired perceptual consequences of a movement in a finite interval of time. In order to replicate this behavior we choose a time-discrete control approach. Hereinafter we will use the superscript $[k]$, $k \in \mathbb{R}$ to

indicate the k-th planned movement. Each interval will have the same fixed duration t_f.

Low level controller abstracts the largely unknown and non-linear system into a discrete one which depends on the choice of the subspace. As a trade-off between complexity and accuracy, we heuristically chose a smaller subspace: fifth order monic polynomial with two constraints, which reduces space dimension to 3, while ensuring that subspace elements juxtaposition is of class C^2. In particular we will focus on trajectories fulling these constraints

$$\left.\frac{\partial^2 q}{\partial t^2}\right|_{t=\{0,t_f\}} = 0 \ , \qquad q_f = q_s + \dot{q}_f t_f \ , \tag{6}$$

where q_s and q_f are the starting and final values of the polynomials, respectively. Following this choice, we find that $\rho = [q_s, \dot{q}_s, \dot{q}_f]$. Given this definition of ρ, the resulting curve is a polynomial spline, and the abstracted dynamics is a discrete integrator

$$q^{[k+1]} = q^{[k]} + t_f \rho_3^{[k]} \ , \tag{7}$$

where $\rho_3^{[k]}$ is the third element of $\rho^{[k]}$. Note that $\rho_1^{[k]}$ and $\rho_2^{[k]}$ are constrained by the initial conditions, thus they do not appear in (7).

Hence, the high level controller uses ρ as control variable, and its role is to choose the sequence of $\rho_3^{[k]}$, generating a polynomial spline reference.

Level C2 in Bernstein classification (Bemstein, 1967) specifies the task to be accomplished. Analogously, we aim at replicating the same behavior in the proposed high level controller. We define as task a cost function and a set of constraints. Thus, the high level controller is defined by a solver and an optimization problem formulated as

$$\begin{aligned} &\min_{\Delta\rho, q} J(\hat{y} - h(q), q^{[k]}, \Delta\rho_3) \\ &\text{s.t.} \quad \|g_q(q^{[k]})\| \le \lambda_q \ , \forall k \\ &\qquad \|g_\rho(\Delta\rho_3)\| \le \lambda_\rho \\ &\qquad q^{[k+1]} = q^{[k]} + t_f \rho_3^{[k]} \ , \end{aligned} \tag{8}$$

where J is the cost function. $h(\cdot)$ is the output function selecting the variables of interest for the task. $\Delta\rho_3$ is the difference between two consecutive control commands, i.e., at the k-th interval we have $\Delta\rho_3 := \rho_3^{[k]} - \rho_3^{[k-1]}$. g_q and g_ρ are generic constraint functions, while $\lambda_q \in \mathbb{R}$ and $\lambda_\rho \in \mathbb{R}$ are the values of the upper bounds. It is worth noting that $\|\Delta\rho_3\|_R$ assumes the role of actuation cost, while the difference between the desired and the actual output $\|\hat{y} - h(q)\|_Q$ is a metric for performance.

We test two different solvers for the high level control:

- Proportional Control (P): it consists in pre-solving the problem and controlling the system over x_{opt} through a proportional controller, which is a dead beat controller for the discrete integrator if $P = t_f^{-1} I$, with the identity matrix.
- Model Predictive Control (MPC): it consists in recalculating the optimum on-line at each time interval, using the first element of the resulting control sequence (Köhler et al., 2020). Conventionally, MPC is hardly applicable to mechanical systems due to their high bandwidths, but the architecture here presented allows MPC application because it is sufficient to apply it only each t_f seconds.

P control and MPC usually present much different performance and implementation complexity. For this reason, we decided to test both of them to check if a simpler P solver is effective enough, or if the difference in performances can justify the use of a more demanding method, such as MPC.

The high level feedback loop consists in a periodical re-plan of the control sequence, if the actual sensory outcomes are different from the expected ones.

To obtain the desired synergistic behavior (iv), we rely on the uncontrolled manifold theory (Scholz and Schöner, 1999). As briefly described in section 2.2, the uncontrolled manifold is the variance through the directions where output is constant and the constraints are verified. This means that the uncontrolled manifold can be identified as the manifold such that $h(q) - \hat{y} = 0$. Focusing on the regulation of the output, rather than on the joint error, is sufficient to obtain the desired synergistic behavior (iv).

It is worth noting that the quality of the task execution is strongly affected by the accuracy of the learned low level map. A pre-learning of the map is time consuming and generally not required. So, we will use an online approach to generate the map: if a new task is not properly executed (i.e., its error is greater than a certain threshold η_{th}) then the accuracy of the map should be improved through the introduction of a new point, obtained through an ILC execution along the failed trajectory. This approach results in a task-oriented learned map: most of the points will be collected in the portions of the subspace $\mathbb{F}$ that are more useful for the tasks, obtaining a very good trade-off between map dimension and accuracy.

5. VALIDATION

In this section, we test the effectiveness of the proposed control architecture through simulations and experiments. In both cases, we employ as testbed a two degrees of freedom robotic arm, actuated by VSAs (**Figure 7**). Specifically, we employ two *qbmoves Maker Pro* (Della Santina et al., 2017b), which are bio-metitic variable stiffness actuators presenting characteristics similar to human muscles (Garabini et al., 2017). In both validations we consider the following gains for the algorithm Γ_{FFp} is blkdiag($[1, 0.1], [1.25, 0.0375]$), Γ_{FFd} is blkdiag($[0.1, 0.001], [0.0375, 0.001]$), Γ_{FBp} is blkdiag($[0.25, 0.025], [0.25, 0.025]$), and Γ_{FBd} is blkdiag($[0.025, 0.001], [0.025, 0.001]$). The parameters of the squared exponential as covariance function in GPR algorithm are $\sigma_f = 1$, $\sigma_n = 0.05$, $\gamma = 2$, and $\delta_{err} = \pi/20$.

For performance evaluation we use the error norm 1 of the tracking error evolution, i.e., the integral over time of the norm of the error, *mean error* hereinafter. Furthermore, we refer as *total error* evolution the sum of the absolute tracking error of each joint at a given time.

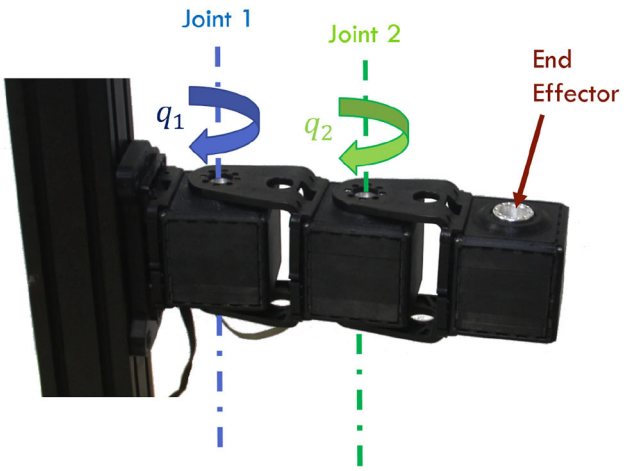


FIGURE 7 | Two degrees of freedom robotic arm used as validation setup. The manipulator is actuated by two *qbmoves Maker Pro*, which are bio-metitic VSAs.

In section 5.1 we present simulations proving that the proposed control architecture presents the desired behaviors (i)–(v) separately. In section 5.2 we present experiments testing the complete control architecture.

5.1. Simulation Results

The employed model is a two degrees of freedom arm. Each link wights 0.5kg and is 0.5m long. Viscous friction equal to 1.2Ns on output shaft is considered. Joints limits are $[0, \frac{\pi}{2}]$rad. The model of the actuators takes into account hardware parameters, such as measure noise, communication delays, saturations, motors dynamics[2]. In the following the test separately the low level and the high level controllers.

5.1.1. Low Level Control

In this section, we verify that the proposed low level control achieves the human-like behaviors described in (i)–(iii). We present a set of three simulations to test each behavior. First, we validate the presence of learning by repetition (i) and anticipatory action (ii). Then, we test the effectiveness of the learned map. Finally, we verify that the system presents aftereffect over know and unknown trajectories (iii).

First, we perform trajectory tracking over 50 trajectories randomly selected in $\mathbb{F}$ through a uniform distribution. Results are shown in **Figure 8**. **Figure 8A** shows that the system profitably implements learning by repetition [behavior (i)], reducing the error by repeating the same movement. **Figure 8B** shows that the controller is able to capitalize the sensory-motor memory over a trajectory increasing the role of anticipatory action [behavior (ii)].

Then, we validate the effectiveness of the map. To this end, we test two scenarios: trajectory tracking without any map and trajectory tracking with a pre-trained map. In the latter case the map is trained on the 50 learning phases performed in the previous simulation. Given the two scenarios, we simulate $2 \cdot 10^3$ trajectories randomly selected in $\mathbb{F}$ through a uniform distribution. The results are reported in **Figure 9**. Results show that the performance using the map learned with only 50 random repetitions are more than one order of magnitude better than the ones without the map, and with a sensibly lower variance.

Finally, we verify the presence of the aftereffect, i.e., behavior (iii). Results are shown in **Figure 10**, specifically we show aftereffect over known trajectories in **Figure 10A**, and aftereffect over unknown trajectories in **Figure 10B**. In the first case, the green asterisk line represents the motion of the robot at the end of the learning phase. Then, we introduce an external force field, which acts on the joints as an external torque described by $\Delta_1(q, \dot{q}) = -\dot{q}_1^3 - 2q_1 + \pi$ and $\Delta_2(q, \dot{q}) = -\dot{q}_2^3 - 0.4q_2$, for the first and second joint, respectively. The trajectory is deformed as a consequence of the force field introduction (diamond red line). We repeat the learning process to recover from performance loss, and the system is again able to follow the initial trajectory (again, green asterisk line). Finally, the field is removed, and the end-effector presents the mirror-image aftereffect, i.e., the trajectory (circle blue line) is specular to the red one.

In the second case we test presence of the aftereffect on unknown trajectories. To this end, we simulate a motor control experiment accounted in Gandolfo et al. (1996). The controller experiences the unknown force field only on two trajectories. In this simulation the external torque is described by $\Delta_1(q, \dot{q}) = -0.5\dot{q}_1 - 0.15$ and $\Delta_2(q, \dot{q}) = -0.5\dot{q}_2 + 0.15$. After field removal, we track five additional trajectories. Each one presents aftereffect. Moreover, its effect is more evident near in the trajectories close to the experienced ones. This result proves that the proposed control architecture presents a typical behavior of the CNS, validating its human resemblance.

5.1.2. High Level

In this section, we verify that the proposed high level control achieves the human-like behaviors described in (iv)–(v). We present a set of two simulations to test each behavior. First, we validate the ability to re-plan an anticipatory action (v) and we compare the two approaches (P and MPC). Then, we verify that the system presents a synergistic behavior (iv).

We evaluate the iterative procedure through 20 tasks. As output we employ the task position of the end-effector along the x axis, i.e., $h(x) = a\cos(q_1) + a\cos(q_1 + q_2)$, where a is the length of both links. Each task consists in moving the arm such that $\|h(x) - \bar{y}_j\|$ is minimized, where $\bar{y}_j$ is the desired evolution of task j. The map is regressed online with a threshold $\eta_{\text{th}} = t_{\text{f}} \frac{\pi}{10} = \frac{\pi}{20}$. This means that there is no pre-learned map and a new learning process is executed each time the tracking error is greater than η_{th}. **Figure 11** shows the result. **Figure 11A** reports the average number of sub-tasks that presents error greater than η_{th} at each iteration. It is worth noting that the map converges to a complete representation of the inverse system, i.e., no more learning is needed, after ∼8 tasks, with both P and MPC algorithms. **Figure 11B** shows that the MPC performance are better than the

[2]The simulink model is available online at www.naturalmachinemotioninitiative.com.

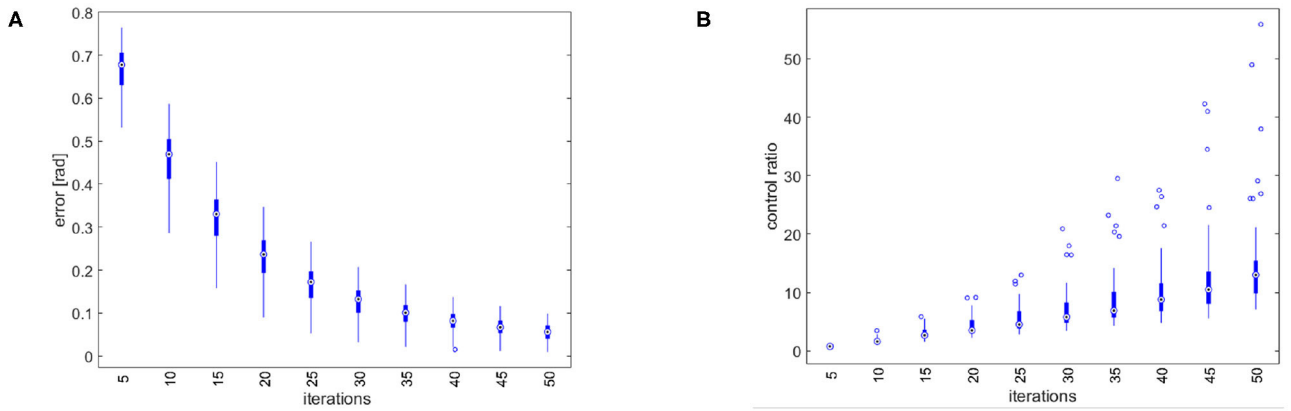


FIGURE 8 | Simulation results of the tracking performance of 50 trajectories randomly selected from $\mathbb{F}$. **(A)** Total error over iterations. The control architecture presents the learning by repetitions behavior. **(B)** Ratio between the feedforward and the feedback action. The control architecture presents the anticipatory behavior.

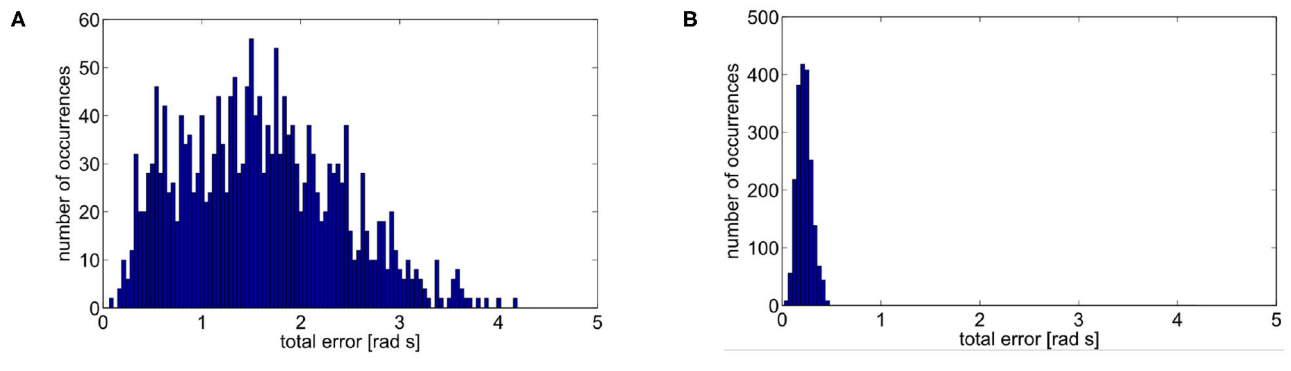


FIGURE 9 | Mean error of $2 \cdot 10^3$ simulations. **(A)** No map is used. The mean error is 1.5929rads with a variance of $0.6272\text{rad}^2\text{s}^2$. **(B)** A learned map is used. The mean error is 0.226rads with a variance of $0.0055\text{rad}^2\text{s}^2$.

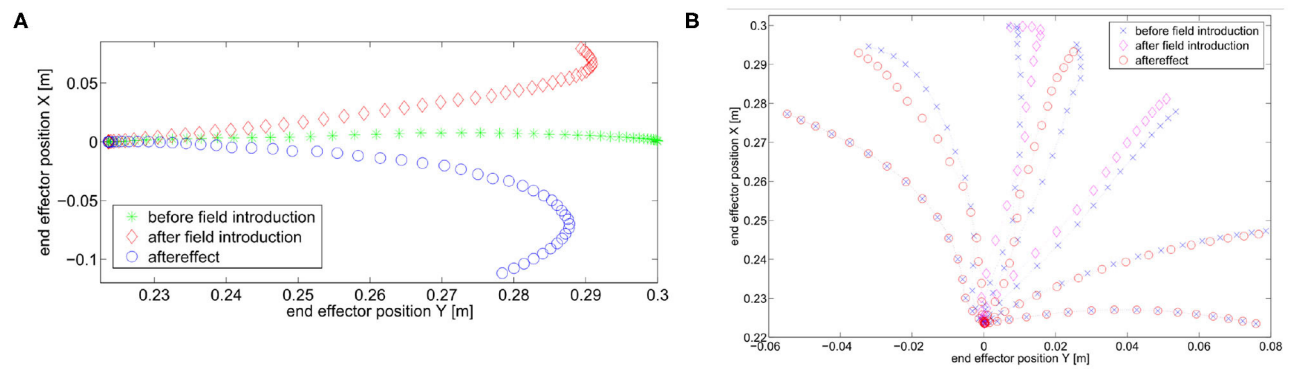


FIGURE 10 | Simulations present aftereffect over known and unknown trajectories. *Before field introduction* are the tracking performance before the introduction of the external force field. The reference trajectory can be considered overlapped. *After field introduction* is the trajectory deformed by the external force field. *Aftereffect* is the trajectory after the field removal. **(A)** Known trajectory. **(B)** Two known trajectories and five unknown trajectories.

P one. This occurs thanks to the re-optimization at each iteration that permits to fully exploit task redundancies. In other terms, if the system moves to a state $\tilde{x}$ different from the desired one $\hat{x}$, but such that $h(\tilde{x}) = h(\hat{x})$, then the P controller reacts trying to regulate the two states to be the same, while the MPC recognizes that the task is accomplished and does not generate any further control action.

In terms of tracking, the P controller presents good performance but worse than MPC. Therefore, due to the greater complexity of the latter method it would be possible to opt for

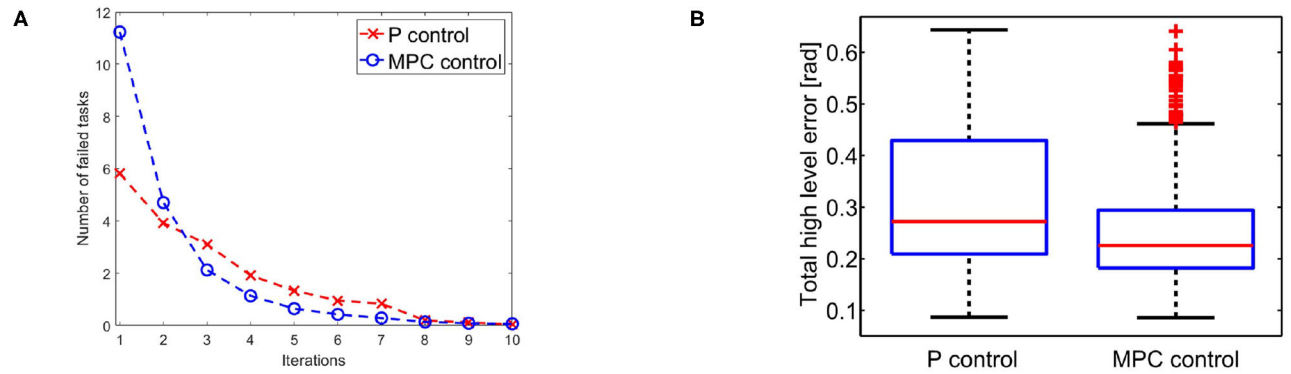


FIGURE 11 | (A) Average number of low level evolution tracking which fails the error test at each iteration. **(B)** Error distributions with the two approaches at the first step of the learning process: the MPC approach presents lower error than P approach exploiting the task redundancy.

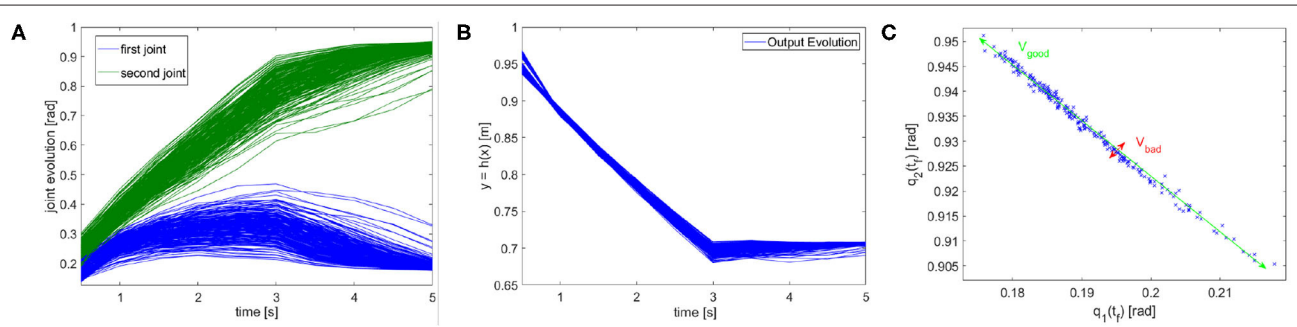


FIGURE 12 | Synergistic behavior. The same task is executed 250 times with randomly selected initial conditions using a normal distribution with standard deviation equal to 0.03 and mean value equal to the correct initial condition value. **(A)** The evolution of the joint present high variability. **(B)** The evolution in the task space presents an analogous behavior, thus the performance are unvaried. **(C)** The distribution in configuration space highlights the synergy-like behavior of the high level controller.

the P controller. However, we are also interested in obtaining a synergistic behavior (iv). To this end, the MPC approach is preferable. To verify the presence of the synergistic behavior (iv), we track a reference trajectory with different initial conditions. In particular, we randomly select 250 initial conditions using a normal distribution with standard deviation equal to 0.03 and mean value equal to the correct initial condition value. **Figure 12A** shows high variability in joints evolution, while **Figure 12B** highlights that the task performance are preserved. Considering the definition of synergy reported in section 4.1, this simulation shows the presence of a synergistic behavior of the controlled system, presenting $V_{\text{good}} >> V_{\text{bad}}$ in the configuration space (**Figure 12C**).

5.2. Experimental Results

In this section we test the complete control architecture, and we verify that it presents the desired behavior (i)–(v). Three experiments are presented, one testing the learning by repetition (i) and anticipatory behavior (ii), one testing the aftereffect (iii), and one testing the performance of the online map learning. It is worth noting that the reference trajectory is provided by the high level control, validating the complete architecture.

The robotic platform is the two degrees of freedom planar arm depicted in **Figure 7**. The output function $h(x)$ is the end-effector position given by $h(x) = [b\cos(q_1) + b\cos(q_1 + q_2)\ ,\ \ b\sin(q_1) + b\sin(q_1 + q_2)]$, where $b = 0.1$m is the length of the links. Given a desired position $\bar{y}$, and a discrete time interval $\bar{k}$, the experimental task is to maximize the velocity of the end effector in the desired position $\bar{y}$ at the desired time step $\bar{k}$. This task can be modeled as the optimization problem

$$\begin{aligned} \min_{\Delta\pi, q}\ & \|\bar{y} - h(q^{[\bar{k}]})\|_{Q_\text{p}} - \|h(q^{[\bar{k}]}) - h(q^{[\bar{k}-1]})\|_{Q_\text{v}} + \|\Delta\pi\|_R \\ \text{s.t.}\ & \underline{\lambda}_\text{q} \geq q^{[k]} \geq \bar{\lambda}_\text{q}\ ,\quad \forall k = 1, \ldots, 10 \\ & q^{[k+1]} = q^{[k]} + t_\text{f}\pi_3^{[k]}\ ,\quad \forall k = 1, \ldots, 9\,, \end{aligned} \tag{9}$$

where $\underline{\lambda}_\text{q}$ and $\bar{\lambda}_\text{q}$ are the joint limits. R, Q_p and Q_v are the weight matrices of the input, the final position cost, and the final velocity, respectively, and their value is set as $R = 0.1\, I_{20\times 20}$, $Q_\text{p} = 20\, I_{2\times 2}$, and $Q_\text{v} = 10\, I_{2\times 2}$.

Figure 13A shows the solution of the optimization problem (9) with parameters $t_\text{f} = 0.5$s, $\underline{\lambda}_\text{q} = [0, 0]^\text{T}$ and $\bar{\lambda}_\text{q} = [\pi/2, \pi/2]^\text{T}$, $\bar{k} = 9$, $\bar{y} = [0.2\ 0]^\text{T}$. This is the reference trajectory of the fist experiment, and it is equal for both joints.

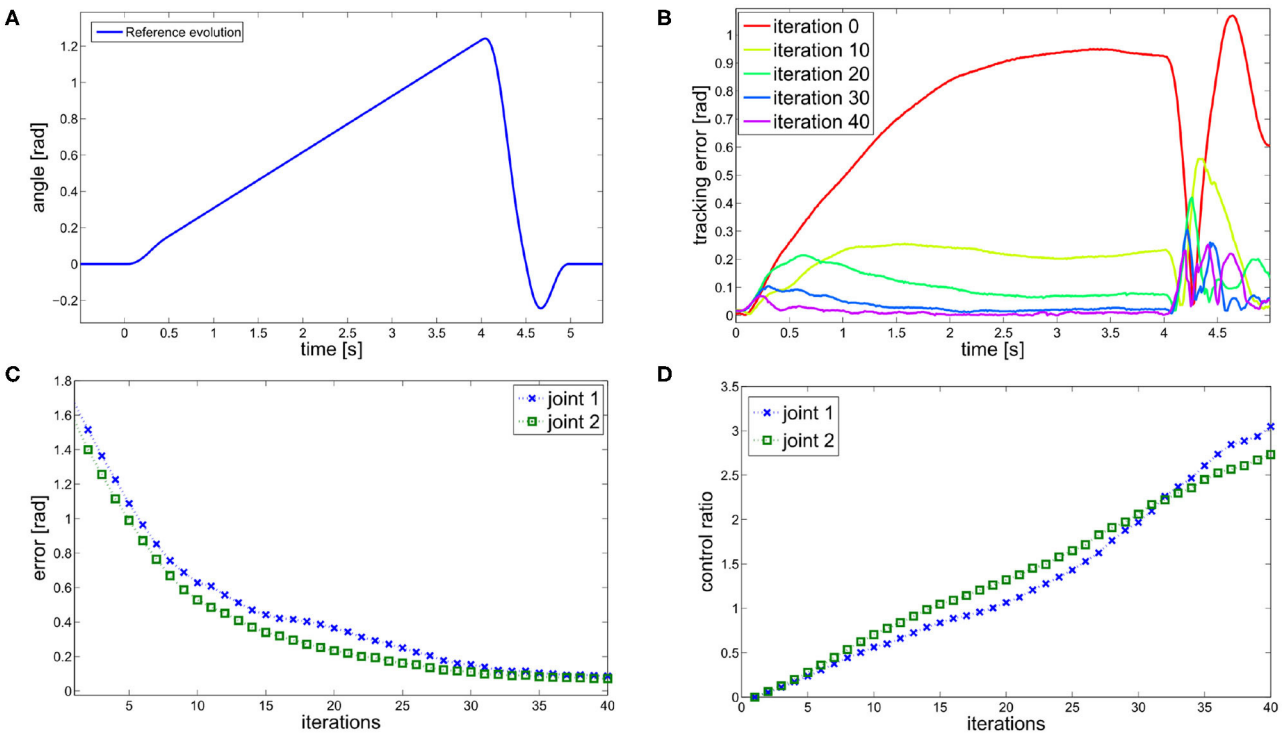


FIGURE 13 | ILC experiment. **(A)** Reference trajectory resulting from the optimization problem (9). The trajectory is equal for both joints. **(B)** Tracking error evolution for different meaningful iteration of the ILC algorithm. **(C)** The evolution of the error over iterations shows the learning by repetitions behavior. **(D)** The ratio between feedforward and feedback actions shows an anticipatory behavior.

The results are shown in **Figure 13**. The proposed algorithm learns the task through repetitions: in 40 iterations the achieved performance are satisfying. **Figure 13B** shows the tracking error evolution over time, for a few meaningful iterations. **Figure 13C** proves that the system implements learning by repetition [behavior (i)], reducing the error exponentially by repeating the same movement. The mean error decreases approximately about 63.7% w.r.t. its initial value in 10 iterations, and of the 95% in 40 iterations. Finally, **Figure 13D** depicts the ratio between total feedforward and feedback action, over learning iterations. This shows the predominance of anticipatory action at the growth of sensory-motor memory [behavior (ii)]. It is worth to be noticed that feedback it is not completely replaced by feedforward, which is coherent with many physiological evidences (e.g., Shadmehr et al., 2010).

The second experiment has two goals. First, it tests the ability of the control algorithm to cope with aggressive external disturbances as springs in a parallel configuration (**Figure 14A**). Then, it validates the presence of mirror-image aftereffect [behavior (iii)]. The robotic arm learns to move its end-effector following the movement depicted in **Figure 14B** (green asterisk line). After the learning process we introduced an external force field. The unknown external force field is generated by a couple of springs of elastic constant 0.05Nm^{-1}, connected as in **Figure 14A**. Due to the spring introduction, the robot end-effector evolution is altered as depicted in **Figure 14B** (red diamond line). At this point, the algorithm recovers the original performance after few iterations, proving its ability to cope with external disturbances (learning process not shown for the sake of clarity). Finally the springs are removed, and the end-effector follows a trajectory (blue circle line in **Figure 14B**), which is the mirror w.r.t. the nominal one, of the one obtained after field introduction, therefore proving the ability of the proposed algorithm to reproduce mirror-image aftereffect [behavior (iii)].

To conclude we test the map in the complete control architecture. The idea is to repeatedly perform similar tasks, and to quantify the map performance. In particular, we are interested in verifying that the map capitalizes upon the information of the previous task executions in the new trials. In this experiment, we sequentially perform 10 tasks. The task parameters are $t_f = 0.5$s, $\underline{\lambda}_q = [0,\ 0]^T$ and $\bar{\lambda}_q = [\pi/2,\ \pi/2]^T$, and $\bar{y} = [0.2\ \ 0]^T$. In this experiment, $\bar{k}$ is chosen randomly with a uniform distribution in the interval $\{2, \ldots, 10\}$ for each task. This means that each task aims to maximize the link velocity at a different time step. The resulting trajectory has a form similar to the one depicted in **Figure 13A**, eventually scaled on the abscissa axis respect to the value of $\bar{k}$, and on the ordinate respect to the values of $\underline{\lambda}_q$ and $\bar{\lambda}_q$: the system moves as slow as possible (i.e., in $\bar{k} - 1$ steps) in the configuration that is most distant from the starting point (i.e., $\bar{\lambda}_q$), then in a time step it moves at the maximal possible speed to the initial position, finally it remains stationary.

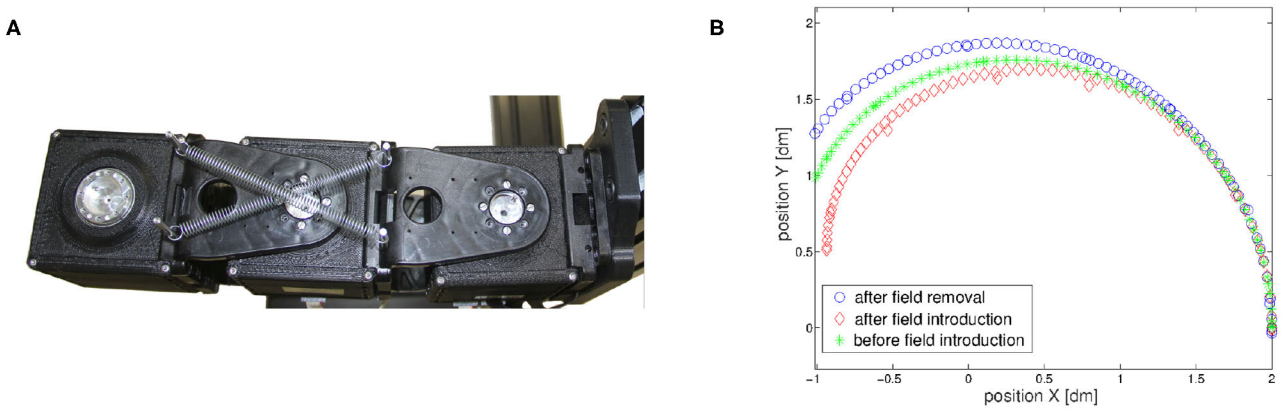


FIGURE 14 | The designed control architecture presents aftereffect on known trajectories. **(A)** An unknown external force field is applied to the robotic arm through the addition of springs. **(B)** The introduction of the force field deforms the trajectory (red line) After some repetitions the strict movement is recovered. If the force field is then removed the trajectory (blue line) is deformed in a way specular to the first deformation.

For each task we performed a learning process lasting for 40 iterations. The resulting low level control is used for map regression. This process is repeated 20 times. Hereinafter each of these repetition is referred as *trial*. To analyze the results we define two error metrics E and I^i. For every i-th task in the j-th trial we evaluate (i) $e^{i,j}_{\text{nm}}$, i.e., the tracking error without the use of the map, and (ii) $e^{i,j}_{\text{wm}}$, i.e., the tracking error with the map learned with previous trajectories.

It is worth to be noticed that both error values $e^{i,j}_{\text{nm}}$ and $e^{i,j}_{\text{wm}}$ are not correlated with index j. However, while $e^{i,j}_{\text{nm}}$ is neither correlated with index i, $e^{i,j}_{\text{wm}}$ appears to be correlated with task i, due to the presence of the map.

What we are interested in evaluating is how much the error $e^{i,j}_{\text{wm}}$ decreases respect to the performance without map $e^{i,j}_{\text{nm}}$. Hence we define the metric

$$E = \frac{1}{N_{\text{i}} N_{\text{j}}} \sum_{\substack{i=1,\ldots,N_{\text{i}} \\ j=1,\ldots,N_{\text{j}}}} \left(\frac{1}{T} \int_0^T \| e^{i,j}_{\text{nm}}(t) \| \text{d}t \right) , \tag{10}$$

where $T = 10t_{\text{f}}$ is the task duration, $N_{\text{i}} = 10$ is the number of tasks in a sequence of learning, $N_{\text{j}} = 20$ is the number of trials. Hence E is the mean value of error without map, and it will be used for normalization purpose.

Therefore the considered error index for the i-th task is defined as

$$I^i = \frac{1}{E} \frac{1}{N_{\text{j}}} \sum_{j=1,\ldots,N_{\text{j}}} \left(\frac{1}{T} \int_0^T \| e^{i,j}_{\text{wm}}(t) \| \text{d}t \right) . \tag{11}$$

I^i represents the normalized mean controlled system behavior over trials at the i-th task. $I^i > 1$ indicates that the map degrades the performance of the system, $I^i = 1$ indicates that the map does not modify the system behavior, $I^i \in [0, 1)$ indicates that the map increases the system performance.

However, it is worth noticing that the regressed map has the goal of improving the performance also of trajectories that differ from the ones stored in the map itself. In particular, the regressed map aims at improving the performance of *dynamically similar* tasks, while maintaining unaltered the performance of *dynamically different* tasks. To analyze this point, we test it in presence of a novel different trajectory w. I^i_{w} represent index (11) for the novel reference. Specifically, the employed trajectories are: s, i.e., *dynamically similar*, and r, i.e., *dynamically different*

$$s_k = \frac{\pi}{4} \sin\left(\frac{3\pi}{2} k\right) \begin{bmatrix} 1 \\ 1 \end{bmatrix} , \qquad r_k = \frac{\pi}{4} \sin\left(\frac{3\pi}{2} k\right) \begin{bmatrix} -2 \\ 1 \end{bmatrix} . \tag{12}$$

The two trajectories are presented in **Figures 15A,B**, respectively. It is worth noticing that the s motion is more similar to the task trajectories than the r motion since both joint evolution are concordant.

This experiment has been performed with two different scenarios: low and high stiffness. The results are reported in **Figures 15C,D**, respectively. Both figures show that the map converges to a complete inversion of the system in the set of tasks of interest in ∼5 iterations, i.e., when five tasks are included in the map there is no more improvement and the best performance are achieved. Furthermore, the method is able to reduce the error on the trajectory dynamically similar, without degrading the performance of the trajectory dynamically different. This result is achieved both in the low stiffness case and in the high one.

6. CONCLUSIONS AND FUTURE WORK

In this work a novel control architecture that simultaneously shows the main characteristics of human motor control system (learning by repetition, anticipatory behavior, aftereffect, synergies) has been stated. The effectiveness of the proposed control framework has been validated in simulations and via experimental tests. The experiments have been conducted on a robotic platform, the qbmoves, closely resembling

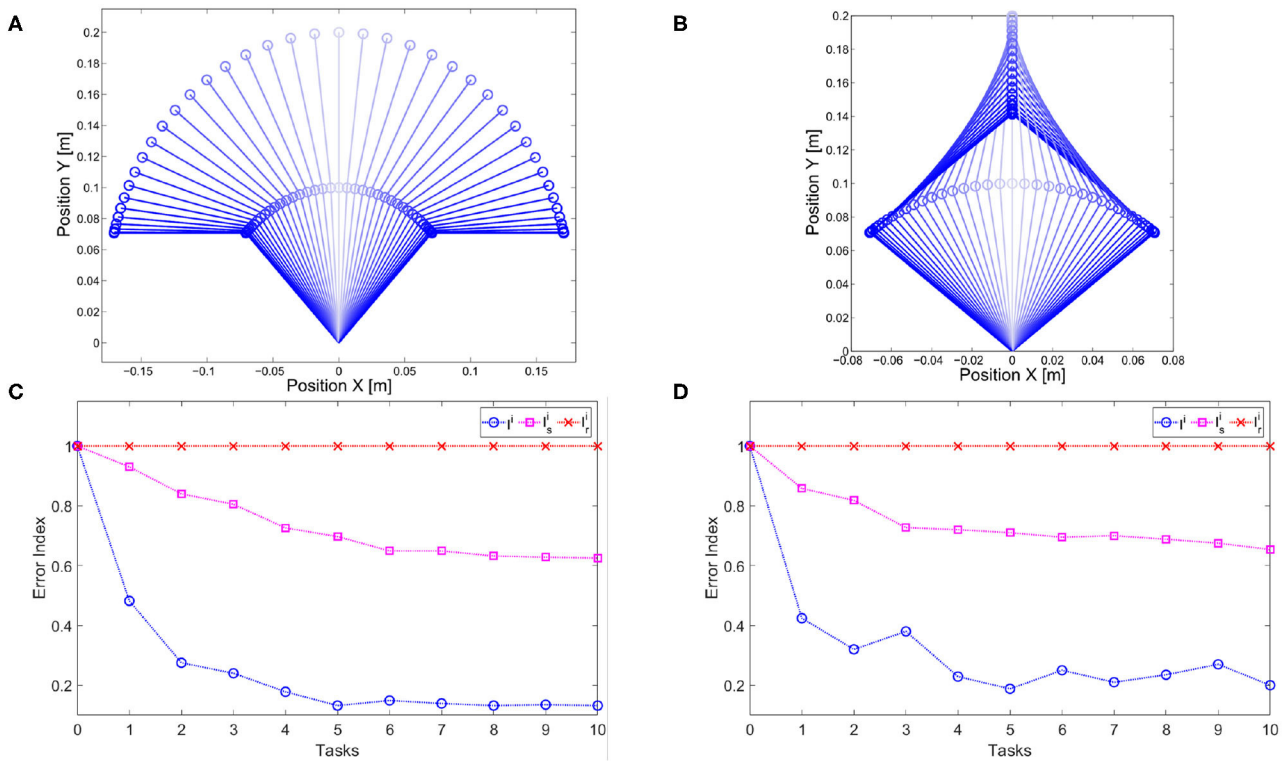


FIGURE 15 | Experimental map evaluation. **(A)** *Dynamically similar* trajectory. **(B)** *Dynamically different* trajectory. Evolution of the error index (11) used for map evaluation in soft **(C)** and stiff **(D)** scenario. The error index I^i on the set of tasks of interest converges to the best reachable performance after ∼5 tasks in both cases. Then, two different trajectories are tested: *s* which is *dynamically similar* and *r* which is *dynamically different*. The map reduces the error on the *dynamically similar* trajectory (I^i_s), and it leaves unadulterated the performance on the *dynamically different* trajectory (I^i_r).

the muscular system and in which the control inputs, namely reference position and stiffness preset, have their biological counterpart in the reciprocal and co-activation, as per Equilibrium Point Hypothesis. The proposed control architecture translates elements of the main motor control theories in well-stated mechanisms belonging to control theory. Control Engineering could provide a useful framework for theory falsification in motor control, and it could give an already well-formed global language for problem definition. Furthermore, human behavior can be used to ensure human-like performance in robotic systems, and hence be used as a starting point for novel control models. We will further analyze this point in future work.

Future work will also aim at increasing the human-likeness of the proposed control architecture. First we will focus on merging the generalization method proposed in Angelini et al. (2020b) and the generalization method based on GPR that was presented in this paper. The union of the two approaches will grant to the robot the ability to track any desired trajectory, with any desired velocity, considerably limiting the amount of required learning procedures. This solution will further close the gap between robot and human capability in terms of previous experience exploitation. Then, we will aim at replicating the impedance behavior learning that is typical of human beings, and it is generally related to the performed task. Indeed, thanks to our control architecture the robot compliance is not altered, meaning that it can be freely exploited. Additionally, we will exploit functional synergies extracted from recorded human motions to increase the human-likeness of the robot movements (Averta et al., 2020). Finally, this work focused on robot powered by mono-articular actuators, i.e., platforms where each motor separately drives each link. However, some systems, e.g., human musculoskeletal system, present a poly-articular structure. In Mengacci et al. (2020), a few preliminary insights about the application of ILC to poly-articular systems have been discussed. Starting from these results, future work will also study the application of the proposed control architecture to poly-articular robots, achieving also a anatomical synergistic behavior.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

FA and CD developed the method and equally contributed to the paper. CD performed the experiments. All authors conceived the idea together and contributed to writing the manuscript.

FUNDING

This project has been supported by European Union's Horizon 2020 research and innovation programme under grant agreement 780883 (THING) and 871237 (Sophia), by ERC Synergy Grant 810346 (Natural BionicS) and by the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence).

REFERENCES

Ahn, H.-S., Choi, C.-H., and Kim, K.-B. (1993). Iterative learning control for a class of nonlinear systems. *Automatica* 29, 1575–1578. doi: 10.1016/0005-1098(93)90024-N

Albu-Schaffer, A., Eiberger, O., Grebenstein, M., Haddadin, S., Ott, C., Wimbock, T., et al. (2008). Soft robotics. *IEEE Robot. Autom. Mag.* 15, 20–30. doi: 10.1109/MRA.2008.927979

Angelini, F., Bianchi, M., Garabini, M., Bicchi, A., and Della Santina, C. (2020a). "Iterative learning control as a framework for human-inspired control with bio-mimetic actuators," in *Biomimetic and Biohybrid Systems. Living Machines 2020. Lecture Notes in Computer Science* (Cham: Springer).

Angelini, F., Della Santina, C., Garabini, M., Bianchi, M., Gasparri, G. M., Grioli, G., et al. (2018). Decentralized trajectory tracking control for soft robots interacting with the environment. *IEEE Trans. Robot.* 34, 924–935. doi: 10.1109/TRO.2018.2830351

Angelini, F., Mengacci, R., Della Santina, C., Catalano, M. G., Garabini, M., Bicchi, A., et al. (2020b). Time generalization of trajectories learned on articulated soft robots. *IEEE Robot. Autom. Lett.* 5, 3493–3500. doi: 10.1109/LRA.2020.2977268

Ansari, Y., Laschi, C., and Falotico, E. (2019). "Structured motor exploration for adaptive learning-based tracking in soft robotic manipulators," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)* (Seoul: IEEE), 534–539. doi: 10.1109/ROBOSOFT.2019.8722767

Arif, M., Ishihara, T., and Inooka, H. (2001). Incorporation of experience in iterative learning controllers using locally weighted learning. *Automatica* 37, 881–888. doi: 10.1016/S0005-1098(01)00030-9

Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering operation of robots by learning. *J. Robot. Syst.* 1, 123–140. doi: 10.1002/rob.4620010203

Asano, Y., Kozuki, T., Ookubo, S., Kawamura, M., Nakashima, S., Katayama, T., et al. (2016). "Human mimetic musculoskeletal humanoid kengoro toward real world physically interactive actions," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)* (Cancun: IEEE), 876–883. doi: 10.1109/HUMANOIDS.2016.7803376

Averta, G., Della Santina, C., Battaglia, E., Felici, F., Bianchi, M., and Bicchi, A. (2017). Unvealing the principal modes of human upper limb movements through functional analysis. *Front. Robot. AI* 4:37. doi: 10.3389/frobt.2017.00037

Averta, G., Della Santina, C., Valenza, G., Bicchi, A., and Bianchi, M. (2020). Exploiting upper-limb functional principal components for human-like motion generation of anthropomorphic robots. *J. Neuroeng. Rehabil.* 17, 1–15. doi: 10.1186/s12984-020-00680-8

Bemstein, N. (1967). *The Co-ordination and Regulation of Movements*. Oxford: Pergamon Press Ltd.

Bristow, D. A., Tharayil, M., and Alleyne, A. G. (2006). A survey of iterative learning control. *IEEE Control Syst.* 26, 96–114. doi: 10.1109/MCS.2006.1636313

Buondonno, G., and De Luca, A. (2016). Efficient computation of inverse dynamics and feedback linearization for vsa-based robots. *IEEE Robot. Autom. Lett.* 1, 908–915. doi: 10.1109/LRA.2016.2526072

Burdet, E., Osu, R., Franklin, D. W., Milner, T. E., and Kawato, M. (2001). The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature* 414, 446–449. doi: 10.1038/35106566

Cao, J., Liang, W., Zhu, J., and Ren, Q. (2018). Control of a muscle-like soft actuator via a bioinspired approach. *Bioinspir. Biomimet.* 13:066005. doi: 10.1088/1748-3190/aae1be

Capolei, M. C., Andersen, N. A., Lund, H. H., Falotico, E., and Tolu, S. (2019). A cerebellar internal models control architecture for online sensorimotor adaptation of a humanoid robot acting in a dynamic environment. *IEEE Robot. Autom. Lett.* 5, 80–87. doi: 10.1109/LRA.2019.2943818

Della Santina, C., Bianchi, M., Grioli, G., Angelini, F., Catalano, M., Garabini, M., et al. (2017a). Controlling soft robots: balancing feedback and feedforward elements. *IEEE Robot. Autom. Mag.* 24, 75–83. doi: 10.1109/MRA.2016.2636360

Della Santina, C., Catalano, M. G., and Bicchi, A. (2020). *Soft Robots*. Berlin; Heidelberg: Springer Berlin Heidelberg. Available online at: https://tinyurl.com/tbdw3g6

Della Santina, C., Piazza, C., Gasparri, G. M., Bonilla, M., Catalano, M. G., Grioli, G., et al. (2017b). The quest for natural machine motion: an open platform to fast-prototyping articulated soft robots. *IEEE Robot. Autom. Mag.* 24, 48–56. doi: 10.1109/MRA.2016.2636366

Emken, J. L., Benitez, R., Sideris, A., Bobrow, J. E., and Reinkensmeyer, D. J. (2007). Motor adaptation as a greedy optimization of error and effort. *J. Neurophysiol.* 97, 3997–4006. doi: 10.1152/jn.01095.2006

Engel, K. C., Flanders, M., and Soechting, J. F. (1997). Anticipatory and sequential motor control in piano playing. *Exp. Brain Res.* 113, 189–199. doi: 10.1007/BF02450317

Flanagan, J. R., and Wing, A. M. (1993). Modulation of grip force with load force during point-to-point arm movements. *Exp. Brain Res.* 95, 131–143. doi: 10.1007/BF00229662

Flash, T., and Hogan, N. (1985). The coordination of arm movements: an experimentally confirmed mathematical model. *J Neurosci.* 5, 1688–1703. doi: 10.1523/JNEUROSCI.05-07-01688.1985

Friedman, J., and Flash, T. (2009). Trajectory of the index finger during grasping. *Exp. Brain Res.* 196, 497–509. doi: 10.1007/s00221-009-1878-2

Fu, Q., Zhang, W., and Santello, M. (2010). Anticipatory planning and control of grasp positions and forces for dexterous two-digit manipulation. *J. Neurosci.* 30, 9117–9126. doi: 10.1523/JNEUROSCI.4159-09.2010

Gandolfo, F., Mussa-Ivaldi, F., and Bizzi, E. (1996). Motor learning by field approximation. *Proc. Natl. Acad. Sci. U.S.A.* 93, 3843–3846. doi: 10.1073/pnas.93.9.3843

Garabini, M., Santina, C. D., Bianchi, M., Catalano, M., Grioli, G., and Bicchi, A. (2017). "Soft robots that mimic the neuromusculoskeletal system," in *Converging Clinical and Engineering Research on Neurorehabilitation II. Biosystems & Biorobotics, Vol. 15*, eds J. Ibáñez, J. González-Vargas, J. Azorín, M. Akay and J. Pons (Cham: Springer). doi: 10.1007/978-3-319-46669-9_45

Gribble, P. L., Ostry, D. J., Sanguineti, V., and Laboissière, R. (1998). Are complex control signals required for human arm movement? *J. Neurophysiol.* 79, 1409–1424. doi: 10.1152/jn.1998.79.3.1409

Haith, M. M., Hazan, C., and Goodman, G. S. (1988). Expectation and anticipation of dynamic visual events by 3.5-month-old babies. *Child Dev.* 59, 467–479. doi: 10.2307/1130325

Herreros, I., Arsiwalla, X., and Verschure, P. (2016). "A forward model at Purkinje cell synapses facilitates cerebellar anticipatory control," in *Advances in Neural Information Processing Systems 29*, eds D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett (Curran Associates, Inc.), 3828–3836. Available online at: http://papers.nips.cc/paper/6151-a-forward-model-at-purkinje-cell-synapses-facilitates-cerebellar-anticipatory-control.pdf

Hofer, M., Spannagl, L., and D'Andrea, R. (2019). "Iterative learning control for fast and accurate position tracking with an articulated soft robotic arm," in *Conference: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Macau), 6602–6607. doi: 10.1109/IROS40897.2019.8967636

Hoffmann J. (2003). "Anticipatory behavioral control," in *Anticipatory Behavior in Adaptive Learning Systems. Lecture Notes in Computer Science, Vol. 2684*, eds M. V. Butz, O. Sigaud, and P. Gérard (Berlin, Heidelberg: Springer). doi: 10.1007/978-3-540-45002-3_4

Hordacre, B., and McCambridge, A. (2018). "Motor control: structure and function of the nervous system," in *Neurological Physiotherapy Pocketbook E-Book*, eds S. Lennon, G. Ramdharry, and G. Verheydenlug (Elsevier Health Sciences), 21.

Jäntsch, M., Wittmeier, S., Dalamagkidis, K., Panos, A., Volkart, F., and Knoll, A. (2013). "Anthrob–a printed anthropomimetic robot," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (Atlanta, GA), 342–347. doi: 10.1109/HUMANOIDS.2013.7029997

Kawato, M. (1996). "2 f\ learning internal models of the motor apparatus," in *The Acquisition of Motor Behavior in Vertebrates (A Bradford Book)*, eds J. R. Bloedel, T. J. Ebner and S. P. Wise (The MIT Press), 409.

Keppler, M., Lakatos, D., Ott, C., and Albu-Schäffer, A. (2018). Elastic structure preserving (esp) control for compliantly actuated robots. *IEEE Trans. Robot.* 34, 317–335. doi: 10.1109/TRO.2017.2776314

Köhler, J., Soloperto, R., Muller, M. A., and Allgower, F. (2020). A computationally efficient robust model predictive control framework for uncertain nonlinear systems. *IEEE Trans. Autom. Control*. doi: 10.1109/TAC.2020.2982585

Kuppuswamy, N., Marques, H. G., and Hauser, H. (2012). "Synthesising a motor-primitive inspired control architecture for redundant compliant robots," in *From Animals to Animats 12. SAB 2012. Lecture Notes in Computer Science, Vol. 7426*, eds T. Ziemke, C. Balkenius, J. Hallam (Berlin; Heidelberg: Springer). doi: 10.1007/978-3-642-33093-3_10

Lackner, J. R., and Dizio, P. (1998). Gravitoinertial force background level affects adaptation to coriolis force perturbations of reaching movements. *J. Neurophysiol.* 80, 546–553. doi: 10.1152/jn.1998.80.2.546

Landkammer, S., Winter, F., Schneider, D., and Hornfeck, R. (2016). Biomimetic spider leg joints: a review from biomechanical research to compliant robotic actuators. *Robotics* 5:15. doi: 10.3390/robotics5030015

Latash, M. L. (2010). Motor synergies and the equilibrium-point hypothesis. *Motor Control* 14:294. doi: 10.1123/mcj.14.3.294

Latash, M. L. (2012). *Fundamentals of Motor Control*. Academic Press.

Lee, J., Huber, M. E., Stemad, D., and Hogan, N. (2018). "Robot controllers compatible with human beam balancing behavior," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE), 3335–3341. doi: 10.1109/IROS.2018.8593549

Loram, I. D., Gollee, H., Lakie, M., and Gawthrop, P. J. (2011). Human control of an inverted pendulum: is continuous control necessary? Is intermittent control effective? Is intermittent control physiological? *J. Physiol.* 589, 307–324. doi: 10.1113/jphysiol.2010.194712

Marques, H. G., Jäntsch, M., Wittmeier, S., Holland, O., Alessandro, C., Diamond, A., et al. (2010). "Ecce1: the first of a series of anthropomimetic musculoskeletal upper torsos," in *2010 10th IEEE-RAS International Conference on Humanoid Robots* (Nashville: IEEE), 391–396. doi: 10.1109/ICHR.2010.5686344

Medina, J. R., Börner, H., Endo, S., and Hirche, S. (2019). Impedance-based gaussian processes for modeling human motor behavior in physical and non-physical interaction. *IEEE Trans. Biomed. Eng.* 66, 2499–2511. doi: 10.1109/TBME.2018.2890710

Mengacci, R., Angelini, F., Catalano, M., G., Grioli, G., Bicchi, A., et al. (2020). On the motion/stiffness decoupling property of articulated soft robots with application to model-free torque iterative learning control. *Int. J. Robot. Res.*

Mombaur, K., Truong, A., and Laumond, J.-P. (2010). From human to humanoid locomotion—an inverse optimal control approach. *Auton. Robots* 28, 369–383. doi: 10.1007/s10514-009-9170-7

Moore, K. L. (1999). "Iterative learning control: an expository overview," in *Applied and Computational Control, Signals, and Circuits*, editor B. N. Datta (Boston, MA: Birkhäuser). doi: 10.1007/978-1-4612-0571-5_4

Morasso, P., and Ivaldi, F. M. (1982). Trajectory formation and handwriting: a computational model. *Biol. Cybernet.* 45, 131–142. doi: 10.1007/BF00335240

Neilson, P., Neilson, M., and O'dwyer, N. (1988). Internal models and intermittency: a theoretical account of human tracking behavior. *Biol. Cybernet.* 58, 101–112. doi: 10.1007/BF00364156

Nguyen-Tuong, D., Peters, J., Seeger, M., and Schölkopf, B. (2008). "Learning inverse dynamics: a comparison," in *European Symposium on Artificial Neural Networks* (Belgium).

Pfeil, S., Henke, M., Katzer, K., Zimmermann, M., and Gerlach, G. (2020). A worm-like biomimetic crawling robot based on cylindrical dielectric elastomer actuators. *Front. Robot. AI* 7:9. doi: 10.3389/frobt.2020.00009

Pratt, G. A., and Williamson, M. M. (1995). "Series elastic actuators," in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, Vol. 1 (Pittsburgh, PA: IEEE), 399–406. doi: 10.1109/IROS.1995.525827

Purwin, O., and D'Andrea, R. (2009). "Performing aggressive maneuvers using iterative learning control," in *ICRA'09. IEEE International Conference on Robotics and Automation, 2009* (Kobe: IEEE), 1731–1736. doi: 10.1109/ROBOT.2009.5152599

Roberts, T. J., and Azizi, E. (2011). Flexible mechanisms: the diverse roles of biological springs in vertebrate movement. *J. Exp. Biol.* 214, 353–361. doi: 10.1242/jeb.038588

Ruan, X., Bien, Z. Z., and Park, K.-H. (2007). Decentralized iterative learning control to large-scale industrial processes for nonrepetitive trajectory tracking. *IEEE Trans. Syst. Man Cybernet. A Syst. Hum.* 38, 238–252. doi: 10.1109/TSMCA.2007.909549

Scholz, J. P., and Schöner, G. (1999). The uncontrolled manifold concept: identifying control variables for a functional task. *Exp. Brain Res.* 126, 289–306. doi: 10.1007/s002210050738

Shadmehr, R., and Mussa-Ivaldi, F. A. (1994). Adaptive representation of dynamics during learning of a motor task. *J. Neurosci.* 14, 3208–3224. doi: 10.1523/JNEUROSCI.14-05-03208.1994

Shadmehr, R., Smith, M. A., and Krakauer, J. W. (2010). Error correction, sensory prediction, and adaptation in motor control. *Annu. Rev. Neurosci.* 33, 89–108. doi: 10.1146/annurev-neuro-060909-153135

Shou, J., Pi, D., and Wang, W. (2003). "Sufficient conditions for the convergence of open-closed-loop pid-type iterative learning control for nonlinear time-varying systems," in *IEEE International Conference on Systems, Man and Cybernetics, 2003*, Vol. 3 (Washington, DC: IEEE), 2557–2562.

Soechting, J., and Lacquaniti, F. (1983). Modification of trajectory of a pointing movement in response to a change in target location. *J. Neurophysiol.* 49, 548–564. doi: 10.1152/jn.1983.49.2.548

Sternad, D. (2018). It's not (only) the mean that matters: variability, noise and exploration in skill learning. *Curr. Opin. Behav. Sci.* 20, 183–195. doi: 10.1016/j.cobeha.2018.01.004

Swanson, L. W. (2012). *Brain Architecture: Understanding the Basic Plan*. Oxford: Oxford University Press.

Tomić, M., Jovanović, K., Chevallereau, C., Potkonjak, V., and Rodić, A. (2018). Toward optimal mapping of human dual-arm motion to humanoid motion for tasks involving contact with the environment. *Int. J. Adv. Robot. Syst.* 15:1729881418757377. doi: 10.1177/1729881418757377

Tseng, Y., Diedrichsen, J., Krakauer, J. W., Shadmehr, R., and Bastian, A. J. (2007). Sensory prediction errors drive cerebellum-dependent adaptation of reaching. *J. Neurophysiol.* 98, 54–62. doi: 10.1152/jn.00266.2007

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

Wang, Y., Gao, F., and Doyle, III, F. J. (2009). Survey on iterative learning control, repetitive control, and run-to-run control. *J. Process Control* 19, 1589–1600. doi: 10.1016/j.jprocont.2009.09.006

Williams, C. K. I. and Rasmussen, C. E. (2006). *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT press.

Zhakatayev, A., Rubagotti, M., and Varol, H. A. (2017). Time-optimal control of variable-stiffness-actuated systems. *IEEE/ASME Trans. Mechatron.* 22, 1247–1258. doi: 10.1109/TMECH.2017.2671371

Zhang, J., Sheng, J., O'Neill, C. T., Walsh, C. J., Wood, R. J., Ryu, J.-H., et al. (2019). Robotic artificial muscles: current progress and future perspectives. *IEEE Trans. Robot.* 35, 761–781. doi: 10.1109/TRO.2019.2894371

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.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 18 September 2020
doi: 10.3389/frobt.2020.00114

Customization Methodology for Conformable Grasping Posture of Soft Grippers by Stiffness Patterning

*Jun-Young Lee, Jaemin Eom, Sung Yol Yu and Kyujin Cho**

Biorobotis Laboratory, Department of Mechanical Engineering, Seoul National University, Seoul, South Korea

Soft grippers with soft and flexible materials have been widely researched to improve the functionality of grasping. Although grippers that can grasp various objects with different shapes are important, a large number of industrial applications require a gripper that is targeted for a specified object. In this paper, we propose a design methodology for soft grippers that are customized to grasp single dedicated objects. A customized soft gripper can safely and efficiently grasp a dedicated target object with lowered surface contact forces while maintaining a higher lifting force, compared to its non-customized counterpart. A simplified analytical model and a fabrication method that can rapidly customize and fabricate soft grippers are proposed. Stiffness patterns were implemented onto the constraint layers of pneumatic bending actuators to establish actuated postures with irregular bending curvatures in the longitudinal direction. Soft grippers with customized stiffness patterns yielded higher shape conformability to target objects than non-patterned regular soft grippers. The simplified analytical model represents the pneumatically actuated soft finger as a summation of interactions between its air chambers. Geometric approximations and pseudo-rigid-body modeling theory were employed to build the analytical model. The customized soft grippers were compared with non-patterned soft grippers by measuring their lifting forces and contact forces while they grasped objects. Under the identical actuating pressure, the conformable grasping postures enabled customized soft grippers to have almost three times the lifting force than that of non-patterned soft grippers, while the maximum contact force was reduced to two thirds.

Keywords: soft gripper, pneumatic actuator, stiffness patterning, shape conforming, design customization, pre-grasping posture

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Yang Yang, Nanjing University of Information Science and Technology, China
Zheng Wang, Southern University of Science and Technology, China
Jun Shintake, The University of Electro-Communications, Japan
Li Wen, Beihang University, China

****Correspondence:***
Kyujin Cho
kjcho@snu.ac.kr

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *15 May 2020*
Accepted: *21 July 2020*
Published: *18 September 2020*

Citation:
Lee J-Y, Eom J, Yu SY and Cho K (2020) Customization Methodology for Conformable Grasping Posture of Soft Grippers by Stiffness Patterning. Front. Robot. AI 7:114. doi: 10.3389/frobt.2020.00114

INTRODUCTION

Softness and flexibility of constituting materials allow soft robotic grippers to be adaptive when interacting with objects (Hughes et al., 2016; Shintake et al., 2018). Recent efforts in this domain have paved ways to implement previously unattainable functionalities of robotic grippers. For example, variable stiffness structures (Amend et al., 2012; Cheng et al., 2016; Wei et al., 2016; Al Abeach et al., 2017; Fei et al., 2018) and friction pads (Zhou et al., 2017; Glick et al., 2018) were integrated into soft fingers; novel materials such as edible gelatin (Shintake et al., 2017), self-healing materials (Cheng et al., 2014; Terryn et al., 2015, 2017), and 3D printable materials (MacCurdy et al., 2016; Mutlu et al., 2017; Hu et al., 2018) were used to fabricate soft fingers. Also, some studies

developed new designs of air chamber sections by employing modular approaches or exploiting multiple materials (Milana et al., 2018; Zhang et al., 2018; Park et al., 2019).

Soft grippers, especially those driven by pneumatic actuation, have been in the spotlight for their ability to grasp variously shaped objects and even fragile objects with a simple on-and-off control (Rus and Tolley, 2015; Gorissen et al., 2017). Recent efforts regarding pneumatically actuated soft grippers have resulted in soft gripers appearing in the service and logistics scenes, where the grippers are expected to face unconstrained situations.

On the other hand, industrial sites predefine and constrain every component of their production system settings, to achieve maximum efficiency. Therefore, industrial grippers are most likely to repetitively interact with predefined objects. However, most processes that handle flexible and soft objects are yet to be automated because traditional suction cups and grippers are incapable of safely handling them.

To handle such fragile and soft objects, it is critical to exert a lifting force that matches the weight of the target object; also, contact pressures need to be distributed to reduce concentrated contact forces, which can damage the object. Accordingly, there are studies that control the concentrated contact force through sensor feedback (Su et al., 2020) and increase the contact area between the object and the gripper (Shian et al., 2015; Hao et al., 2018). However, most studies rarely consider the effects of both the contact force and area, simultaneously. In addition, the sensory feedback systems require additional control schemes and resources, which may be burdensome to industrial applications.

In this paper, we introduce design and fabrication methods to develop customized soft grippers with highly conformable pre-grasping finger postures that outline the shapes of specified target objects (**Figure 1**) to reduce the contact pressure by increasing the contact area. Implementing grasping postures that match the outlining shapes of target objects has been studied as one way of designing robotic grippers (Shimoga and Goldenberg, 1992; Hurtado and Melkote, 2001; Dollar and Howe, 2006). The adaptiveness of the current soft gripper is manifested when a gripper's structure is deformed upon contacting an object. During this interaction, force is applied to the target object. By implementing conformable grasping postures to soft grippers, grippers can deform to their predefined shapes that outline the target objects without the presence of contact forces. As a result, compared to non-customized soft grippers, the customized soft grippers not only applied the same lifting force with lower actuating pressures but also applied smaller surface contact forces to objects. The lowered actuating pressures and contact forces correspond to efficiency and safety, respectively.

Designing customized soft grippers could be achieved by investigating the bending behavior of a PneuNet-type soft bending actuator, and by engineering the moments generated by the air chamber and constraint layer sections of the soft bending actuator. PneuNet-type actuators have been widely used in the field of soft robotics, and their design could intuitively be segmentized into a series of air chambers. In addition, fabrication of the PneuNet-type actuators is divided into two main parts: the upper air chamber section and the bottom constraint layer section. The constraint layer refers to the section attached to the air chamber section, as shown in **Figure 1**. Geometric approximations and the pseudo-rigid-body modeling theory were employed to formulate the simplified analytical model. The analytical model describes the interaction between the air chambers and the constraint layer as intersections of moment surfaces. The moment surfaces can be modified and tuned to establish intended conforming grasping postures, by patterning the stiffness of the bottom constraint layer.

The main contributions of this paper are as follows:

- Design methodology for customized soft grippers that are matched to the shape outlines of target objects by patterning stiffness of constraint layers. This approach enables customization without changing the main form factors of the gripper.

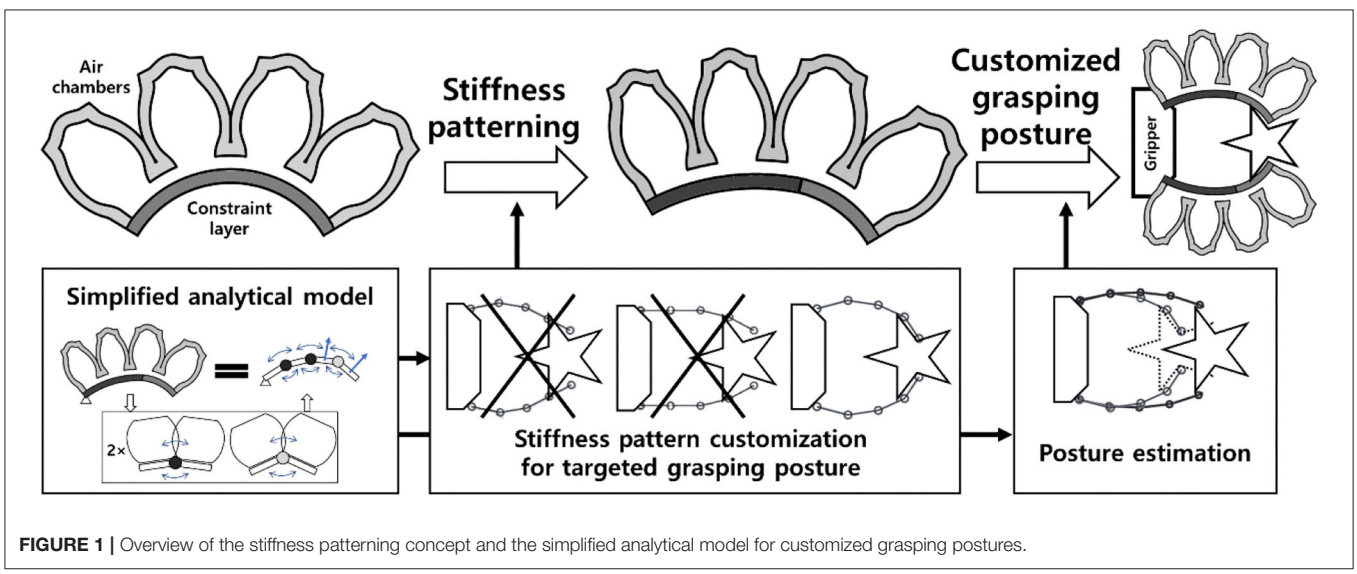


FIGURE 1 | Overview of the stiffness patterning concept and the simplified analytical model for customized grasping postures.

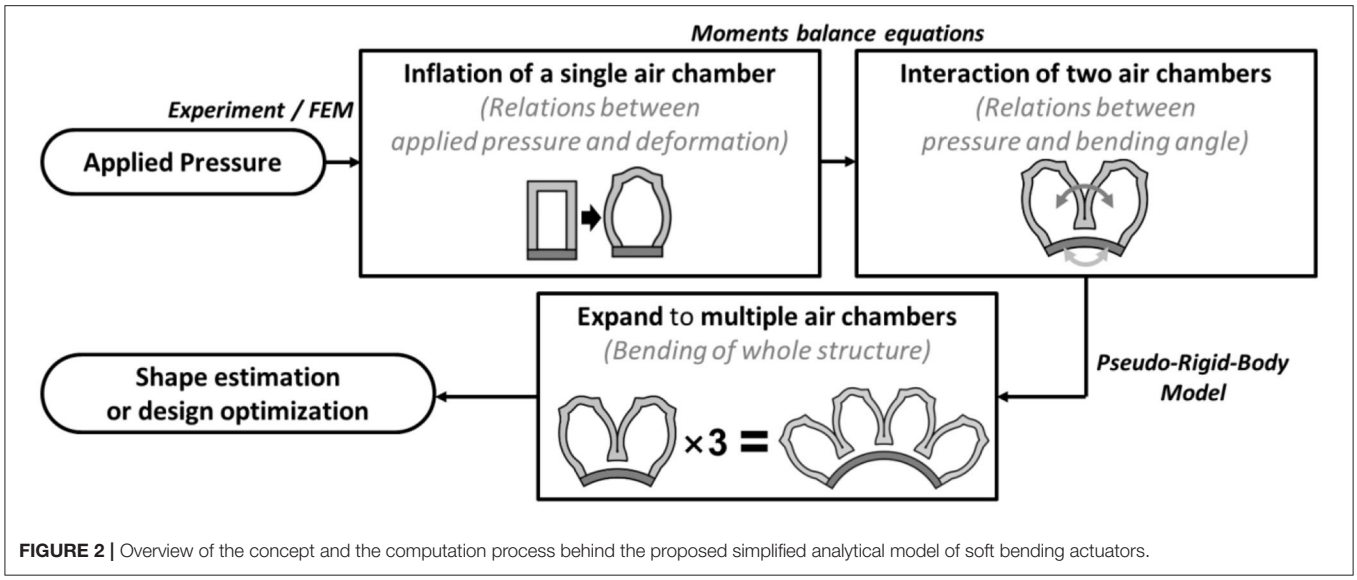


FIGURE 2 | Overview of the concept and the computation process behind the proposed simplified analytical model of soft bending actuators.

- Fabrication methods for customized soft grippers with a modular design approach and a thickness tuning mold approach. The approach enables the manufacturing process of the customized grippers to be cost- and time-efficient.
- A simplified analytical model that estimates and customizes the conforming grasping postures of pneumatically actuated soft bending actuators. The model proposes an insight to analyze and engineer moment surfaces to customize grasping postures. Implementation of the analytical model into the design and fabrication of customized soft grippers with conforming grasping postures.

This report is organized as follows. Section Simplified Analytical Model for Posture Estimation presents a simplified analytical model. In Section Stiffness Patterning of Constraint Layers, discussions about the moment surfaces of soft grippers are presented. Section Fabrication Process of Customized Soft Grippers outlines the fabrication processes of customized soft grippers. Finally, in Section Experimental Results for Customized Soft Gripper, experimental results and comparisons between customized soft grippers and non-patterned soft grippers are presented.

SIMPLIFIED ANALYTICAL MODEL FOR POSTURE ESTIMATION

Building an analytical model that represents the soft actuator's entire structure is challenging. Our approach is to divide the soft bending actuator into a serial arrangement of interactions between two adjacent air chambers. Then, the chain algorithm for the Pseudo-Rigid-Body (PRB) model (Howell, 2001; Pauly and Midha, 2006a,b) was employed to expand and apply the two-chamber interaction into the whole configuration of the soft bending actuator (**Figure 2**).

Model for a Single Interaction Between Two Air Chambers

The first step of the simplified model was to investigate the inflation behaviors of a single air chamber. During inflation, the height and width of the air chamber walls are deformed, as shown in **Figure 3**. In this paper, the height (d_{height}) and width (d_{width}) displacements due to wall inflation are determined based on the fitting curves obtained via experimental measurements (**Supplementary Figures 2**, **3**, **7**), due to the non-linearity of hyper-elastic materials. In future works, it may be possible to investigate the inflation behaviors of a single air chamber with numerical analysis methods.

The bending motion of a soft finger is initiated when two air chambers push each other as they come into contact with each other, after being inflated. In this research, we use chambers with square adjacent facets, which have an approximately circular shape when inflated, as shown in **Figure 3**. The interaction between the two air chambers may exhibit non-linear characteristics that originate from the material properties and irregular distortion of the inflated chambers. This non-linear distortion of the air chambers was assumed to be negligible, and the model consisted of two circular balloons pushing against each other. Based on this geometric approximation, the radius of the inflated chamber wall (R) can be obtained as a function of d_{height} and d_{width} as **Equation (2)**. Height of the single air chamber L can be described with initial height ($L_{initial}$) and displacement by inflation (**Figure 3B**).

$$L = L_{initial} + d_{height} \tag{1}$$

$$R = \frac{d_{width}}{2} + \frac{L^2}{8 \cdot d_{width}} \tag{2}$$

The bending moment, $M_{chamber}$, originated from the pushing force between air chambers, could be described using geometric

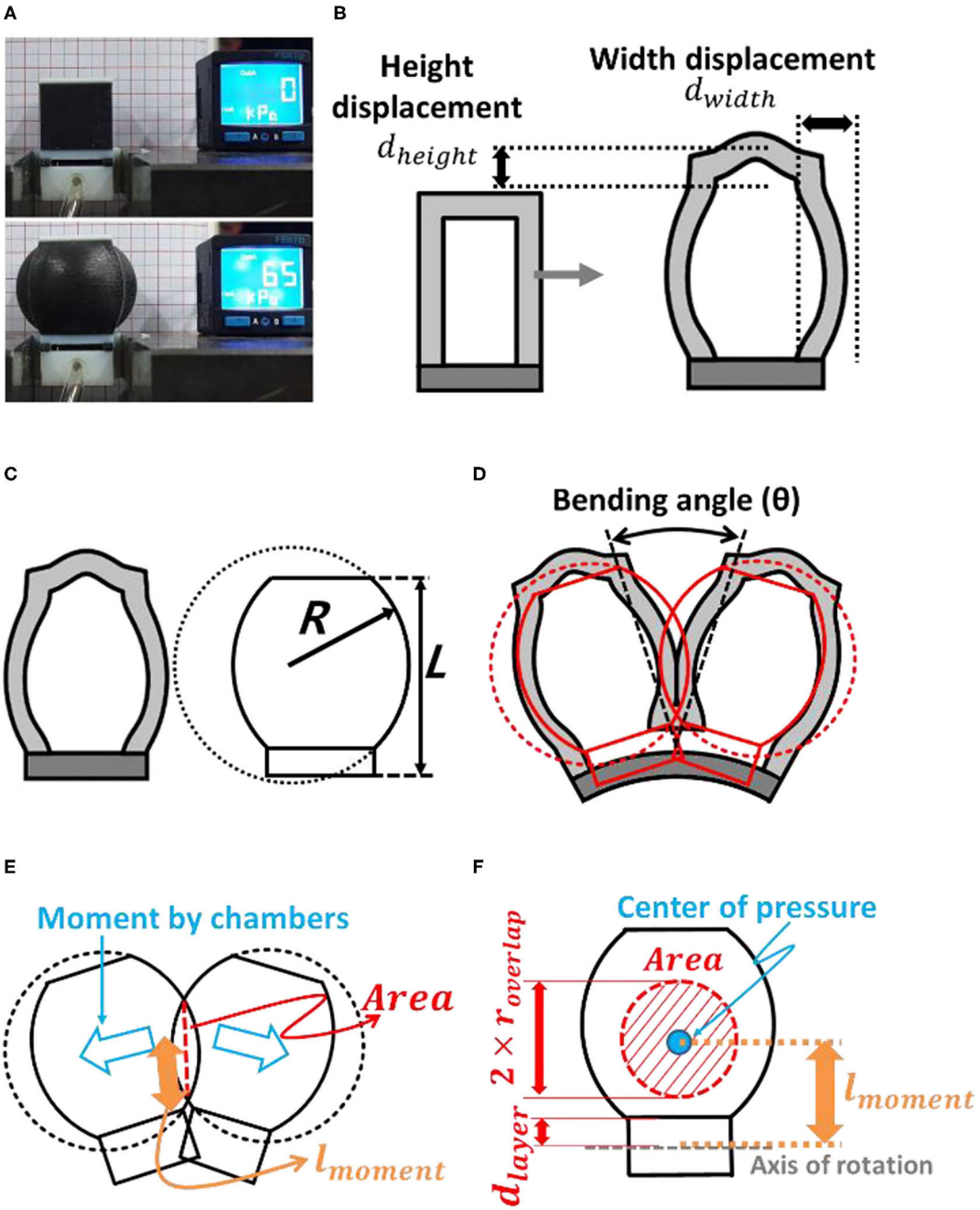


FIGURE 3 | (A) Inflation of a modular design single air chamber block. **(B)** During inflation, the heights of the chamber and its side walls were deformed. **(C)** Geometric approximation of the inflated air chamber wall. **(D)** Schematic diagram of the interaction between two air chambers. **(E)** Contact surface and moment arm. **(F)** Interaction of the face.

approximations. The moment generated by the air chambers ($M_{chamber}$) is described by applied inflating pressure P as in **Equation (3)**.

$$M_{chamber} = Area \cdot P \cdot l_{moment} \tag{3}$$

$$Area = \pi \cdot \left(r_{overlap}\right)^2 \tag{4}$$

$$r_{overlap} = 2 \cdot \sqrt{R^2 - \left(\frac{\left(\frac{L}{2} + d_{layer}\right) \cdot \sin\left(\theta/2\right)}{2} + \frac{\sqrt{-L^2 + 4 \cdot R^2} \cdot \cos\left(\theta/2\right)}{2} \right)^2} \tag{5}$$

$$d_{layer} = \begin{cases} const. = d_{layer,module} \ (for\ modular\ design) \\ \frac{t_{layer}}{2} \ (for\ elastomer\ molding\ design) \end{cases} \tag{6}$$

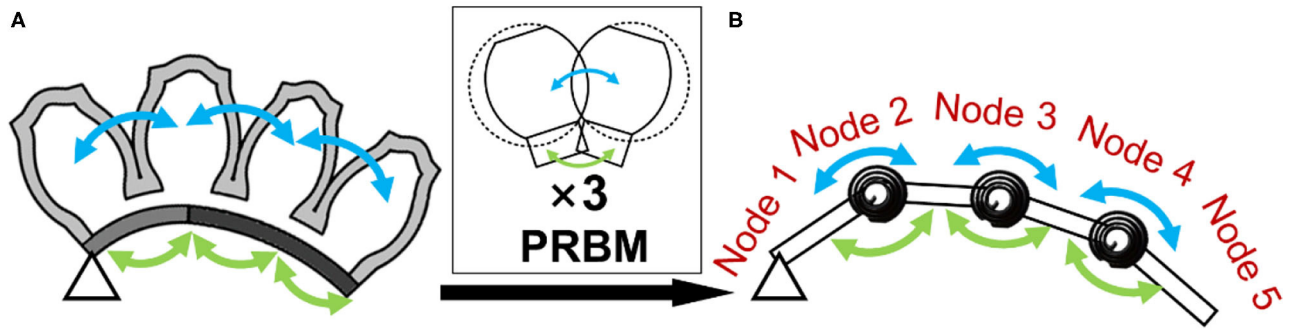


FIGURE 4 | (A) Cross-sectional view of a four-air chamber-long soft bending actuator. **(B)** Pseudo-rigid-body model-based approximation of the soft bending actuator. Arrows indicate the moment associated with the air chambers and the layers, respectively.

$$l_{moment} = \frac{\sqrt{\begin{array}{l} 2 \cdot L \cdot d_{layer} + 2 \cdot R^2 + 2 \cdot d_{layer}^2 \\ + \cos(\theta)\left(L^2 + 2 \cdot L \cdot d_{layer} - 2 \cdot R^2 + 2 \cdot d_{layer}^2\right) \\ - L \cdot \sin(\theta) \cdot \sqrt{4 \cdot R^2 - L^2} - 2 \cdot d_{layer} \cdot \sin(\theta) \cdot \sqrt{4 \cdot R^2 - L^2} \end{array}}}{2} \quad (7)$$

The overlapping area of the interaction of the two air chambers in **Figure 3D** was also assumed to have a circular shape. The radius of the overlapped area ($r_{overlap}$) could be described based on geometric approximations. The moment arm (l_{moment}) could be obtained from the geometric relations derived from the distance between the bottom of the air chamber and the axis of rotation (d_{layer}). The bending of the constraint layer was considered as pure bending. Therefore, the neutral plane of this layer was assumed to be located at its center.

During the bending motion, the constraint layer at the bottom of the structure also generated a moment to return back to the initial state. Based on the Pseudo-Rigid-Body model theory, the constraint layer was assumed to be a non-linear torsional spring. Therefore, the moment generated by the bottom layer (M_{layer}) can be described using the non-linear torsional spring coefficient (k_{layer}) and the bending angle (θ). The non-linear spring coefficient could be obtained from the fitting curve obtained by the three-point bending experiments (**Supplementary Figure 3**). The moment generated by the constraint layer could be described as a function of the bending angle according to **Equation (8)**.

$$M_{layer} = k_{layer}(\theta) \times \theta = f(\theta) \quad (8)$$

Finally, the bending angle between two air chambers at a given actuation pressure is computed by solving the moment equivalent equation between M_{layer} and $M_{chamber}$. Details regarding the relationship between M_{layer} and $M_{chamber}$ will be discussed in chapter 3.

Expand to Multiple Air Chambers

Based on the Pseudo-Rigid-Body model theory, the entire structure of the soft bending actuator could be considered as a superposition of the interactions between two air chambers that are arranged in the longitudinal direction (**Figure 4**). Therefore, the bending posture could be represented by the bending angle of each node of the structure. The bending angles could be determined by solving the moment equivalent equations for each node (**Algorithm 1**). As a result of the simplified analytical model, the posture of the entire structure at any given actuation pressure could be obtained within tens of seconds. For future work, the model could be also applied to other kinds of soft actuators that can be segmentized into a series of interactions between force implying elements (e.g., air chambers) and constraint elements (e.g., constraint layers).

Algorithm 1: Estimating actuated posture of soft bending actuator.

Input: p_{input} // Actuation pressure
$S = [s_1, s_2, \ldots, s_n]$ // Stiffness pattern of the soft gripper with n number of nodes

Output: $G_{shape} = [\theta_1, \theta_2, \ldots, \theta_n]$ // Posture of the soft finger represented by the bending angle of each node

1: **for** $i := 1$ **to** n **Do**
2: $d_{height,i} \leftarrow$ FunctionHeight(p_{input}); // Calculate height displacement of a single air chamber at a given actuation pressure
3: $d_{width,i} \leftarrow$ FunctionWidth(p_{input}); // Calculate width displacement of a single air chamber at a given actuation pressure
4: **Find** θ_i **that**
5: $M_{chamber,i}(\theta_i, p_{input}, d_{height,i}, d_{width,i}) = M_{layer,i}(\theta_i, s_i)$; // Solve the equation about moments generated by the chambers and layer at the i-th node
6: **end for**
7: **return** $G_{shape} = [\theta_1, \theta_2, \ldots, \theta_n]$

STIFFNESS PATTERNING OF CONSTRAINT LAYERS

Analysis of Moment Surfaces and Intersection

The moment generated by the air chambers ($M_{chamber}$) can be represented by the applied pressure (p) and the bending angle (θ), as shown in **Figure 5A**. The region where the magnitude of

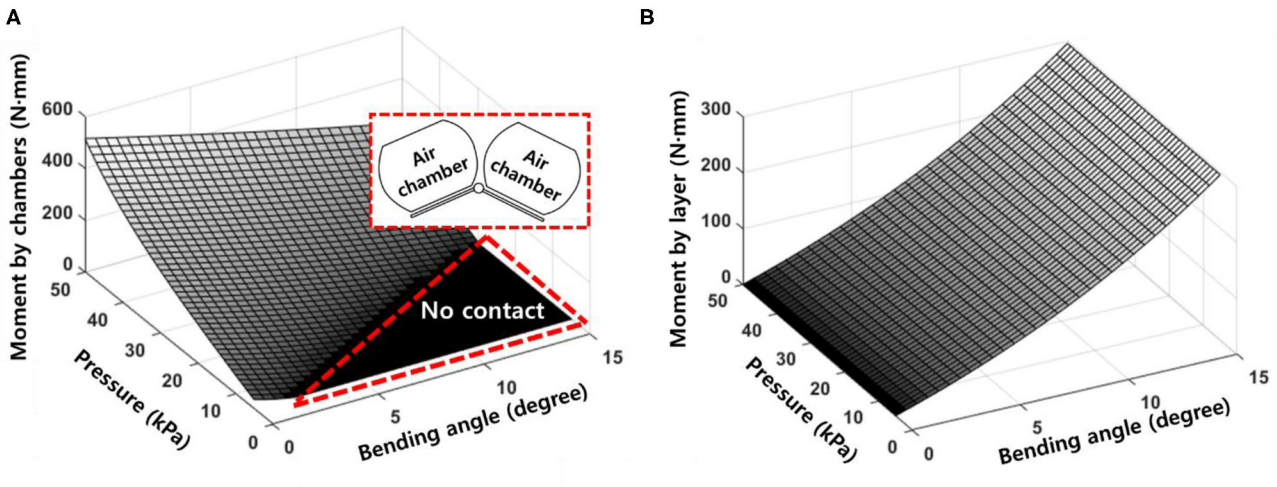


FIGURE 5 | (A) Moment surface generated by the air chambers. The no contact area indicates when the air chambers do not contact each other. **(B)** The moment surface generated by the bottom layer. The moment generated by the bottom layer is only related to the bending angle.

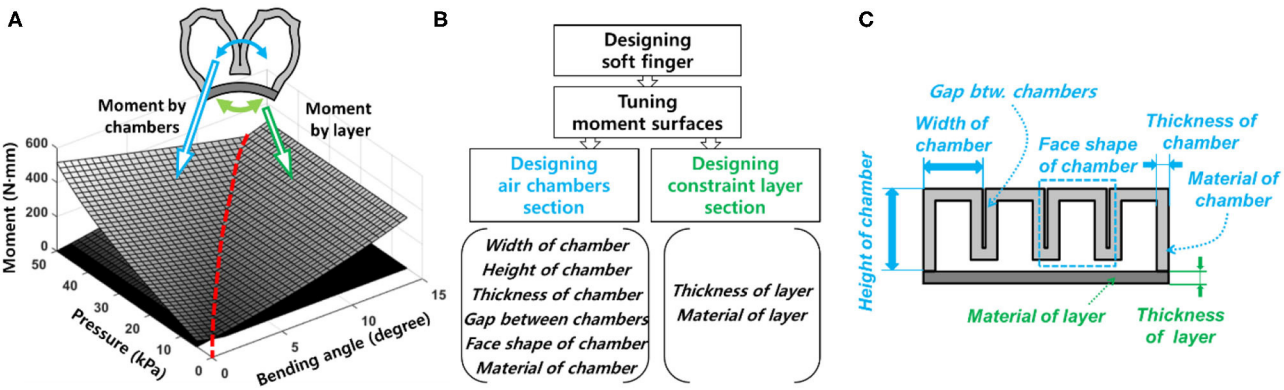


FIGURE 6 | (A) The intersection between the moment surfaces (red dashed line) provides the characteristics regarding bending properties of a soft finger. **(B)** Design process for a soft finger. The bending properties and the contact response can be modified by tuning the moment surfaces through designing each section. **(C)** Design parameters for a soft finger.

the moment is zero represents the instance when the air chambers are inflated but do not contact each other. However, the moment generated by the constraint layer that is plotted in **Figure 5B** is only related to the bending angle (θ).

Figure 6A illustrates the intersection that occurs when the two moment surfaces are plotted together. The intersection characterizes the configuration of the soft bending actuator when there is no contact with the environment. Besides, **Figure 6** implies that the shape of the intersection can be modified by tuning the moment surfaces.

Modifying Moment Surfaces to Design Soft Grippers

Moment surfaces can be tuned through engineering the air chambers and the constraint layer. The air chamber design is affected by the width, cross-sectional shape, thickness of the chamber shell, constituting material, etc. (**Figure 6**). The design parameters are related to both the air chambers' inflation process and the pushing interactions between the air chambers that generate bending moment. On the other hand, the constraint layer has fewer design parameters than the air chambers. Design parameters for the constraint layers mainly relate to the bending stiffness that resists the bending motion generated by the air chamber section.

Designing the air chambers is relatively difficult than designing the constraint layer section. The air chambers non-linearly deformed and push against each other as they get inflated. However, the constraint layer section only experiences bending motion, but neither inflation nor interaction. Therefore, designing the constraint layer was relatively easy than designing the air chamber section, despite the inherent non-linearity of the material. Furthermore, modifying the air chamber section with different designs may require different molds for fabrication; therefore, changing the design parameters of the constraint layers

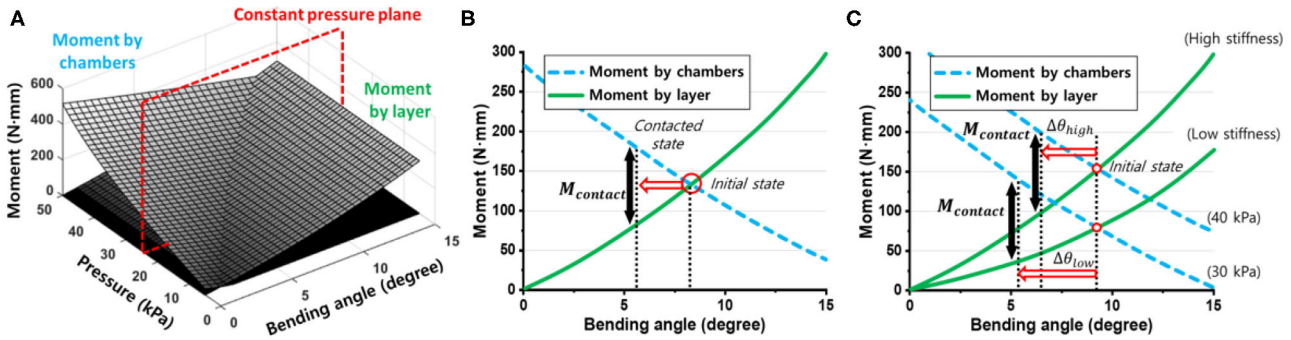


FIGURE 7 | (A) Constant pressure plane at a given actuation pressure for a modular design soft finger. **(B)** The moment generated by a contact force ($M_{contact}$) shifts the equilibrium state from the initial state to the contacted state. **(C)** Comparison between high-stiffness and low-stiffness patterns using modular design soft fingers. When the same contact moment ($M_{contact}$) was applied to soft fingers with different stiffness patterns, soft fingers with high and low stiffness patterns did not bend at θ_{high} and θ_{low}, respectively.

has several advantages over the air chamber section in terms of manufacturing.

Stiffness Patterning for Conformable Grasping

By tuning the constraint layer design, the equilibrium point of each node can be modified, which directly affects the overall configuration of the fingers. In other words, the proper arrangement of nodes, each with different stiffness, in the constraint layer is crucial for establishing conformable pre-grasp postures for specified target objects. In this paper, we will call this arrangement the stiffness patterning of the constraint layer. Most conformable stiffness patterns for predefined objects can be found with the analytical model and the genetic algorithm, provided by MATLAB (MathWorks, Inc.) (**Algorithm 2**). A stiffness pattern that minimizes the mean square error between the shape of the actuated soft gripper and object was selected as the customized pattern design. The outlining shape of the target objects was imported into the algorithm as a fitting curve function and a set coordinates. In this paper, we used three different levels of stiffness patterns: 4, 6, and 8 mm of thickness, as presented in section Model for a Single Interaction between Two Air Chambers of the **Supplementary Material**. The proposed analytical model enables the customization of stiffness patterns within tens of seconds with a personal computer that has general specifications.

Concept of Stiffness Patterning

Stiffness patterning affects the shape of the constant pressure plane of the moment surfaces (**Figure 7A**). Equilibrium between the moment generated by the air chambers and the constraint layer results in a bending angle that represents the initial steady state of the soft finger without any contact force. However, when the soft finger contacts an object, a moment generated by the contact ($M_{contact}$) is applied to the finger. The moment due to contact shifts the steady state from the initial state to the contacted state, as shown in **Figure 7B**.

Algorithm 2: Stiffness pattern customization based on a target object.

Input: p_{input} // Actuation pressure
O_{shape} // Shape of a target object

Output: $S = [s_1, s_2, \ldots, s_m]$ // Stiffness pattern of a soft gripper with m number of nodes to maximize conformability for O_{shape} while actuated at p_{input}

1: **Genetic Algorithm:** Find S_i s.t. minimize $error_{shape}$
2: $S_i \leftarrow$ GetPopulation() // Generate the population for the i-th evolution
3: $G_{shape,i} \leftarrow$ ShapeEstimator(p_{input}, S_i) // Get estimated grasping posture of the soft gripper with the stiffness pattern set, S_i, at actuation pressure p_{input}
4: $error_{shape,i} \leftarrow$ MSE(O_{shape}, $G_{shape,i}$) // Calculate mean square error between O_{shape} and $G_{shape,i}$
5: **end Genetic Algorithm**
6: **return** $S = [s_1, s_2, \ldots, s_m]$

Without any contact, the soft finger with a high stiffness pattern requires a higher actuation pressure than the soft finger with a low stiffness pattern, to achieve the same amount of bending. As shown in **Figure 7C**, when the same amount of the contact moment ($M_{contact}$) is applied to both the high-stiffness and low-stiffness patterned soft fingers, the straightening angles of each stiffness pattern (θ_{high}, θ_{low}) are different. The straightening angle of the soft finger with a high stiffness pattern (θ_{high}) is smaller than that of the soft finger with a low stiffness pattern (θ_{low}).

There are trade-offs between the low-stiffness and high-stiffness patterns in terms of the bending and straightening characteristics. **Table 1** shows a qualitative comparison and the trade-offs between high- and low-stiffness patterns. In the end, it is possible to design actuators that have the same overall appearance by combining different pressures and stiffness. However, because these complementary relationships exist, it is important to properly customize the stiffness pattern according

to the specified conditions, such as maximum applicable pressure and the weight of a gripping object.

FABRICATION PROCESS OF CUSTOMIZED SOFT GRIPPERS

In this chapter, the fabrication process for the customized soft grippers is presented. The aforementioned constraint layer design customization method yields a relatively simple fabrication process for soft grippers.

Single Mold Fabrication Process for Customized Soft Grippers

Fabricating customized soft grippers may present the presupposition of using customized molds for different designs. However, we present a single mold fabrication process for customized soft grippers. The single mold refers to the fabrication of the air chamber section. As mentioned in section Modifying Moment Surfaces to Design Soft Grippers, modifying the stiffness of the constraint layer, to change the actuated posture, is relatively easy compared to changing the air chamber section's design. Therefore, the stiffness of the constraint layer is modified using a varying stiffness pattern mold, then it is bonded to the air chamber section, which stays constant for different designs. The following outlines the fabrication process of a customized soft gripper: first, the thickness tuning plates are stacked inside the base mold of the constraint layer, according to the desired design; then, a pre-cured elastomer is poured into the assembled mold and cured; finally, the fully cured constraint layer is bonded together with the air chamber section (**Figure 8**). The resulting soft gripper achieves customized actuated postures with irregular bending curvatures at different pressures compared to their non-patterned counterparts (**Supplementary Figure 6**).

Also, a stiffness patterning method using modularized blocks is introduced in **Figure 9**. The idea of using modularized soft robotic blocks was introduced in previous research (Lee et al., 2016, 2018). In this paper, we have implemented stiffness patterning into the previous concept of using modular blocks by using flexure blocks with different stiffnesses (**Figure 9C**). The advantage of the modular design is that it simplifies the tuning of the stiffness patterns into a process of disassembly and rejoining of the blocks. Details are described in section Expand to Multiple Air Chambers of the **Supplementary Material**.

TABLE 1 | Qualitative comparison between low stiffness and high stiffness patterned layers.

Low-stiffness patterned layer	High-stiffness patterned layer
Large bending angle per applied pressure	Small bending angle per applied pressure
Small energy consumption for required grasping posture	Large energy consumption for required grasping posture
Weak against external distortion	Strong against external distortion
Weak against sagging by object weight	Strong against sagging by object weight
More adaptive to environment	Less adaptive to environment

EXPERIMENTAL RESULTS FOR CUSTOMIZED SOFT GRIPPER

In this chapter, the experimental results that compare soft grippers with stiffness patterned constraint layers and soft grippers with typical non-patterned homogeneous constraint layers are presented. The experiment consisted of measuring the pulling forces of the soft grippers while grasping target objects. Each gripper had two identical soft fingers mounted in a single plane parallel to the ground (**Figure 10**). The stiffness patterns were optimized based on the target objects.

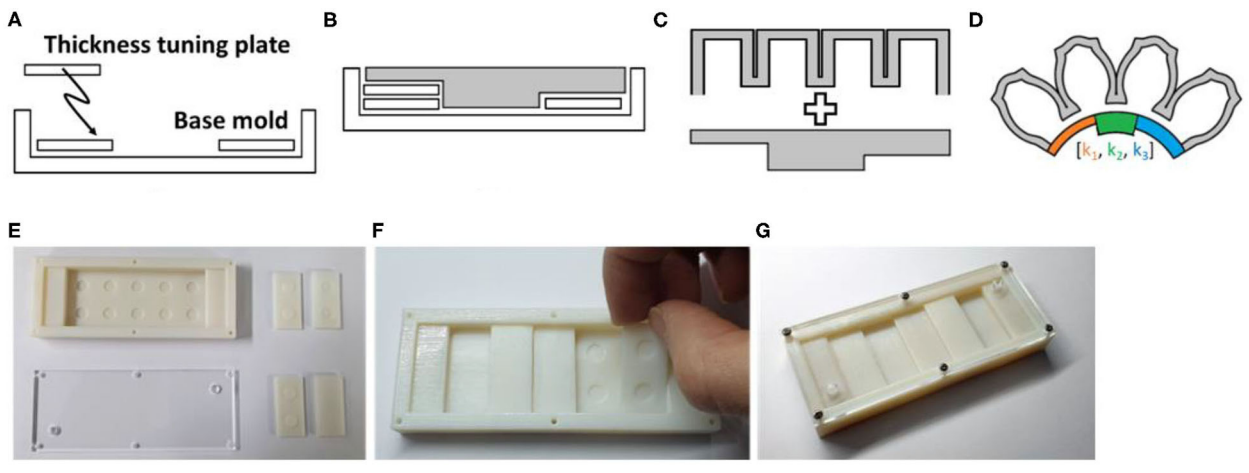


FIGURE 8 | Fabrication process of non-uniform stiffness varying patterned layer. **(A)** The thickness tuning plates are stacked inside the base mold. **(B)** The procured elastomer poured inside the assembled mold. **(C)** The air chamber section and constraint layer are bonded together. **(D)** Soft finger with stiffness patterned layer. **(E)** 3D-printed base mold and thickness tuning plates. **(F)** Thickness tuning plates are stacked inside the base mold. **(G)** Fully assembled constraint layer mold.

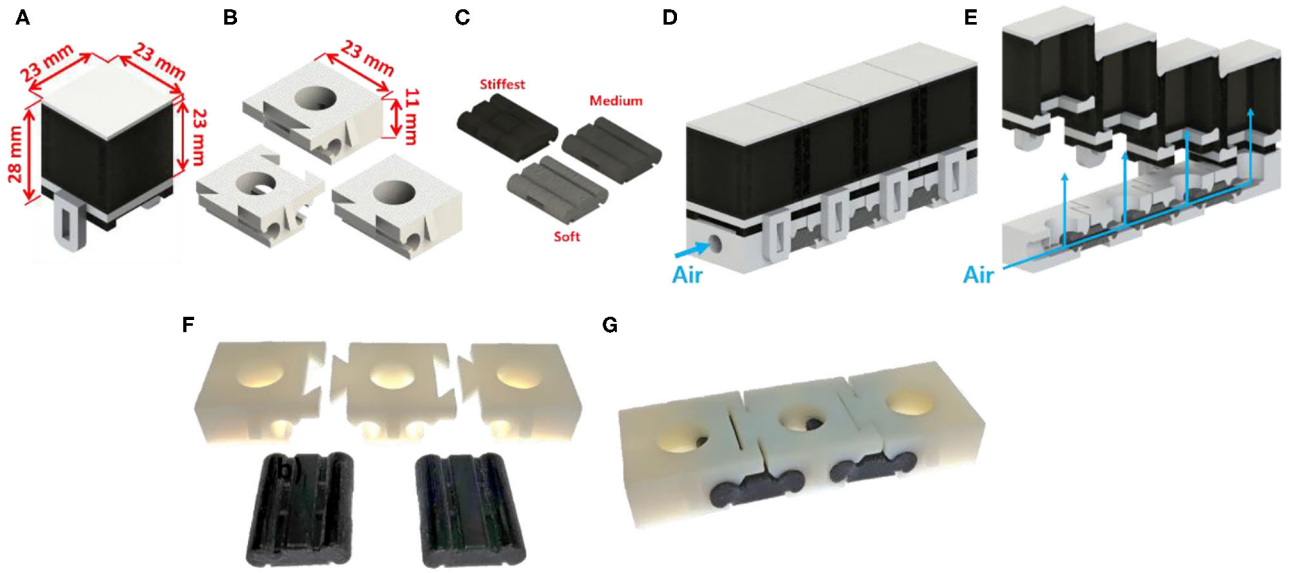


FIGURE 9 | Modular design for customized soft gripper. **(A)** Air chamber block. **(B)** Three kinds of bottom blocks. **(C)** Flexure block. **(D)** Soft finger built by assembling module blocks. **(E)** Cross-section view of the soft finger. Blue arrows indicate actuating pressure. **(F)** Preassembled state of flexure and bottom blocks. **(G)** Assembled bending block.

The actuation pressure was maintained at a constant value throughout the experiment.

Experimental Setups for Object Grasping and Pulling Tests

Soft grippers with stiffness patterned constraint layer and soft grippers with typical non-patterned homogeneous constraint layers were tested. Both kinds of grippers were fabricated using the same material (Dragon Skin 30, Smooth-On Inc.). In addition, the designs of the air chamber section for both the patterned and non-patterned grippers were identical. The dimensions of the soft fingers were the same as those of the soft fingers presented in **Supplementary Material** Chapter 2.

Two kinds of objects were selected as target objects. One was a star-shaped object, and the other was a sphere-shaped object (**Figure 10**). These shapes were chosen because they are very well-known structures, each lying on either end of the extremes: one with no angles and the other with multiple concave acute angles. Both objects were 3D printed using ABS material and fused filament fabrication method. The star-shaped object had a maximum width of 65 mm whereas the sphere-shaped object had a diameter of 60 mm. The objects were connected to a load cell (333FDX, Ktoyo Co. Ltd.) which was mounted onto a linear guide. Force-sensitive resistor sensors (FSR 400, Interlink Electronics, Inc.) were placed on the surface of the objects. The grippers and the objects were designed to be bisymmetric about their center lines. Therefore, the sensors were only attached to a single side of the objects.

Each experiment measured the pulling force and the contact forces between an object and a gripper while the gripper was actuated, and the object was pulled in the outward direction. A load cell and FSR sensors were used to obtain the pulling force and the contact force values, respectively. A draw-wire displacement sensor (CWP-S500R, CALTSensoR) was attached to the same mount where the load cell was positioned. The soft grippers were fabricated based on optimized stiffness pattern designs for each target object.

Customization of Stiffness Pattern for a Given Target Object

The stiffness pattern designs of the constraint layer were optimized for each target-grasping object at preselected actuation pressures. Customization of stiffness patterns was completed within tens of seconds with a personal computer that has general specifications. This rapid speed of calculation was enabled by the proposed analytical model.

The experimental setup in a two-dimensional plane space is shown in **Figure 10C**. The soft fingers of the gripper were placed 60 mm apart. The initial angles of the fingers were rotated 30° in the outward direction from the centerline of the grippers. The centers of the spherical and the star-shaped objects were placed 35 and 55 mm apart from the base of the soft grippers, respectively.

Figure 11 shows the soft grippers with customized stiffness patterns and those without stiffness patterns. The actuated pressure was selected to be 35 kPa because it resulted in better conformability compared to other pressure values. The customized stiffness pattern at this actuating pressure was 4, 6, 8, 8, and 8 mm, from the proximal node to the distal node. However, the constraint layer of the non-patterned soft gripper was also selected to maximize the conformability to the target object. Based on the simulated results, the soft gripper with the

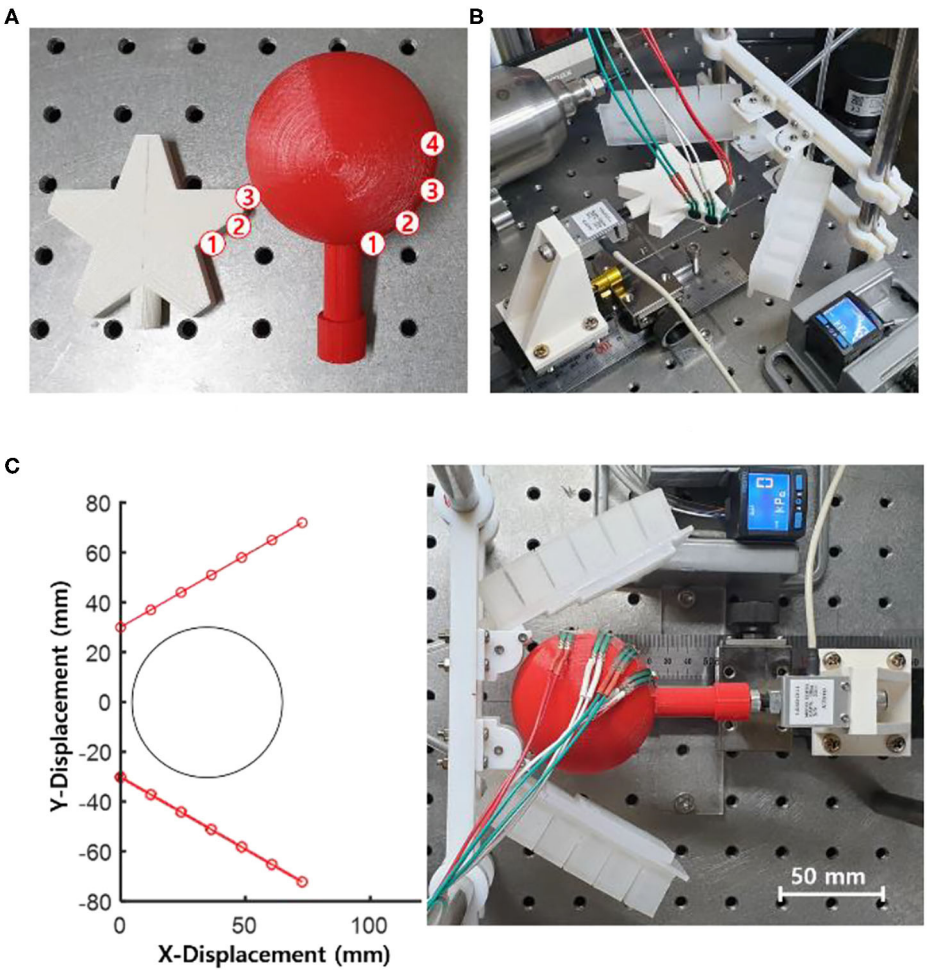


FIGURE 10 | Experimental setup for object gripping and pulling tests. **(A)** Star-shaped and spherical objects were prepared for the pulling tests. FSR sensors were attached to the surfaces of the objects. The circled numbers indicate the locations of the sensors. **(B)** Experimental setup for the star-shaped object. **(C)** View of the experimental setup using the spherical object and the soft gripper with the stiffness patterned soft fingers before actuation.

non-patterned constraint layer with 6-mm thickness had the best conformability among other grippers with 4- and 8-mm layers.

The stiffness pattern of the soft gripper was also optimized to target the star-shaped object. The actuating pressure was selected to be 30 kPa. The optimized stiffness pattern was 6, 8, 8, 4, and 4 mm, from the proximal node to the distal node. Similar to the sphere-shaped object, the thickness of the constraint layer for the non-patterned soft gripper was selected to be 6 mm, which maximized the conformability to the star-shaped object. Both the customized and non-patterned soft grippers were fabricated with the same material (Dragon Skin 30, Smooth-On Inc.). In addition, the designs of the air chamber sections were identical for both grippers.

Comparison Between Customized and Non-patterned Grippers

To evaluate the efficiency and safety, the object pulling force and the contact forces of the soft grippers with customized stiffness patterned constraint layers and the non-patterned were compared. The contact force determines the safety of the object; a lower contact force yields a safer interaction between the object and the gripper. The pulling force determines the load capacity of the gripper; a higher pulling force under the same actuation pressure enables the gripper to grasp heavier objects. First, the object was placed at the predetermined location without actuating the gripper. Then, the soft gripper was actuated to grasp the object. The object was forced out of the gripper by slowly pulling it toward the outward direction. The pulling force of the gripper, the contact forces between the gripper and the object, and the pulling displacement were measured simultaneously. Each experiment was performed five times, and the two results with maximum and minimum values were excluded from the analysis.

Experimental Results for Sphere-Shaped Object

The patterned soft gripper and the non-patterned soft gripper were actuated with the same predetermined actuating pressure, 35 kPa (**Figure 12**). The experimental results regarding the

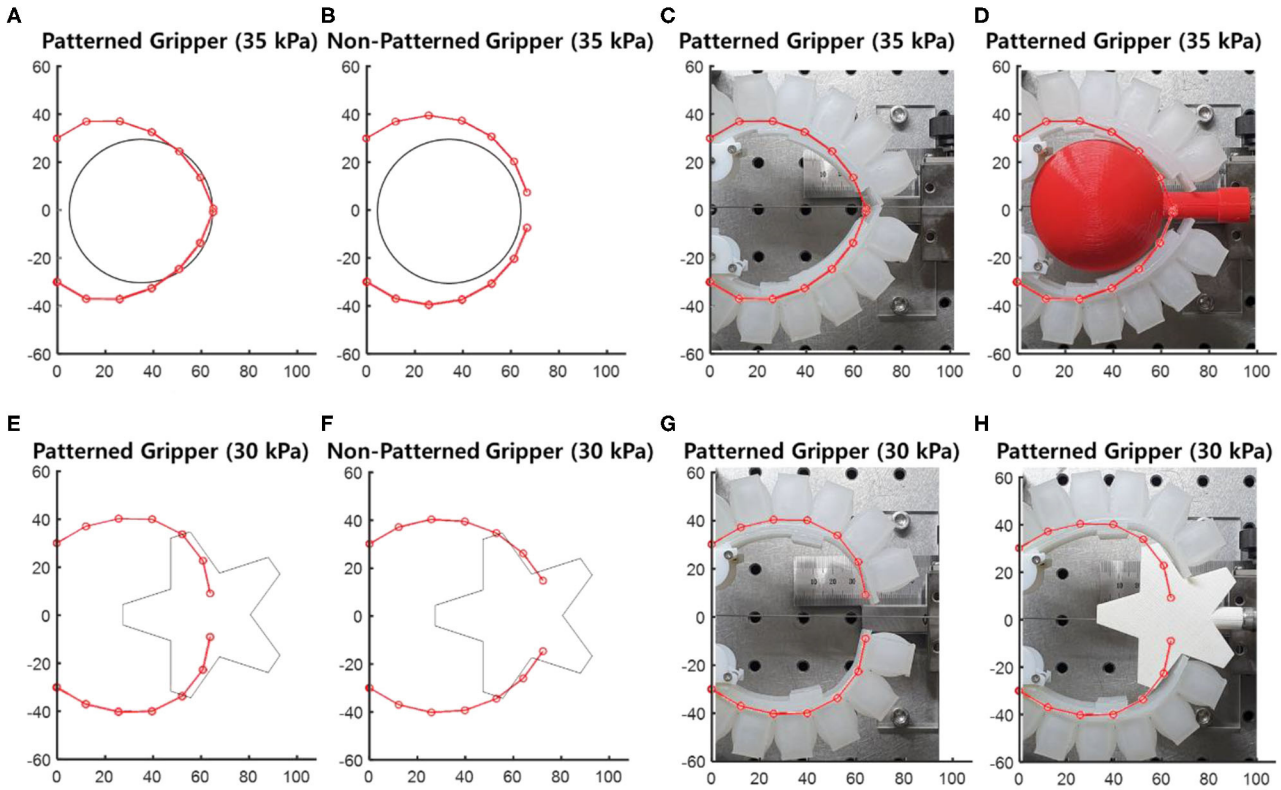


FIGURE 11 | Four soft grippers and spherical object with 60 mm diameter, and star-shaped object. The center of the objects was located at 35, 60 mm from the base of the grippers, respectively. **(A,E)** Customized soft gripper. **(B,F)** Non-patterned soft gripper. **(C,G)** Comparison between experimental and simulated results without the object. **(D,H)** Comparison between the experimental and simulated results with the object.

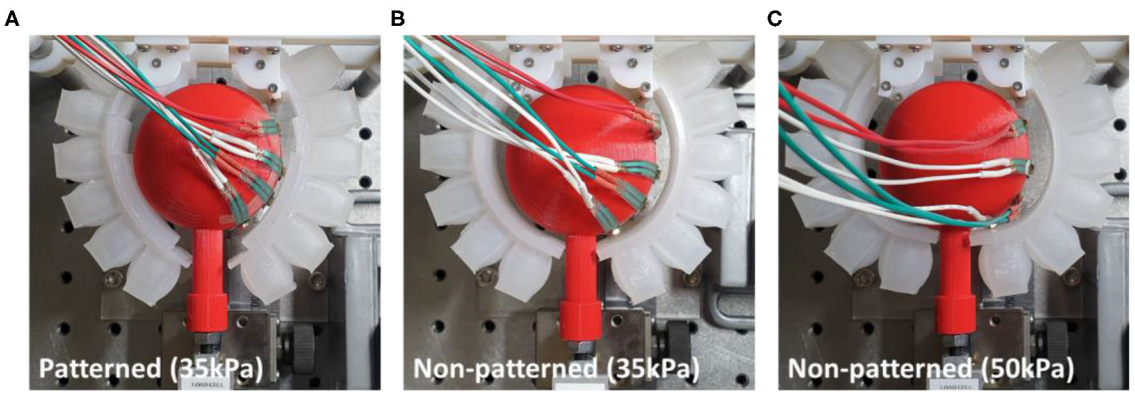


FIGURE 12 | Actuated state of three cases of patterned and non-patterned soft grippers. **(A)** The soft gripper with stiffness patterned constraint layers actuated up to 35 kPa. **(B)** The non-patterned soft gripper with 35 kPa of actuation pressure. **(C)** The non-patterned soft gripper with 50 kPa of actuation pressure. **Figure 13** of the supplementary section illustrates a more detailed version of the experimental results, including the sequence of grasping, and markers that correspond to the model simulations.

pulling forces for the patterned and non-patterned grippers are illustrated in **Figure 13**. The maximum pulling force for the patterned soft gripper is almost three times larger than that obtained for the non-patterned soft gripper under the same actuation pressure (**Table 2**). The non-patterned soft gripper required an actuation pressure of 50 kPa, which is approximately 1.4 times greater compared to that of the patterned soft gripper, which required 35 kPa of actuation pressure.

The contact forces obtained from the four FSR sensors attached on the object's surface are illustrated in **Figure 13**. The

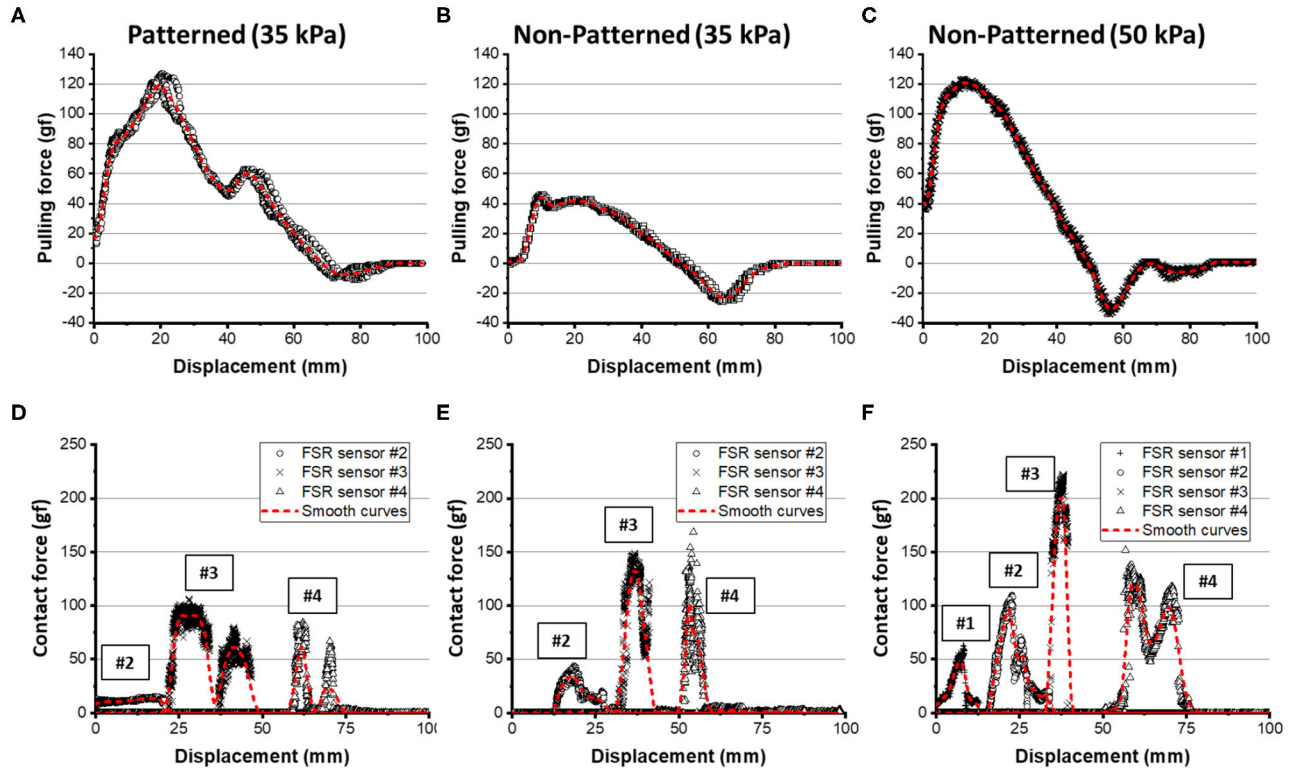


FIGURE 13 | Experimental results for pulling force and contact forces on the spherical object. **(A,D)** The results for the patterned soft gripper actuated at 35 kPa. **(B,E)** The results for the non-patterned soft gripper actuated at 35 kPa. **(C,F)** The results for the non-patterned soft gripper actuated at 50 kPa. Red dashed lines are smoothed curves obtained by using a LOWESS regression with a span of 0.1.

TABLE 2 | The experimental results about grasping sphere-shaped objects.

	Patterned (35 kPa)	Non-patterned (35 kPa)	Non-patterned (50 kPa)
Pulling force (experiments)	119.12	44.03	121.05
FSR #1	0	0	48.22
FSR #2	13.76	34.28	96.16
FSR #3	90.89	132.54	201.67
FSR #4	61.84	102.60	120.45

customized gripper and the non-patterned gripper with 35 kPa of actuation had almost zero forces on FSR sensor #1. The contact force on sensor #2, or the patterned soft gripper, was almost constant. However, the non-patterned soft gripper actuated up to 35 kPa applied a higher contact force on sensor #2. In addition, the non-patterned soft gripper applied almost 1.5 times the force on sensors #3 and #4 than the customized gripper. Moreover, contact durations for both sensors, for the non-patterned gripper, were relatively shorter than those of the customized soft gripper.

The non-patterned soft gripper actuated up to 50 kPa exerted contact force on the FSR sensor #1 from the beginning of the experiments, unlike the previous cases. Sensors #3 and #4, in this case, were applied with almost twice the maximum contact forces than those of the customized soft gripper. Moreover, the position of the sensor #4 is opposite to the lifting direction of the object.

Experimental Results for a Star-Shaped Object

The patterned and non-patterned soft grippers both grasped the star-shaped object with 30 kPa of actuating pressure (**Figure 14**). The patterned soft gripper had about 1.3 times the pulling force than the non-patterned soft gripper (**Figure 15**). The non-patterned soft gripper actuated with 35 kPa of pressure had slightly higher pulling force than the patterned soft gripper with 30 kPa of pressure (**Table 3**).

The contact force applied on the FSR sensor #1 was almost zero, which were out of the measuring range of the sensor, for all three cases. The maximum contact force applied on FSR sensor #2 was almost similar to both for the stiffness patterned and non-patterned soft grippers. However, the contact duration was longer with the stiffness patterned soft gripper. Meanwhile, the non-patterned soft gripper with 35 kPa of actuating pressure exerted a higher contact force on the sensor #3, which was attached to the surface that is opposite to the lifting direction of the object (**Figure 15**).

In summary, the experimental results show that the shape-conformable soft grippers with customized constraint layer stiffness pattern designs had better grasping performance in

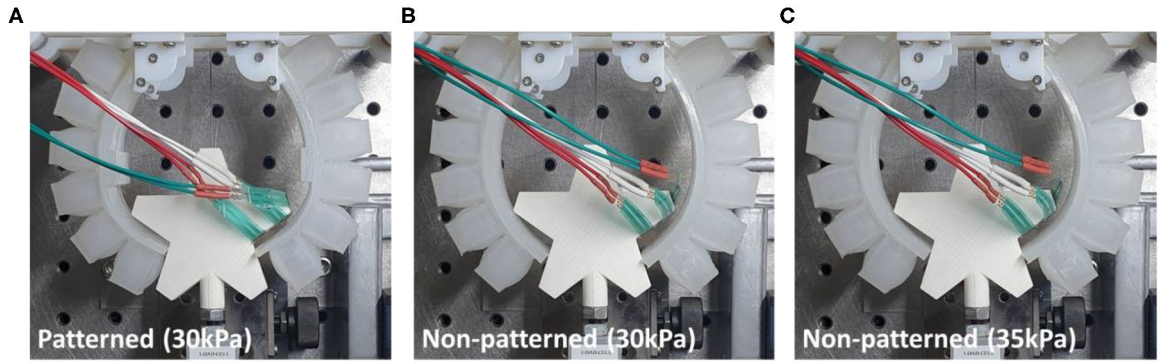


FIGURE 14 | Actuated state of three cases of patterned and non-patterned soft grippers. **(A)** The soft gripper with stiffness patterned constraint layers actuated up to 30 kPa. **(B)** The non-patterned soft gripper with 30 kPa of actuation pressure. **(C)** The non-patterned soft gripper with 35 kPa of actuation pressure. **Figure 13** of the supplementary section illustrates a more detailed version of the experimental results, including the sequence of grasping, and markers that correspond to the model simulations.

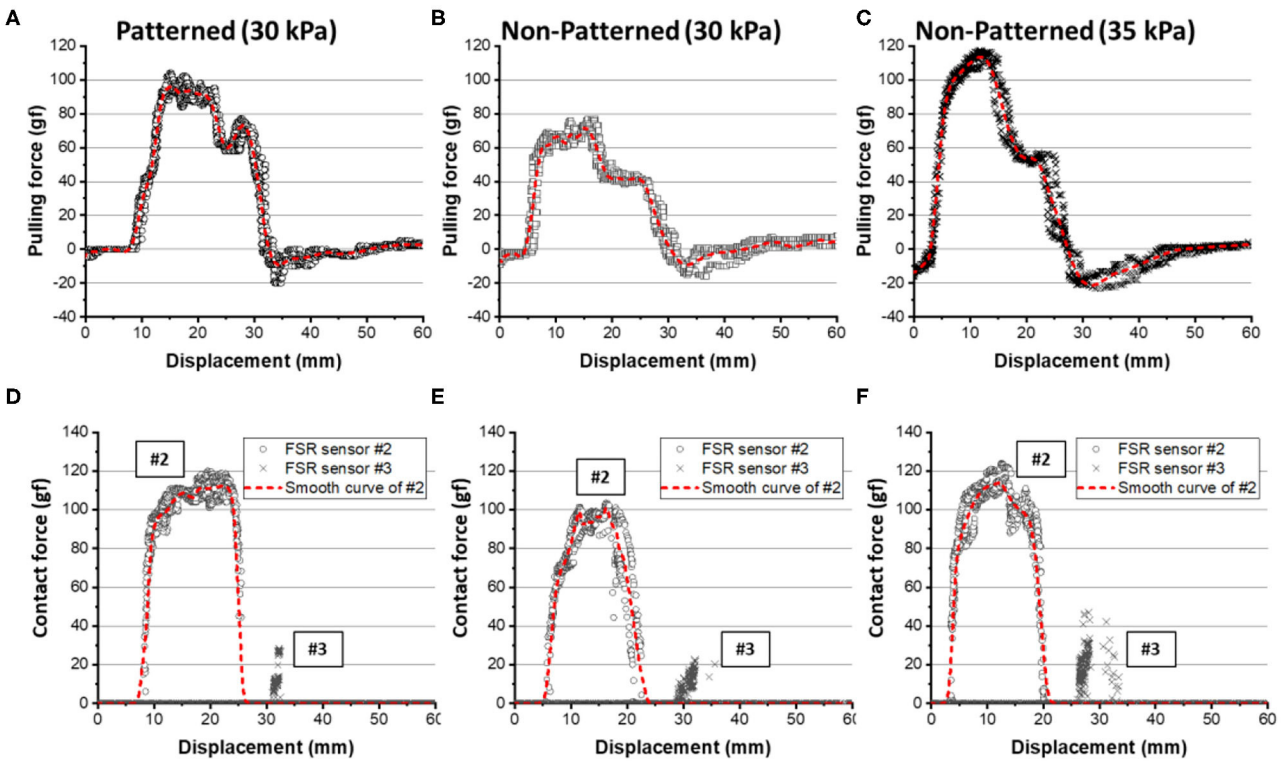


FIGURE 15 | Experimental results for pulling force and contact forces on the star-shaped object. **(A,D)** The results for the patterned soft gripper actuated at 30 kPa. **(B,E)** The results for the non-patterned soft gripper actuated at 30 kPa. **(C,F)** The results for the non-patterned soft gripper actuated at 35 kPa. The red dashed lines are smoothed curves obtained by using a LOWESS regression with a span of 0.1.

terms of object pulling and contact forces. The stiffness patterned soft gripper may exhibit better stability and require lower actuation pressure. Furthermore, the contact forces, which may be related to the integrity of the interaction between the gripper and the object, decreased for the shape-conformable soft gripper due to stiffness patterning of the constraint layers.

CONCLUSION

In this paper, we presented an analytical approach that allows us to estimate and experimentally implement customized postures of soft pneumatic grippers. The model suggested that the moment surfaces generated in the air chamber section and the constraint layer section correspond to the bending behavior of

TABLE 3 | The experimental results about grasping star-shaped objects.

	Patterned (30 kPa)	Non-patterned (30 kPa)	Non-patterned (35 kPa)
Pulling force (experiments)	95.98	71.81	113.44
FSR #1	–	–	–
FSR #2	112.45	101.20	114.01
FSR #3	28.25	31.65	46.95

soft grippers. The computation speed of the model was relatively fast than that of numerical methods, which have been mainly used in existing studies of soft robots. Therefore, it was possible to obtain the optimal converged postures with rapid iterations for given outlining shapes of target objects. Stiffness patterning of the constraint layers of soft grippers was proposed as a facile and powerful methodology to tune the moment surfaces, in conjunction with suitable fabrication methods. Experimental results about the grasping of objects with different shapes showed that the customized grasping posture effectively reduces the contact force and the actuating pressure while maintaining the lifting force.

Future works include enhancing the proposed analytical model and further developing the customization approach. The proposed analytical model requires experimental results regarding the single air chamber inflation test and the three-point bending test. However, the results, obtained from the numerical analysis, such as finite-element analysis, can replace experimental results of the model. Ultimately, the model can be expanded into a hybrid framework that uses the rapid computing speeds of the analytical approach and the preciseness of the numerical method. Implementing topological optimization methodologies into the constraint layer can provide smooth transitions of stiffness profiles that establish grasping postures with better conformability to target objects. Furthermore, the rapid computing speeds of the analytical model can be utilized to generate an abundance of data for machine learning-based optimization processes.

Finally, with our grasping posture customization approach, we hope that soft grippers would take a step closer to the industrial scenes.

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 author.

AUTHOR CONTRIBUTIONS

J-YL built the analytical model, planned and performed the experiments, and also was the main writer of the manuscript. JE performed experiments, supported building the model, and supported the preparation of the manuscript. SY performed experiments and supported the preparation of the manuscript. KC reviewed and guided the proposed analytical model, experimental plans, and the preparation of the manuscript. All authors have contributed to the article and approved the submitted version.

FUNDING

This work was supported by the Technology Innovation Program (10051287, development of fundamental technology of soft robotics for advanced soft gripper), funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea), and was supported by the Korea Institute of Machinery and Materials (NK224G).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2020.00114/full#supplementary-material

REFERENCES

Al Abeach, L. A. T., Nefti-Meziani, S., and Davis, S. (2017). Design of a variable stiffness soft dexterous gripper. *Soft Robot.* 4, 274–284. doi: 10.1089/soro.2016.0044

Amend, J. R., Brown, E. M., Rodenberg, N., Jaeger, H. M., and Lipson, H. (2012). A Positive Pressure Universal Gripper Based on the Jamming of Granular Material. *IEEE Trans. Robot.* 28, 341–350. doi: 10.1109/TRO.2011.2171093

Cheng, N., Amend, J., Farrell, T., Latour, D., Martinez, C., Johansson, J., et al. (2016). Prosthetic jamming terminal device: a case study of untethered soft robotics. *Soft Robot.* 3, 205–212. doi: 10.1089/soro.2016.0017

Cheng, N. G., Gopinath, A., Wang, L., Iagnemma, K., and Hosoi, A. E. (2014). Thermally tunable, self-healing composites for soft robotic applications. *Macromol. Mater. Eng.* 299, 1279–1284. doi: 10.1002/mame.201400017

Dollar, A. M., and Howe, R. D. (2006). A robust compliant grasper via shape deposition manufacturing. *IEEE/ASME Trans. Mechatron.* 11, 154–161. doi: 10.1109/TMECH.2006.871090

Fei, Y., Wang, J., and Pang, W. (2018). A novel fabric-based versatile and stiffness-tunable soft gripper integrating soft pneumatic fingers and wrist. *Soft Robot.* 6, 1–20. doi: 10.1089/soro.2018.0015

Glick, P., Suresh, S. A., Ruffatto, D., Cutkosky, M., Tolley, M. T., and Parness, A. (2018). A soft robotic gripper with gecko-inspired adhesive. *IEEE Robot. Autom. Lett.* 3, 903–910. doi: 10.1109/LRA.2018.2792688

Gorissen, B., Reynaerts, D., Konishi, S., Yoshida, K., Kim, J. W., and De Volder, M. (2017). Elastic inflatable actuators for soft robotic applications. *Adv. Mater.* 29:1604977. doi: 10.1002/adma.201604977

Hao, Y., Wang, T., Xie, Z., Sun, W., Liu, Z., Fang, X., et al. (2018). A eutectic-alloy-infused soft actuator with sensing, tunable degrees of freedom, and stiffness properties. *J. Micromech. Microeng.* 28:024004. doi: 10.1088/1361-6439/aa9d0e

Howell, L. L. (2001). *Compliant Mechanisms*. Hoboken, NJ: John Wiley & Sons.

Hu, W., Li, W., and Alici, G. (2018). 3D printed helical soft pneumatic actuators. In: *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)* (Auckland), 950–955. doi: 10.1109/AIM.2018.8452456

Hughes, J., Culha, U., Giardina, F., Guenther, F., Rosendo, A., and Iida, F. (2016). Soft manipulators and grippers: a review. *Front. Robot. AI* 3:69. doi: 10.3389/frobt.2016.00069

Hurtado, J. F., and Melkote, S. N. (2001). "Effect of conformability on grasp static stability," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)* (Seoul), 1086–1091.

Lee, J. Y., Eom, J., Choi, W. Y., and Cho, K. J. (2018). Soft LEGO: bottom-up design platform for soft robotics. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid), 7513–7520. doi: 10.1109/IROS.2018.8593546

Lee, J. Y., Kim, W. B., Choi, W. Y., and Cho, K. J. (2016). Soft robotic blocks: introducing SoBL, a fast-build modularized design block. *IEEE Robot. Autom. Mag.* 23, 30–41. doi: 10.1109/MRA.2016.2580479

MacCurdy, R., Katzschmann, R., Kim, Y., and Rus, D. (2016). Printable hydraulics: a method for fabricating robots by 3D Co-printing solids and liquids. *Maccurdy*. Available online at: http://dspace.mit.edu/handle/1721.1/103072 (accessed January 9, 2019). doi: 10.1109/ICRA.2016.7487576

Milana, E., Gorissen, B., Voider, M. D., and Reynaerts, D. (2018). Design of a bi-segmented soft actuator with hardware encoded quasi-static inflation sequence. In: *2018 IEEE International Conference on Soft Robotics (RoboSoft)* (Livorno), 108–113. doi: 10.1109/ROBOSOFT.2018.8404905

Mutlu, R., Tawk, C., Alici, G., and Sariyildiz, E. (2017). A 3D printed monolithic soft gripper with adjustable stiffness. In: *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society* (Beijing), 6235–6240. doi: 10.1109/IECON.2017.8217084

Park, W., Seo, S., and Bae, J. (2019). A hybrid gripper with soft material and rigid structures. *IEEE Robot. Autom. Lett.* 4, 65–72. doi: 10.1109/LRA.2018.2878972

Pauly, J., and Midha, A. (2006a). "Pseudo-Rigid-Body Model Chain Algorithm: Part 1 — Introduction and Concept Development," in *ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Philadelphia, PA), 173–181. doi: 10.1115/DETC2006-99460

Pauly, J., and Midha, A. (2006b). "Pseudo-Rigid-Body Model Chain Algorithm: Part 2 — Equivalent Representations for Combined Load Boundary Conditions," in *ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Philadelphia, PA), 183–190. doi: 10.1115/DETC2006-99463

Rus, D., and Tolley, M. T. (2015). Design, fabrication and control of soft robots. *Nature* 521, 467–475. doi: 10.1038/nature14543

Shian, S., Bertoldi, K., and Clarke, D. R. (2015). Dielectric elastomer based "grippers" for soft robotics. *Adv. Mater.* 27, 6814–6819. doi: 10.1002/adma.201503078

Shimoga, K. B., and Goldenberg, A. A. (1992). Soft materials for robotic fingers. In: *Proceedings 1992 IEEE International Conference on Robotics and Automation* (Nice), 1300–1305. doi: 10.1109/ROBOT.1992.220069

Shintake, J., Cacucciolo, V., Floreano, D., and Shea, H. (2018). Soft robotic grippers. *Adv. Mater.* 30:1707035. doi: 10.1002/adma.201707035

Shintake, J., Sonar, H., Piskarev, E., Paik, J., and Floreano, D. (2017). Soft Pneumatic Gelatin Actuator for Edible Robotics. *arXiv:1703.01423 [cs]*. Available online at: http://arxiv.org/abs/1703.01423 (accessed March 14, 2017). doi: 10.1109/IROS.2017.8206525

Su, Y., Fang, Z., Zhu, W., Sun, X., Zhu, Y., Wang, H., et al. (2020). A high-payload proprioceptive hybrid robotic gripper with soft origamic actuators. *IEEE Robot. Autom. Lett.* 5, 3003–3010. doi: 10.1109/LRA.2020.2974438

Terryn, S., Brancart, J., Lefeber, D., Assche, G. V., and Vanderborght, B. (2017). Self-healing soft pneumatic robots. *Sci. Robot.* 2:eaan4268. doi: 10.1126/scirobotics.aan4268

Terryn, S., Mathijssen, G., Brancart, J., Lefeber, D., Assche, G. V., and Vanderborght, B. (2015). Development of a self-healing soft pneumatic actuator: a first concept. *Bioinspir. Biomim.* 10:046007. doi: 10.1088/1748-3190/10/4/046007

Wei, Y., Chen, Y., Ren, T., Chen, Q., Yan, C., Yang, Y., et al. (2016). A novel, variable stiffness robotic gripper based on integrated soft actuating and particle jamming. *Soft Robot.* 3, 134–143. doi: 10.1089/soro.2016.0027

Zhang, H., Kumar, A. S., Chen, F., Fuh, J. Y. H., and Wang, M. Y. (2018). Topology optimized multimaterial soft fingers for applications on grippers, rehabilitation and artificial hands. *IEEE/ASME Trans. Mechatron.* 24, 120–131. doi: 10.1109/TMECH.2018.2874067

Zhou, J., Chen, S., and Wang, Z. (2017). A soft-robotic gripper with enhanced object adaptation and grasping reliability. *IEEE Robot. Autom. Lett.* 2, 2287–2293. doi: 10.1109/LRA.2017.2716445

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.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 05 October 2020
doi: 10.3389/frobt.2020.558027

Model Reference Predictive Adaptive Control for Large-Scale Soft Robots

*Phillip Hyatt, Curtis C. Johnson and Marc D. Killpack**

Robotics and Dynamics Lab, Department of Mechanical Engineering, Brigham Young University, Provo, UT, United States

Past work has shown model predictive control (MPC) to be an effective strategy for controlling continuum joint soft robots using basic lumped-parameter models. However, the inaccuracies of these models often mean that an integral control scheme must be combined with MPC. In this paper we present a novel dynamic model formulation for continuum joint soft robots that is more accurate than previous models yet remains tractable for fast MPC. This model is based on a piecewise constant curvature (PCC) assumption and a relatively new kinematic representation that allows for computationally efficient state prediction. However, due to the difficulty in determining model parameters (e.g., inertias, damping, and spring effects) as well as effects common in continuum joint soft robots (hysteresis, complex pressure dynamics, etc.), we submit that regardless of the model selected, most model-based controllers of continuum joint soft robots would benefit from online model adaptation. Therefore, in this paper we also present a form of adaptive model predictive control based on model reference adaptive control (MRAC). We show that like MRAC, model reference predictive adaptive control (MRPAC) is able to compensate for "parameter mismatch" such as unknown inertia values. Our experiments also show that like MPC, MRPAC is robust to "structure mismatch" such as unmodeled disturbance forces not represented in the form of the adaptive regressor model. Experiments in simulation and hardware show that MRPAC outperforms individual MPC and MRAC.

Keywords: model predictive control, adaptive control, continuum robot, dynamic modeling, MRAC, parameter mismatch, structure mismatch, soft robot

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Cosimo Della Santina, Delft University of Technology, Netherlands
Surya Girinatha Nurzaman, Monash University Malaysia, Malaysia

****Correspondence:***
Marc D. Killpack
marc_killpack@byu.edu

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *01 May 2020*
Accepted: *19 August 2020*
Published: *05 October 2020*

Citation:
Hyatt P, Johnson CC and Killpack MD (2020) Model Reference Predictive Adaptive Control for Large-Scale Soft Robots. Front. Robot. AI 7:558027. doi: 10.3389/frobt.2020.558027

1. INTRODUCTION

Large-scale soft robots hold promise as platforms that are safe for human and delicate environments, and are able to accomplish tasks for which rigid robots are ill-suited. Some tasks for which large-scale soft robots are uniquely capable include whole-arm wiping tasks, reaching through unmodeled cluttered environments, and any task where incidental unmodeled contact is likely or desirable. Continuum joint soft robots have specifically been modeled after examples in nature that excel at these types of tasks (anteaters, octopi, elephants, etc.).

One major obstacle to the use of continuum joint soft robots is the lack of accurate models to enable model-based control. Because flexible continuum joints are not necessarily constrained to rotate about a single well-defined axis, even the kinematic modeling of these robots is relatively complex when compared to rigid robots. The rigid-body dynamics equation that govern the motion of traditional robots are further complicated in pneumatically-actuated continuum joint soft robots by pressure dynamics, energy storage and dissipation in the joints, as well as buckling in some load cases. These factors make the accurate modeling and model-based control of continuum joint soft robots very difficult.

In this work we present a novel dynamic model of a continuum joint robot that can be evaluated fast enough for real-time model predictive control (MPC). This novel dynamic model is in fact a small extension of well-established dynamic model of continuum joint robots based on piecewise constant curvature (PCC) approximations, and a relatively new choice of configuration variables. While only derived for a one joint robot (two degree of freedom), the ideas in this paper are extensible to continuum joint robots with multiple actuated joints.

We also present a form of adaptive MPC that can update our model in order to improve dynamic performance and eliminate steady state error. The adaptive law and much of the theoretical basis for this controller are derived from model reference adaptive control (MRAC) techniques.

The structure of this paper is as follows: section 2 presents the state of the art in continuum soft robot modeling and control, as well as the hardware, models and methods specific to this work; section 3 explains our hypotheses about the new model and proposed controller as well as the design of the experiments performed; section 4 shows the results of the experiments performed and discusses their importance; section 5 discusses the importance of the presented work to the field and provides suggestions for future work.

1.1. Related Work

There is a significant body of work relative to accurately modeling the kinematics and dynamics of soft robots. In Renda et al. (2012) and Thuruthel et al. (2016) the continuum joint is modeled using Cosserat-beam theory. In Kang et al. (2011) and Khalil et al. (2007) methods based on recursive Newton-Euler approaches are used, while in Tatlicioglu et al. (2007) and Godage et al. (2011) dynamic equations are derived using Lagrangian mechanics. In Zheng et al. (2012) and Giri and Walker (2011) lumped parameter models are derived by dividing the continuum joint into a number of finite length sections. The trade-off between accuracy and computational complexity in these methods can be seen by varying the number of the finite sections. The authors of Walker (2013) provide a more comprehensive review of dynamic models for soft and continuum joint robots. Notably, there has also been work to show that learned models can represent soft robot dynamics as in Thuruthel et al. (2017).

In Mochiyama and Suzuki (2002) and Mochiyama and Suzuki (2003) the authors derive the dynamic equations of a continuum arm by integrating over infinitesimal disks and using the method of Lagrange. No assumptions of constant curvature are made. These works are similar to the modeling efforts presented in this paper, the main differences being our choice of generalized coordinates and our assumption of constant curvature. These two differences allow us to derive closed-form analytical expressions for the terms in our equations of motion such as the mass and Coriolis matrices.

In Falkenhahn et al. (2014) and Falkenhahn et al. (2015) the authors derive simpler models based on the PCC assumption. However, they neglect generalized forces caused by rotational inertias. They also model the mass of each PCC section as being concentrated at a point that is fixed in some coordinate frame. In Della Santina et al. (2020b), the authors derive a similar PCC-based model (also neglecting rotational inertia) and then match it to a dynamically equivalent rigid body model. Because the mass and inertia of the joints used in our work are non-negligible, we model the mass as distributed uniformly throughout infinitesimal disks and the center of mass of each joint is calculated analytically assuming uniform density. This approach yields closed-form equations of motion for the continuum joint while more accurately representing the dynamics by including the effects of rotational inertia. This approach also illustrates to a greater extent the effect of dynamic models that include rotational inertia on the performance of highly underdamped systems when compared to the work found in Della Santina et al. (2020b).

Control strategies for soft robots vary from open-loop control such as in Shepherd et al. (2011) and Tolley et al. (2014) to Reinforcement Learning (Zhang et al., 2017) to model predictive control (Best et al., 2016). In Hyatt et al. (2019) and Hyatt and Killpack (2020) the authors demonstrate the performance of MPC on the same joints used for this work. These implementations of MPC used a learned model of the dynamics based on a less-accurate representation of the continuum joint dynamics. The model inaccuracy that resulted in less aggressive control in that work prompted the development of the more accurate model and adaptive control techniques presented in this paper.

Given a dynamic model of the correct form, the nature of soft robots is still such that certain parameters of that model may be difficult to estimate. In terms of adaptive control for soft robots, the most similar to our work is Trumić et al. (2020), where they use a similar formulation of MRAC (although with a different dynamic model and no optimal control law). Although not common in soft robotics, combining MPC and adaptive control is beginning to be an established control strategy where the strengths of MPC are combined with a variety of adaptive control schemes (see Adetola et al., 2009; Kim, 2010; Chowdhary et al., 2013; Bujarbaruah et al., 2018; Pereida and Schoellig, 2018; Abdollahi and Chowdhary, 2019; Zhang and Shi, 2020). The method developed in this paper is a unique form of adaptive MPC that borrows ideas from model reference adaptive control (MRAC) for robot manipulators (Slotine and Li, 1987). Specifically, our work can be considered an extension to the adaptive MPC presented in Terry et al. (2019). The main extensions are an adaptive law formulated specifically for robot manipulators and a regressor based on a more accurate continuum joint dynamic model. These extensions allow greater flexibility to adapt both the parameters and structure of the model.

2. MATERIALS AND METHODS

2.1. Robot Platform Description and Modeling

The robot used for this work is composed of a continuum joint such as the one seen in **Figure 1**. These joints are made of four separate pressure-controlled chambers surrounding a relatively inextensible central cable. The two antagonistically placed pairs

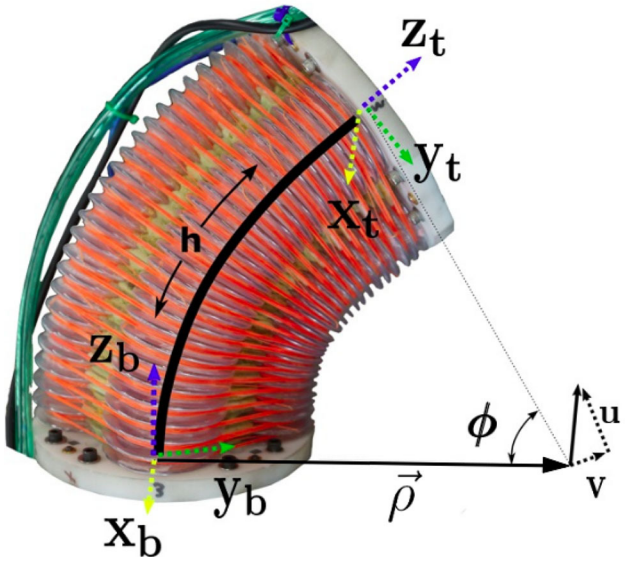


FIGURE 1 | A continuum joint robot such as the one used for this work. The variables $\vec{\rho}$, ϕ, u, v, and h are labeled for reference.

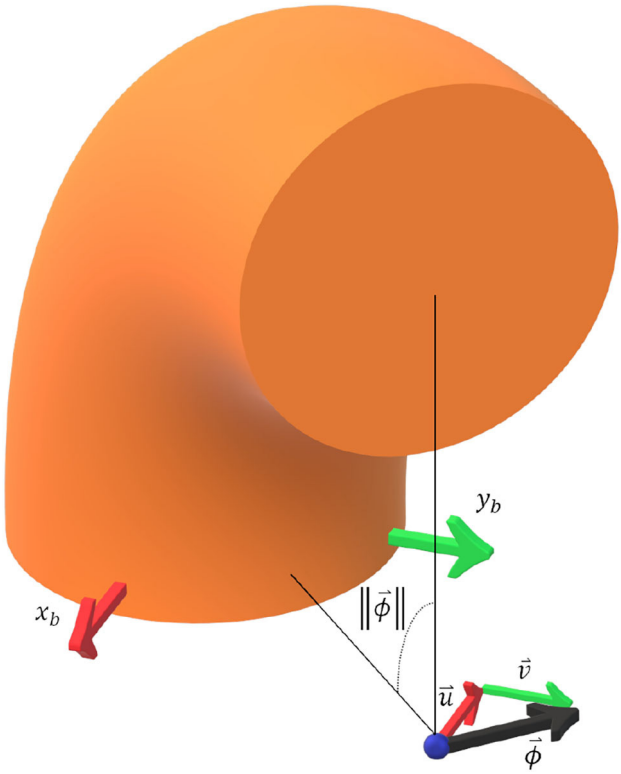


FIGURE 2 | A 3D schematic to illustrate the kinematic relationships used in the presented model derivation. $\vec{\phi}$ is the axis-angle vector which can be decomposed into components parallel with the base frame b. Note that $\vec{u}$ points in the negative x_b direction and that $\vec{v}$ points in the positive y_b direction. The magnitude of the axis-angle vector $\vec{\phi}$ is also the total bend angle.

of pressure chambers allow the joint to bend about two axes. We choose to model the kinematics of this joint using arcs of constant curvature. Each arc, which traces out the path in space occupied by the inextensible spine, can be defined using three variables as described in Allen et al. (2020) (see similar derivation in Della Santina et al., 2020a). These variables are the length of the inextensible spine (h) and two components of the axis-angle vector that describes the rotation from the bottom of the joint to the top. Because the joint cannot twist about the inextensible spine (to which the z axis is tangent) the axis-angle vector consists of only two non-zero variables which we call u and v. These values are labeled in **Figure 1** and correspond to the rotation about the x and y axes, respectively. We assume that the spine is perfectly inextensible so that h in this work becomes a constant kinematic parameter.

First we note some useful kinematic relationships. Because u and v are the non-zero elements of the axis-angle vector we can write

$$\phi = \sqrt{u^2 + v^2} \tag{1}$$

where ϕ is the magnitude of the axis-angle vector $[u, v, 0]^T$, or total bend angle (see **Figure 2**).

Although the joint is modeled as an arc with an arc length of h, we often want to refer to a position at some intermediate point along the arc. We denote an intermediate length along the arc using the variable l where l can take on any value between 0 and h (see **Figure 3**). Note that a frame tangent to the arc at a length l rotates as l is increased, therefore ϕ_l, u_l, and v_l are not constant along the entire arc. However, we note that the vector $\vec{\rho}$ from the base of the joint to the center of curvature is the same for all points along the arc because the center of curvature does not move. At any point l along the arc this vector can be calculated as

$$\vec{\rho} = \frac{l}{\phi_l^2} \begin{bmatrix} v_l \\ -u_l \\ 0 \end{bmatrix}. \tag{2}$$

Because the magnitude of this vector $||\vec{\rho}||$ is the radius of curvature, we may also relate ϕ and l using the arc-length formula

$$\phi_l = \frac{l}{||\vec{\rho}||} \tag{3}$$

We now wish to derive a means by which we can calculate u_l and v_l at any point l along the arc given only l, h, and u and v at the end of the arc. Given a point that lies at a distance l along the arc, we may say using Equation (2)

$$\begin{gathered} \vec{\rho}_l = \vec{\rho}_h \\ \frac{l}{\phi_l^2} \begin{bmatrix} v_l \\ -u_l \\ 0 \end{bmatrix} = \frac{h}{\phi_h^2} \begin{bmatrix} v_h \\ -u_h \\ 0 \end{bmatrix}. \end{gathered} \tag{4}$$

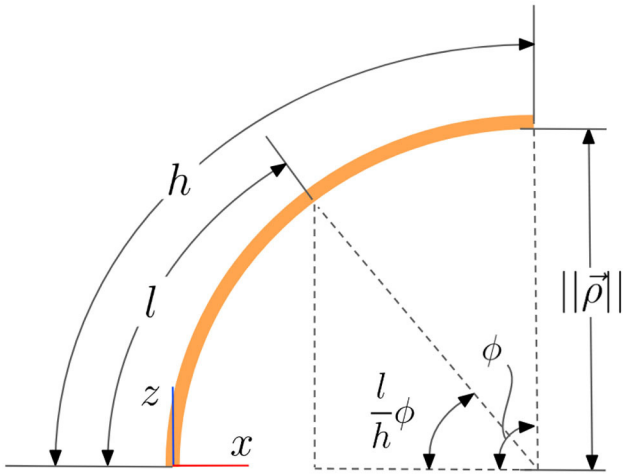


FIGURE 3 | A side view of the kinematic model of a continuum joint showing trigonometric relationships between variables.

Replacing ϕ terms using Equation (3) we obtain

$$\frac{l||\vec{\rho}||^2}{l^2}\begin{bmatrix} v_l \\ -u_l \\ 0 \end{bmatrix} = \frac{h||\vec{\rho}||^2}{h^2}\begin{bmatrix} v_h \\ -u_h \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} v_l \\ u_l \\ 0 \end{bmatrix} = \frac{l}{h}\begin{bmatrix} v_h \\ u_h \\ 0 \end{bmatrix}. \tag{5}$$

Differentiating with respect to time yields the relationship

$$\begin{bmatrix} \dot{v}_l \\ \dot{u}_l \\ 0 \end{bmatrix} = \frac{l}{h}\begin{bmatrix} \dot{v}_h \\ \dot{u}_h \\ 0 \end{bmatrix}. \tag{6}$$

In other words, the generalized coordinates u_l and v_l and their time derivatives vary linearly along the length of the arc. This becomes a very useful property of this kinematic representation when deriving equations of motion.

Using the method of Lagrange, the equations of motion for a system of rigid bodies take the form

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + g(q) = \tau \tag{7}$$

where $M(q)$ is the mass matrix, $C(\dot{q}, q)$ is the Coriolis matrix, $g(q)$ is a vector of gravity torques, q is a vector of the generalized coordinates, and τ is a vector of the generalized torques including friction terms. These matrices are derived using partial derivatives of kinetic and potential energy terms. Since partial derivatives are easily taken using a symbolic mathematics toolbox such as Sympy (see Meurer et al., 2017), the problem of dynamic modeling is reduced to the selection of generalized coordinates and the representation of kinetic and potential energy.

In order to accurately express kinetic and potential energy we choose to model the continuum joint, as many have done before, with an infinite set of infinitesimally small disks. However, the assumption of constant curvature, the choice of generalized coordinates, and current tools in symbolic math libraries allows us to produce analytical expressions for M, C, and g, whereas previous methods have not yielded these closed-form expressions.

We can define the kinetic energy of an infinitesimally thin disc at a length l along the arc as

$$\begin{aligned} T_l &= \frac{1}{2}(\mu \mathrm{d}l)\dot{p}_l^T\dot{p}_l + \frac{1}{2}\omega_l^T I \omega_l \\ &= \frac{1}{2}(\mu \mathrm{d}l)\dot{p}_l^T\dot{p}_l + \frac{1}{2}\omega_l^T \begin{bmatrix} \frac{\mu \mathrm{d}l r^2}{4} & 0 & 0 \\ 0 & \frac{\mu \mathrm{d}l r^2}{4} & 0 \\ 0 & 0 & \frac{\mu \mathrm{d}l r^2}{2} \end{bmatrix} \omega_l \\ &= \frac{1}{2}(\mu \mathrm{d}l)\dot{p}_l^T\dot{p}_l + \frac{1}{2}\left[\frac{\mu \mathrm{d}l r^2}{4}\omega_{l,x}^2 + \frac{\mu \mathrm{d}l r^2}{4}\omega_{l,y}^2 + \frac{\mu \mathrm{d}l r^2}{2}\omega_{l,z}^2\right] \\ &= \frac{\mu}{2}\left[\dot{p}_l^T\dot{p}_l + r^2\left(\frac{1}{4}\omega_{l,x}^2 + \frac{1}{4}\omega_{l,y}^2 + \frac{1}{2}\omega_{l,z}^2\right)\right]\mathrm{d}l \end{aligned} \tag{8}$$

where μ is the linear density of the disc, $\mathrm{d}l$ is some infinitesimal length, $\dot{p}_l$ is the velocity of the center of the disc, ω_l is the angular velocity of the disc expressed in the disc frame, and I is the inertia of the infinitesimally thin disc expressed in the disc frame.

The linear and angular velocity of each disc ($\dot{p}_l$ and ω_l) can be found using a configuration dependent jacobian J (meaning it is a function of joint configuration variables u_l and v_l) that is defined such that

$$\begin{aligned} \begin{bmatrix} \dot{p}_l \\ \omega_l \end{bmatrix} &= J(u_l, v_l, l)\begin{bmatrix} \dot{u}_l \\ \dot{v}_l \end{bmatrix} \\ \begin{bmatrix} \dot{p}_l \\ \omega_l \end{bmatrix} &= \begin{bmatrix} J_{\dot{p}_l}(u_l, v_l, l) \\ J_{\omega_l}(u_l, v_l, l) \end{bmatrix}\begin{bmatrix} \dot{u}_l \\ \dot{v}_l \end{bmatrix} \\ \begin{bmatrix} \dot{p}_l \\ \omega_l \end{bmatrix} &= \begin{bmatrix} J_{\dot{p}_l}(u_l, v_l, l) \\ J_{\omega_l}(u_l, v_l, l) \end{bmatrix}\begin{bmatrix} \dot{u}_h \\ \dot{v}_h \end{bmatrix}\frac{l}{h}. \end{aligned} \tag{9}$$

A definition of this Jacobian for the choice of u and v as generalized coordinates can be found in Allen et al. (2020).

Using this relationship, we see that we can simplify the expression for kinetic energy (Equation 8) by scaling portions of the jacobian. The new inertia-weighted jacobian is defined as

$$J_{\text{weighted}}(u_l, v_l, l) = \begin{bmatrix} \sqrt{\mu}J_{\dot{p}_{l,x}} \\ \sqrt{\mu}J_{\dot{p}_{l,y}} \\ \sqrt{\mu}J_{\dot{p}_{l,z}} \\ \frac{\sqrt{\mu}r}{2}J_{\omega_{l,x}} \\ \frac{\sqrt{\mu}r}{2}J_{\omega_{l,y}} \\ \frac{\sqrt{\mu}r}{\sqrt{2}}J_{\omega_{l,z}} \end{bmatrix} \tag{10}$$

allowing us to rewrite Equation (8) for the kinetic energy of a disc as

$$T_l = \frac{1}{2}\dot{q}^T J_{\text{weighted}}(u_l, v_l, l)^T J_{\text{weighted}}(u_l, v_l, l)\dot{q}\mathrm{d}l. \tag{11}$$

By treating a continuum joint as a series of infinitesimal disks and integrating the kinetic energy of each disc along the length of the arc we can write the total kinetic energy of a joint as

$$T = \frac{1}{2}\dot{q}^T\left[\int_0^h J_{\text{weighted}}(u_l, v_l, l)^T J_{\text{weighted}}(u_l, v_l, l)\mathrm{d}l\right]\dot{q} \tag{12}$$

We note here that the Jacobian can be expressed analytically at every point along the joint as a function of l and the configuration variables u_l and v_l (which are $\frac{l}{h}u_h$ and $\frac{l}{h}v_h$, respectively) thanks to Equation (5). Given this analytical expression for J_{weighted} we can integrate this expression with respect to l over the definite bounds 0 to h to get an analytical expression for $J^T_{\text{weighted}}J_{\text{weighted}}$, which we recognize as the joint space inertia matrix or mass matrix M.

We use the symbolic mathematics library Sympy (see Meurer et al., 2017) to calculate $J^T_{\text{weighted}}J_{\text{weighted}}$, and to integrate this expression analytically between the definite bounds 0 and h in order to obtain $M(q)$. Once $M(q)$ has been obtained symbolically, it is then relatively straightforward to take partial derivatives using Sympy in order to obtain an expression for the Coriolis matrix $C(\dot{q}, q)$ from Equation (7) using the method outlined in Bruno et al. (2010). The resulting coefficients that multiply $\dot{q}$ to calculate the Coriolis matrix are commonly called the Christmases symbols of the first kind.

In order to find the gravity torques (g) we must first find the vector from the joint base to the joint center of mass ($\vec{p}$). By inspection we can see that a joint's center of mass must project down onto the vector $\vec{\rho}$ which is from the center of curvature to the base of the joint, however the vector to the center of mass must also contain some component in the z direction (orthogonal to the plane of the bottom plate of the joint). We find the components of the center of mass vector $\vec{p}$ by again dividing the joint into a series of infinitesimal disks of height dl.

Using the definition of the center of mass assuming the joint has uniform density along its length, the portion of $\vec{p}$ along the z axis is given by

$$\bar{z} = \frac{\int_0^h z\mathrm{d}V}{\int_0^h \mathrm{d}V} \tag{13}$$

Using the trigonometric relationship seen in **Figure 3**, namely

$$z(l) = ||\vec{\rho}||\sin(\frac{l}{h}\phi) \tag{14}$$

as well as the volume formula for an infinitesimally thin disc

$$\mathrm{d}V = \pi r^2 \mathrm{d}l, \tag{15}$$

we can now integrate to find $\bar{z}$:

$$\begin{aligned}
\bar{z} &= \frac{\int_0^h ||\vec{\rho}||\sin(\frac{l}{h}\phi)\pi r^2 \mathrm{d}l}{\int_0^h \pi r^2 \mathrm{d}l} \\
\bar{z} &= \frac{\pi r^2||\vec{\rho}||\int_0^h \sin(\frac{l}{h}\phi)\mathrm{d}l}{\pi r^2 h} \\
\bar{z} &= \frac{-\left[||\vec{\rho}||\frac{h}{\phi}\cos(\frac{l}{h}\phi)\right]_0^h}{h} \\
\bar{z} &= \frac{-||\vec{\rho}||}{\phi}(\cos(\phi) - 1).
\end{aligned} \tag{16}$$

Recognizing that $||\vec{\rho}|| = \frac{h}{\phi}$,

$$\bar{z} = \frac{h}{\phi^2}(1 - \cos(\phi)). \tag{17}$$

In order to find the component of $\vec{p}$ that lies in the plane of u and v we follow a similar procedure. We will use x to represent the portion of $\vec{p}$ that lies along $\vec{\rho}$. Using the trigonometric relationship seen in **Figure 3**, namely

$$x(l) = ||\vec{\rho}||(1 - \cos(\frac{l}{h}\phi)), \tag{18}$$

we can now integrate to find $\bar{x}$:

$$\begin{aligned}
\bar{x} &= \frac{\int_0^h ||\vec{\rho}||(1 - \cos(\frac{l}{h}\phi))\pi r^2 \mathrm{d}l}{\int_0^h \pi r^2 \mathrm{d}l} \\
\bar{x} &= \frac{\pi r^2||\vec{\rho}||\int_0^h (1 - \cos(\frac{l}{h}\phi))\mathrm{d}l}{\pi r^2 h} \\
\bar{x} &= \frac{||\vec{\rho}||\left[l - \frac{h}{\phi}\sin(\frac{l}{h}\phi)\right]_0^h}{h} \\
\bar{x} &= \frac{||\vec{\rho}||}{\phi}(\phi - \sin(\phi)).
\end{aligned} \tag{19}$$

Recognizing that $||\vec{\rho}|| = \frac{h}{\phi}$,

$$\bar{x} = \frac{h}{\phi^2}(\phi - \sin(\phi)). \tag{20}$$

Using the derived equations for $\bar{z}$, $\bar{x}$, and the normalized version of $\vec{\rho}$ we obtain the vector from the base of the joint to the center of mass:

$$\vec{p} = \frac{h}{\phi^2}\begin{bmatrix} (\phi - \sin(\phi))\frac{v}{\phi} \\ (\phi - \sin(\phi))\frac{-u}{\phi} \\ (1 - \cos(\phi)) \end{bmatrix}. \tag{21}$$

The potential energy of the joint due to gravity is simply the dot product of this vector, expressed in the inertial frame, with the gravity vector ($\vec{G}$) expressed in the same frame:

$$V = \vec{p} \cdot \vec{G}. \tag{22}$$

Having calculated the potential energy due to gravity, the gravity torques are calculated simply by taking the negative partial derivative of V with respect to q:

$$g = -\frac{\partial V}{\partial q}. \tag{23}$$

The method above has yielded analytical expressions for M, C, and g with the generalized coordinates u and v. Although complex, these closed-form expressions can be exported from the Sympy library into C code that can be evaluated within microseconds, allowing for real-time model-based control of these continuum joints.

In the absence of applied pressures, the joints used for this paper tend to drive themselves toward an equilibrium position at roughly $u = v = 0$ with slight overshoot and brief oscillation. This spring force could have been modeled as a part of the potential energy, however we choose to model the spring and damping separately from the traditional Lagrangian equations of motion. We approximate the spring forces as a linear spring term $K_{\text{spring}}q$ and friction as a linear viscous damping term $K_d\dot{q}$. Including these terms, the final model used is,

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + g(q) = \tau - K_d\dot{q} - K_{\text{spring}}q \tag{24}$$

2.2. Development of Model Reference Predictive Adaptive Control

In this section we give brief overviews of both MPC and MRAC in order to clarify notation and establish a background for the development of MRPAC. For in-depth explanations of MPC and MRAC we refer the interested reader to Hyatt et al. (2020) and Lavretsky and Wise (2013) respectively.

2.2.1. Model Predictive Control

Any dynamic system may be represented in state variable form as

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}, \mathbf{u})\mathbf{x} + \mathbf{B}(\mathbf{x}, \mathbf{u})\mathbf{u} + \mathbf{w}(\mathbf{x}, \mathbf{u}) \tag{25}$$

where $\mathbf{x}$ is the vector of states, $\mathbf{u}$ is the vector of system inputs, and $\mathbf{w}$ is a vector of offsets or disturbances. By linearizing this system and using any discretization method (Euler, semi-implicit Euler, matrix exponential, etc.) we can create a linear discretized state space model:

$$\mathbf{x_{k+1}} = \mathbf{A_d}\mathbf{x_k} + \mathbf{B_d}\mathbf{u_k} + \mathbf{w_d}. \tag{26}$$

The above equation can be used to forward simulate the states of our system, given initial conditions and inputs. In MPC these discretized dynamic equations are the constraints of our optimization while $\mathbf{x_k}$ and $\mathbf{u_k}$ are the optimization variables. In an MPC solver predicting over a horizon of T time steps, a trajectory optimization may be formulated as:

$$\begin{aligned} J(\mathbf{x}, \mathbf{u}) = \sum_{k=0}^{T} \Big[&(\mathbf{x_{goal}} - \mathbf{x_k})^T Q(\mathbf{x_{goal}} - \mathbf{x_k}) \\ &+ (\mathbf{u_{goal}} - \mathbf{u_k})^T R(\mathbf{u_{goal}} - \mathbf{u_k}) \Big] \end{aligned} \tag{27}$$

$$s.t.$$

$$\mathbf{x_{k+1}} = \mathbf{A_d}\mathbf{x_k} + \mathbf{B_d}\mathbf{u_k} + \mathbf{w_d} \quad \forall \quad k = 0, ..., T-1$$

where J is the objective function value, $\mathbf{x_{goal}}$ and $\mathbf{u_{goal}}$ are the goal states and inputs respectively. Other constraints may easily be added to this formulation to place bounds on inputs or states. By defining a quadratic cost function and enforcing only linear dynamics constraints we have defined a convex optimization problem suitable for solution using a very fast convex solver. We choose to use the state of the art solver OSQP (from Stellato et al., 2017) for our implementation of MPC. In order to lengthen the horizon of MPC and decrease solve times we also use the input parameterization technique presented in Hyatt et al. (2020).

MPC solves the above trajectory optimization for the entire horizon of length T, however only the first input ($\mathbf{u_0}$) is applied to the system. After applying this input, the optimization is solved again using state information that is updated from sensor feedback. The discrete-time model can also be updated with a new linearization centered at the new operating point. This process is repeated with MPC only ever applying the first input, but solving over an entire horizon of value T. The fact that MPC re-solves the trajectory optimization problem with the most current state and model information is what leads to MPC being robust to model error as will be shown hereafter.

2.2.2. Model Reference Adaptive Control

MRAC is a form of adaptive control that seeks to drive a system to behave like a reference system. Because we are interested in controlling continuum joint soft robots we specifically follow the implementation of MRAC outlined in Slotine and Li (1987) which is specific to robot manipulators. In this derivation of MRAC for manipulators, the authors take advantage of several special properties of manipulator dynamics. Firstly, they express the mass matrix, coriolis matrix, and gravity torques as being linear in certain manipulator parameters. Stated mathematically:

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + g(q) = Y(\ddot{q}, \dot{q}, q)a = \tau \tag{28}$$

where $Y(\ddot{q}, \dot{q}, q)$ is the nxp regressor and a is a px1 vector containing the manipulator dynamic parameters which may be unknown or changing over time. In rigid body manipulators it can be shown that a contains the link masses, inertias, and the positions of centers of mass. Using the soft robot continuum joint dynamic model from section 2.1 to derive M, C, and g it can be seen by inspection that all of these terms are linear in the joint mass m, as well as square of the joint radius r^2 and joint height h^2.

In Slotine and Li (1987) the authors present a method by which joint accelerations need not be measured or estimated in order to calculate the regressor. Instead they exploit several properties of manipulator dynamics in order to rewrite the regressor as a function of joint positions (q), joint velocities ($\dot{q}$), reference system velocities ($\dot{q}_{\text{ref}}$), and reference system accelerations ($\ddot{q}_{\text{ref}}$):

$$\tau = Y(q, \dot{q}, \dot{q}_{\text{ref}}, \ddot{q}_{\text{ref}})a. \tag{29}$$

The reference system includes a set of differential equations that describe a system of our choosing with desirable characteristics (such as being a 2nd-order critically damped system with a desired rise time). This is useful in practice because while

accurate measurements or estimates of joint accelerations are hard to obtain, the acceleration of the reference system is a calculated value that we know perfectly.

When using MRAC, we generally do not know the parameter vector a perfectly (especially for soft robots), so we desire to estimate it. We will denote our estimate $\hat{a}$. The adaptive parameter vector $\hat{a}$ is adapted according to the law:

$$\dot{\hat{a}} = -\Gamma^{-1} Y(q, \dot{q}, \dot{q}_{\text{ref}}, \ddot{q}_{\text{ref}})^T s \tag{30}$$

where

$$\begin{aligned} s &= \dot{\tilde{q}} + \Lambda \tilde{q} \\ \dot{\tilde{q}} &= \dot{q} - \dot{q}_{\text{ref}} \\ \tilde{q} &= q - q_{\text{ref}}. \end{aligned} \tag{31}$$

The terms $\tilde{q}$ and $\dot{\tilde{q}}$ are the position and velocity tracking errors with respect to the response of the reference system, and so therefore s is a sort of weighted tracking error term. Γ can be thought of as the learning rate of the adaptive controller.

The final step in manipulator MRAC as explained in Slotine and Li (1987) guarantees that not only parameter error, but also position error will be driven to zero. In order to ensure this, the final control law for MRAC is defined as:

$$\tau = Y(q, \dot{q}, \dot{q}_{\text{ref}}, \ddot{q}_{\text{ref}})^T \hat{a} - K_D s \tag{32}$$

Note that because s is a weighted sum of our position and velocity tracking errors, the matrices K_D and Λ can be thought of as a feedback controller on position error. This feedback term, in addition to the feed-forward term from the adaptive parameters, helps to decrease steady-state position error.

In the above equations, Γ, Λ, and K_D are all tuning parameters used to determine how quickly the adaptive parameters can change and how quickly position error is driven to zero. In general, selecting higher values for the tuning parameters causes the adaptive parameters to change more quickly and the tracking error to decrease more quickly. However, as one may expect, increasing these values to be too high can lead to instability.

Defining $f = M(q)\ddot{q}_{\text{ref}} + C(\dot{q}, q)\dot{q}_{\text{ref}} + g(q) + K_d\dot{q} + K_{\text{spring}}q$, the regressor used for the continuum joint soft robot in this work is of the form:

$$Y(q, \dot{q}, \dot{q}_{\text{ref}}, \ddot{q}_{\text{ref}}) = \begin{bmatrix} \frac{\partial f}{\partial m} & \frac{\partial f}{\partial h^2} & \frac{\partial f}{\partial r^2} & \frac{\partial f}{\partial q} & \frac{\partial f}{\partial \dot{q}} \end{bmatrix}. \tag{33}$$

2.2.3. Model Reference Predictive Adaptive Control

MRPAC combines the strengths of both MPC and MRAC to yield a model-based optimal controller that can adapt its model online, but remains robust to unmodeled disturbances. As with MPC we begin with a model of the system, however this time we explicitly model the error in our model as a torque disturbance term:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{w} + \tau_{\text{disturbance}}. \tag{34}$$

If the error in our model is simply due to incorrect estimates of the manipulator parameters, then we should be able to represent this disturbance exactly using the same regressor as MRAC, namely:

$$\tau_{\text{disturbance}} = -Y(q, \dot{q}, \dot{q}_{\text{ref}}, \ddot{q}_{\text{ref}})\hat{a}. \tag{35}$$

The negative sign is necessary because we adapt the parameters in $\hat{a}$ according to the MRAC adaptation law. MRAC's adaptation law is designed to estimate a torque that, when applied to the system, will "cancel out" the system's dynamics. In MRPAC we want to represent the system's dynamics instead of the torque needed to cancel them out. These two quantities are opposite in sign, hence the negative sign shown here.

It is important to note that in MRPAC we are using the regressor and adaptive parameters to represent our model error, while in MRAC they are used to represent the system dynamics in their entirety. We therefore can not expect $\hat{a}$ to contain the same values for MRAC and MRPAC. In fact, if given a perfect model, $\hat{a}$ should theoretically remain zero for MRPAC. This is because given a perfect model, MRPAC, like MPC, should track perfectly from the beginning and $\tilde{q}$ and $\dot{\tilde{q}}$ will remain zero. As one can see from the adaptive law in Equation (30), as long as these tracking errors remain zero, the adaptive parameters will not change.

Also, it is important to note that Γ and Λ are the only tuning parameters for the estimation of $\hat{a}$ in MRPAC. While in MRAC there is an error term multiplied by K_D in order to ensure that position error is decreased, in MPC the tracking error is decreased by virtue of the optimization that seeks to minimize error.

In order to make a fair comparison between MRAC and MRPAC we use the same regressor for both controllers.

3. DESCRIPTION OF EXPERIMENTS

Adaptive control techniques are useful in the case where we do not know a complete and accurate model of our system a priori. After all, if we did have a complete and accurate model then we could perfectly predict the behavior of our system for model-based control techniques. We will classify all modeling error into two categories: parameter mismatch and structure mismatch. Parameter mismatch correspond to terms, physical phenomena, or parameters in our model that we are accounting for, but whose values are uncertain or unknown. For example inertias, damping coefficients, and spring coefficients may be parameter mismatch. Structure mismatch in our model corresponds to phenomena that occur in the real system, but are not represented in our model for whatever reason. If we assume all spring and damping elements in our system are linear while they are in fact non-linear, then we do not have the ability to represent the non-linear effect of the spring and this non-linear effect is structure mismatch.

3.1. Simulation Experiments

In the simulation portion of the experiments, a simulation is created using the model outlined in section 2.1 and this simulated system is controlled using three different controllers. The goal of each controller is to drive the system to follow a reference trajectory generated by a reference system. The three controllers

implemented are MPC, MRAC, and the MRPAC algorithm detailed in section 2.2.3.

The reference system used for these experiments can be thought of as two uncoupled, critically-damped mass-spring-damper systems each modeled by the equation:

$$m\ddot{x} + b\dot{x} + k(x - r) = 0. \tag{36}$$

The masses (of mass m) are driven by the springs to the reference positions (r) and the damping coefficient (b) is always chosen such that the system is critically damped ($b = \sqrt{4mk}$). The rise time of the reference system can be altered by varying the spring constant (k). We choose a rise time such that the system has settled to steady state within about 1 s.

As mentioned in the adaptive control literature, model parameter estimation and adaptive control schemes require sufficient "excitation" in order to converge or to adapt. We provide this excitation by changing the reference positions (r) of our system every 2 s. Reference positions are drawn from a uniform distribution bounded above and below by $-\frac{\pi}{2\sqrt{2}}$ and $\frac{\pi}{2\sqrt{2}}$. These bounds are chosen so that the resulting total bend angle ($\phi = \sqrt{u^2 + v^2}$) is never greater than $\frac{\pi}{2}$ radians.

3.1.1. Case 1: Perfect Regressor (Parameter Mismatch)

The first experiment performed is designed to show the performance of all three controllers in the case where the regressor can fully describe the dynamics of the system (e.g., there is no structure mismatch). The hypothesis to be tested is that given a perfect regressor (speaking in terms of form and not initial values), both MRAC and MRPAC should be able to compensate for the system's dynamics perfectly and should drive the system to follow the reference trajectory exactly. For MPC, since it cannot adapt its model we expect that increasing model error (but not adding additional unmodeled terms) will lead to increasing tracking error.

To test this hypothesis we control the same system using the three controllers outlined in section 2.2 (i.e., MPC, MRAC, and MRPAC) and provide MRAC and MRPAC each with the same regressor. Because MPC and MRPAC require a discretized model, we introduce model error in order to see the effect on their performance. The method used for introducing model error is to make our estimates of h, m, K_{spring}, K_{damper} a scalar multiple of their simulated value. Because MRAC does not utilize a model apart from the regressor, it is invariant to model error. All adaptive parameters for MRAC and MRPAC are initialized at zero.

Each controller is run in simulation for 5 min of "excitation" (new reference commands every 2 s) in order to allow the adaptive parameters to settle. After 5 min of "excitation" the performance of each controller is evaluated during one additional minute. Because MPC is not adapting at all, this excitation period makes no difference in its performance. The integral of the position error during the 1 min evaluation is shown in **Figure 4** as a function of the model error. As an example, the joint trajectories during the evaluation using a modeling error scalar of 1.5 are shown in **Figure 5**. Note that the green line cannot be seen because it is directly beneath the blue and red lines.

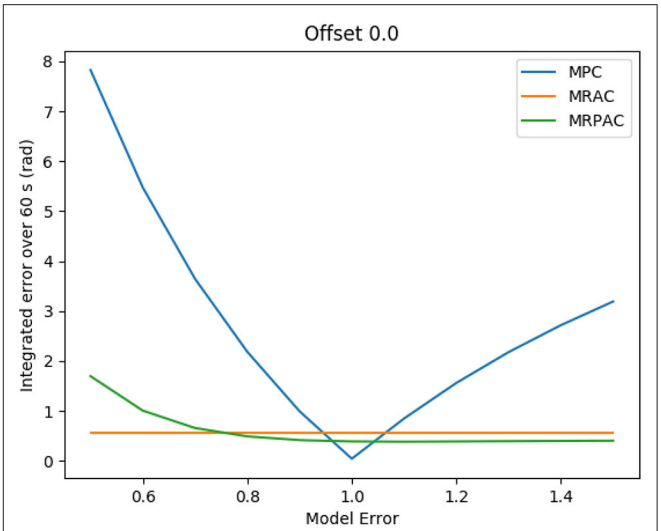


FIGURE 4 | Tracking error sensitivity to model error for all three controllers in simulation.

3.1.2. Case 2: Imperfect Regressor (Structure Mismatch)

The second experiment performed is designed to show the performance of all three controllers in the case where the regressor cannot fully describe the dynamics of the system. The hypothesis to be tested is that neither MRAC nor MRPAC should be able to adapt for the system's dynamics perfectly given an imperfect regressor, and both should therefore struggle to drive the system to follow the reference trajectory exactly. However, because MPC has been shown to be robust to modeling error, both MPC and MRPAC should be more robust to the unmodeled forces that affect the dynamics.

To test this hypothesis, instead of simulating a system in which a spring force drives the joint toward the zero configuration, we simulate a system in which the spring force drives the joint toward a non-zero configuration. This is a phenomenon observed in real soft robot hardware because of slight inconsistencies in the manufacturing of the plastic bellows. This offset spring force can be thought of as a constant torque that is applied to the joint in one direction. Because the regressor does not contain any terms that correspond to a constant torque offset, this force cannot be represented by the regressor and therefore constitutes a "structure mismatch." While we *do* actually know about this constant offset and likely would include a constant term in the regressor, we anticipate that there will be forces which we do not know about or whose form is unknown for any real soft robot. This simple experiment allows us to see the potential effects of these completely unmodeled forces.

In order to see the sensitivity of each controller to this unmodeled force that cannot be represented with the regressor, we vary the spring force equilibrium offset between $u = v = 0.05$

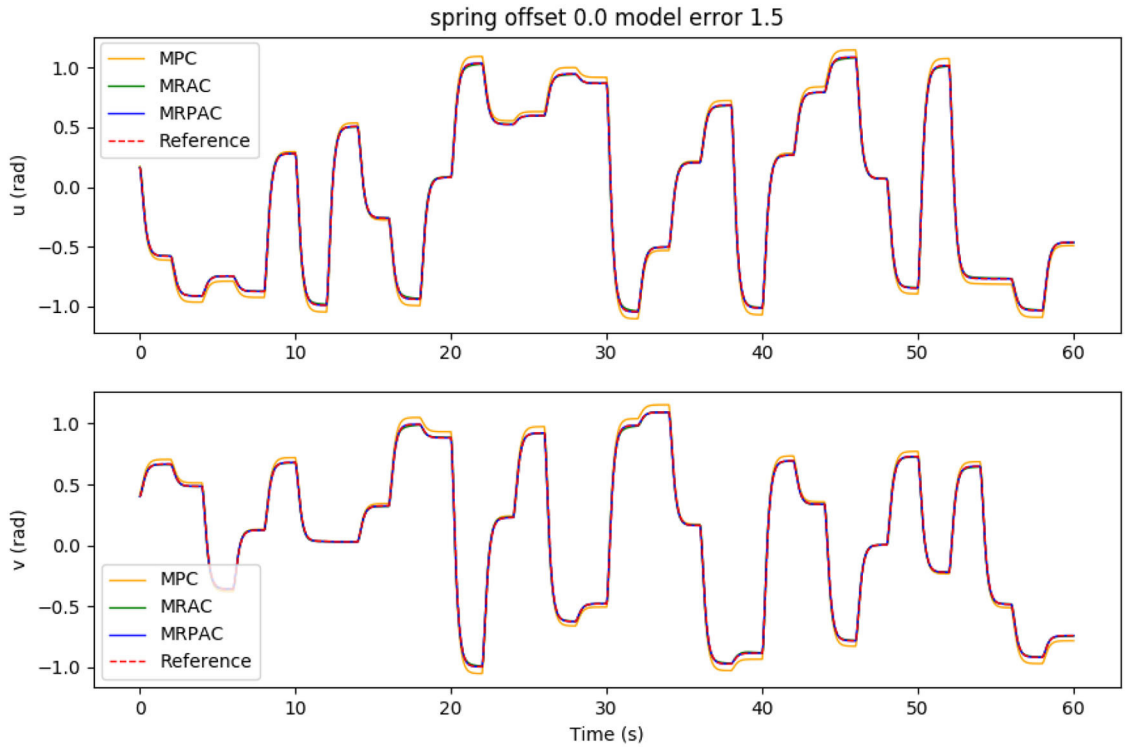


FIGURE 5 | Joint trajectory tracking using all three controllers in simulation. This is for the case of errors in the model parameters used for MPC and MRPAC. Note that the reference trajectory corresponds to q_{ref}, the position states of our dynamic reference system defined in Equation (36). Also note that the performance of MRAC and MRPAC is indistinguishable.

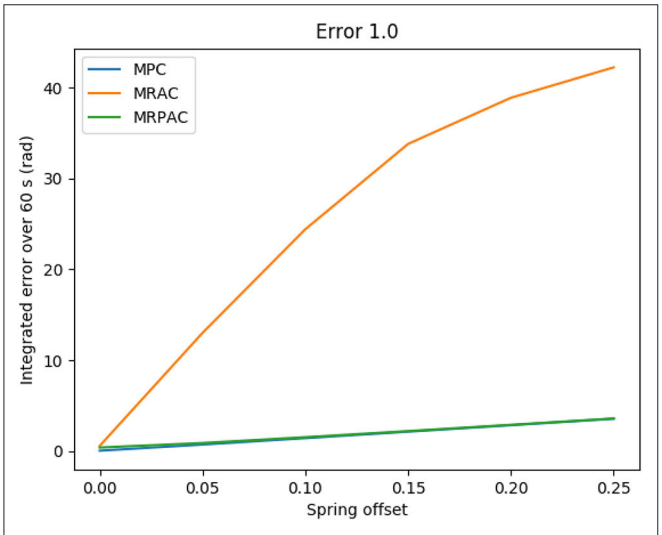


FIGURE 6 | Simulated tracking error sensitivity to unmodeled offset forces/torques (structure mismatch) if the rest of the model is perfect.

rad and $u = v = 0.25$ rad. We do this for each setting of % model error tested in the first experiment, yielding a surface of tracking error that is a function of both a scaled model error (parameter mismatch) as well as an unmodeled constant torque (structure mismatch).

Again, after 5 min of "excitation" the performance of each controller is evaluated during one additional minute. The integrated position error during the evaluation minute is shown in **Figure 6** as a function of the model error. As an example, the joint trajectories during the minute evaluation using a spring offset of $u = v = 0.25$ are seen in **Figure 7**.

3.2. Hardware Experiments

In order to validate both simulations, we implement the same three controllers (MPC, MRAC, and MRPAC) on the soft continuum joint shown in **Figure 1** and compare their performance.

The soft continuum joint used for this experiment is actuated by four plastic bellows, each of which can be controlled independently. A pressure difference in each of the bellows causes a rotation about one or both of the joint's axes. The angle about each of these axes (denoted u and v in **Figure 1**) is the robot's position and are the variables that we attempt to control. We expect this hardware platform to illustrate the sensitivity of each controller to both parameter mismatch and structure mismatch.

Both sources of error are present in hardware. Because no system identification was performed previously, the aforementioned model parameters such as h, m, r, K_{spring}, and

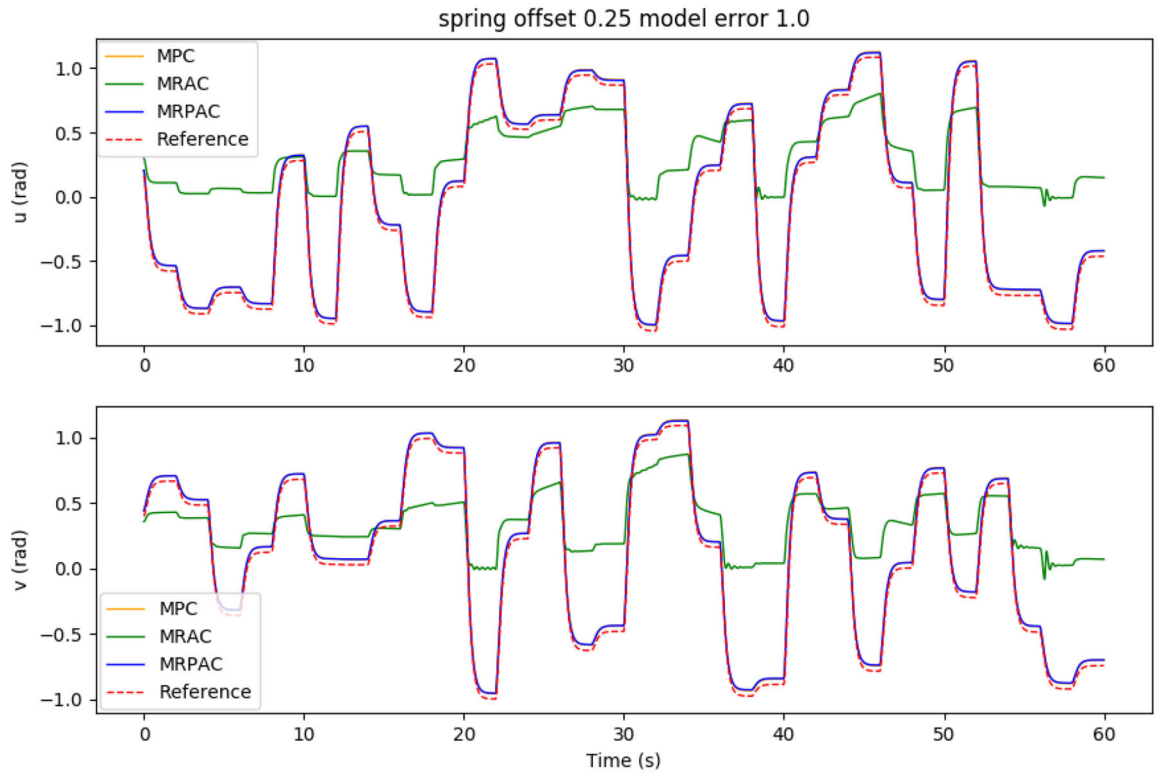


FIGURE 7 | Simulated joint trajectory tracking of all three controllers with a perfect model besides an unmodeled offset torque. Note that the reference trajectory corresponds to q_{ref}, the position states of our dynamic reference system defined in Equation (36). Also note that the performance of MPC and MRPAC is indistinguishable.

K_d are not known perfectly. Additionally, the continuum joint exhibits unknown non-linear behavior near the extremes of its range of motion or in certain directions, where its stiffness or damping vary non-linearly with respect to u and v. In addition to the non-linear effects, we observe the effects of various offset forces in the plastic bellows used to actuate the joint. For example, even with equal pressures in each of the four bellows, the continuum joint remains slightly bent, indicating the presence of some constant unmodeled forces. For our simulations (see section 3.1.2) we represented this as a constant spring offset, but the actual source of this offset is unknown. In order to allow the adaptive control methods to compensate for this constant offset force we add to the regressor an identity matrix. This identity matrix means that the adaptive parameters that multiply it will be mapped directly to generalized torques in the dynamic model.

We track the orientation of a frame on top of the joint relative to a frame below the joint in order to estimate the state of the joint in real-time. We reuse the same reference trajectory from the simulation with one minor change: the command changes every 5 s instead of every two. This was adjusted in an attempt to be conservative with experimental hardware and software while still validating the performance of each controller.

As in the simulation experiments, we excite the system with the same 150 commands used in simulation (12.5 min) before evaluating each of the controllers for the last 30 commands (2.5 min). The joint trajectories for this evaluation period are shown in **Figure 9**.

4. RESULTS

4.1. Simulation Experiments

4.1.1. Case 1: Perfect Regressor (Parameter Mismatch)

The first experiment was designed to see the sensitivity of each controller to parameter mismatch, or model error where at least the form of the model is known. The results of this experiment can be seen in **Figure 4**. An example of the joint angle trajectories achieved by each controller is shown in **Figure 5**. As expected, MRAC is unaffected by this kind of model error because MRAC was initialized with all parameters equal to zero and adapted the parameters to their values based on the MRAC adaptation law. We see that given a correct form of the model, MRAC is able to find a very good model and track the reference trajectory with very little error. When MPC is given a perfect model, we see that it performs better than either MRAC or MRPAC, reducing tracking error to near zero over the entire evaluation period of 60 s. However, we see that it is the most sensitive to

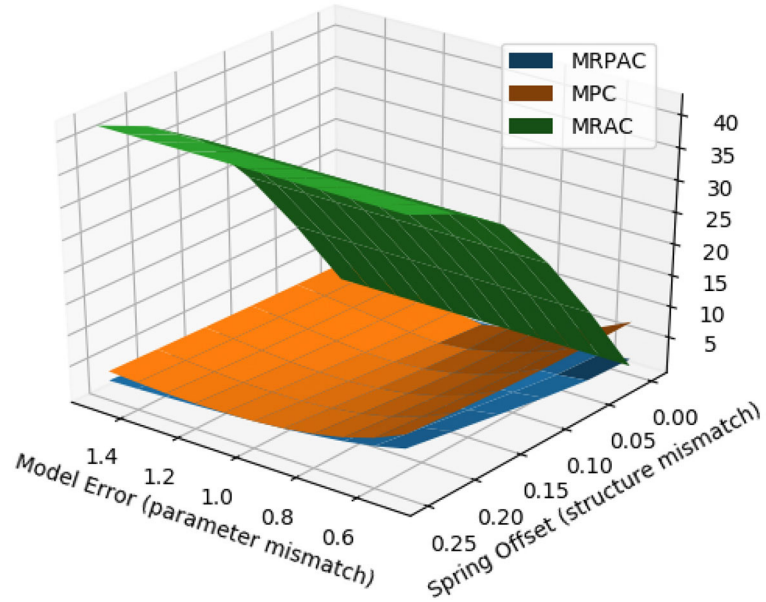


FIGURE 8 | Simulated joint trajectory tracking error as a function of both model parameter error (parameter mismatch) and a spring offset error (structure mismatch).

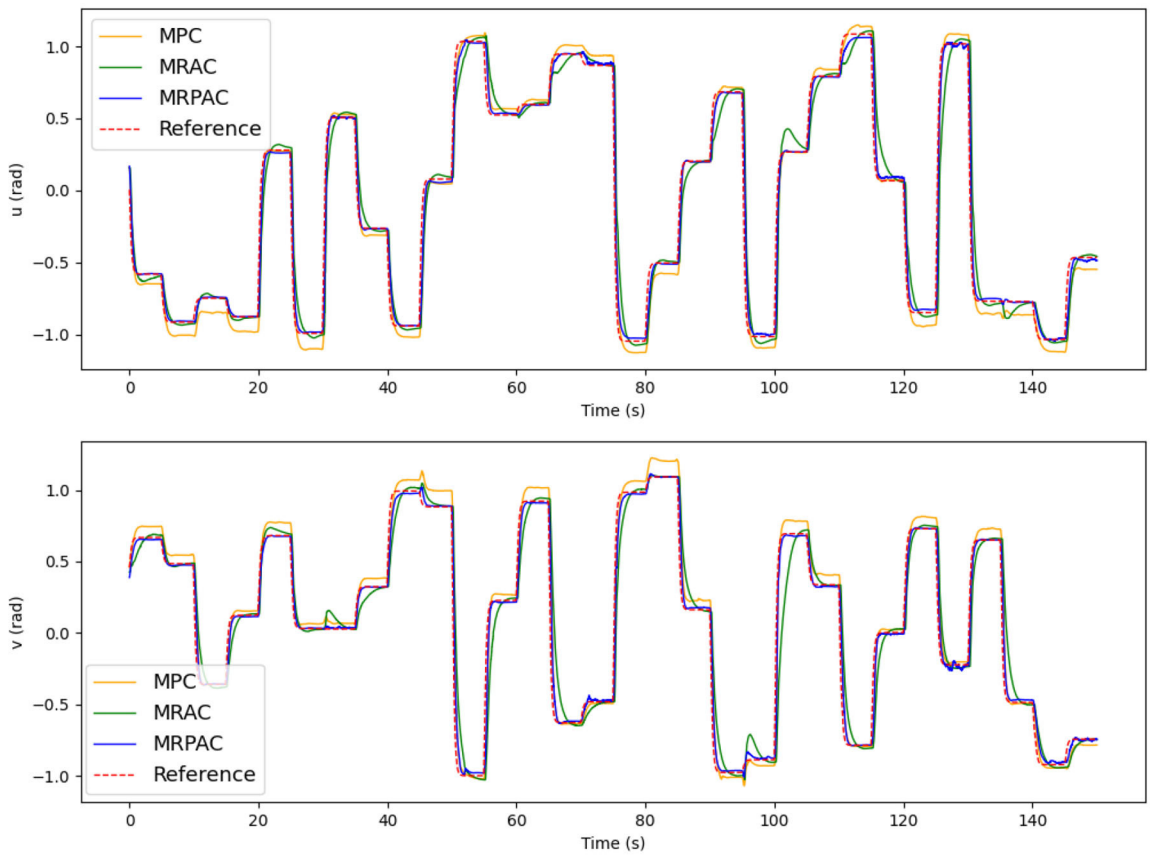


FIGURE 9 | Joint trajectory tracking of all three controllers in hardware. Note that the reference trajectory corresponds to q_{ref}, the position states of our dynamic reference system defined in Equation (36).

model error, especially when inertial, damping, and spring effects are underestimated.

The data presented in **Figure 4** seem to validate the hypothesis that MRAC and MRPAC can both compensate for model error, given a model with the perfect form. We see that MRPAC is able to perform almost identically to MRAC in all cases except when inertial, damping, and spring effects are grossly underestimated. Upon further inspection of the data we found that for this case the adaptive parameters for MRPAC had not quite settled during the 5 min excitation period and that given more time, the tracking performance of MRPAC again approached that of MRAC. This is an interesting and important note - that where MPC performs worst, MRPAC has the most tracking error to overcome, and therefore may take longer to converge its adaptive parameters to a steady state. This suggests that the tuning of Γ and Λ as well as the transient response of the adaptive terms of these controllers are important topics of future research.

4.1.2. Case 2: Imperfect Regressor (Structure Mismatch)

The second experiment was designed to see the sensitivity of each controller to structure mismatch, or model error where the form of the model is not known. The results of this experiment can be seen in **Figure 6**. An example of the joint angle trajectories achieved by each controller is shown in **Figure 7**. As can be seen from the figure, every controller's performance suffers because of this additional modeling error, however MRAC is by far the most sensitive. Note that the x axis of the plot denotes the value of both u and v, and the entire bend angle is equal to $\phi = \sqrt{u^2 + v^2}$. Keeping this in mind, with a spring offset of about 4° ($u = v = 0.05$ radians) MRAC's tracking performance is worse than MPC with 50% error on estimates of masses, lengths and spring and damper coefficients. This represents a very significant decrease in performance due to a relatively small, but completely unmodeled, disturbance. This is the main motivation behind the development of MRPAC. MRPAC can be seen from this figure to inherit from MPC insensitivity to completely unmodeled disturbances or dynamics, and can be seen from **Figure 4** to inherit from MRAC insensitivity to partially modeled disturbances or dynamics.

We can vary the magnitude of both scalar modeling error as well as the unmodeled spring offset in order to develop a surface of tracking error that is a function of both parameter mismatch and structure mismatch. This surface can be seen in **Figure 8**. This is useful information because in reality we are likely to encounter both types of unknowns instead of just one. From the figure we can see that MRPAC consistently has the lowest tracking error of the three controllers, except when MPC has a perfect model or when the model used for MRPAC grossly underestimates inertial, damping, and spring effects. As stated earlier, we have observed that the performance of MRPAC can be improved in the latter case by allowing it to adapt for longer. However, these experimental results outline an important fact, which is that the transient responses of the adaptive terms of MRAC and MRPAC are not the same for the same Γ and Λ values. The exact differences between them and the exact reasons remain for future work.

TABLE 1 | Position tracking error statistics for all three controllers during the 2.5 min evaluation.

	Integrated error	Mean error	Median error	Std. Dev. of error
MPC	18.24	–0.0043	–0.0037	0.1198
MRAC	21.63	–0.0027	–0.0005	0.1829
MRPAC	9.529	–0.0009	–0.0002	0.0924

4.2. Hardware Experiments

The joint trajectories for the hardware experiments are shown in **Figure 9** and the integral of the position tracking error is reported in **Table 1**. It is important to note that, unlike for the simulation, we cannot separate the perfect regressor and imperfect regressor cases on real hardware. Because of the nature of the continuum joint, we expect some combination of both cases to influence the controller performance results.

Generally, we see from the results that MPC struggles to eliminate steady state error. This matches the simulated behavior in **Figure 5** and is expected because MPC does not have the ability to compensate for modeling errors that exist in the continuum joint. MRAC and MRPAC, on the other hand, do have the ability to compensate for modeling errors. Consequently they both track the steady state reference trajectory much closer than MPC. This indicates that the hypothesis presented in section 3.1.1 is demonstratively true at least for this hardware platform. MRAC and MRPAC certainly compensate for the modeling errors and drive the system to follow the reference trajectory. In hardware however, we see that neither controller is capable of following the reference trajectory *exactly*. In other words, we do not see in hardware the same performance as we see in the simulation results in **Figure 5**, where both trajectories deviate very little from the reference. This is because in addition to the modeling error (parameter mismatch) for which MRAC and MRPAC can compensate, there are still system dynamics for which they cannot fully compensate (structure mismatch).

The effect of structure mismatch in simulation is shown in **Figure 6**. Tracking error increases for all control methods as the magnitude of these modeling errors increase, but they increase dramatically for MRAC, hence its poor simulation performance exhibited in **Figure 7**. Importantly, this same pattern emerges in our hardware experiments. There are several instances during the evaluation period where unknown forces cause deviation from the reference trajectory. For examples of this, see the upper plot (u) of **Figure 9** at 65, 100, and 135 s and the bottom plot (v) at 30, 45, and 95 s. All controllers are negatively affected, but MPC and MRPAC are more robust than MRAC. In other words, when encountering such disturbances, MRAC is forced to artificially adapt dynamic parameters in an attempt to eliminate the error. In contrast, MPC and MRPAC are better able to respond to disturbances because they re-solve the trajectory optimization over the whole time horizon, not just a single time step. These results support the hypothesis outlined in section 3.1.2 as well. MRAC and MRPAC do not track the reference trajectory perfectly because of the unknown disturbances but MPC and MRPAC are quantifiably more robust to the structure mismatch.

The results reported in **Table 1** add a quantitative performance analysis in addition to the qualitative analysis from **Figure 9**. From the table we can see that MRPAC accumulates about half of the integrated tracking error of the other two controllers during the 1 min evaluation. It is interesting to note that MPC and MRAC have similar integrated tracking error, although qualitatively their trajectories look different. While MPC has a good transient response and large steady state error, MRAC has a poor transient response and small steady state error. This is also reflected in the statistics, since MRAC has lower mean and median error than MPC, but a higher standard deviation. According to these results, it seems that MRPAC has taken the strengths of the two approaches yielding a good transient response and smaller steady state error.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a novel dynamic modeling approach for one joint of a continuum joint robot. We have shown that while not linear in the same parameters as rigid robots, joint accelerations using this model can be shown to be linear in other dynamic and kinematic parameters. This linearity in model parameters can be exploited for system identification, or as we show later in the paper, for adaptive control. Future work in the area of continuum joint dynamic modeling may include system identification on hardware, as well as verification that the proposed model accurately describes the joint's dynamics. While the presented model is only valid for one joint, another straightforward extension to this work would be to derive dynamic models using the same ideas and assumptions (constant curvature assumptions, u and v parameterization) in order to derive a dynamic model for a robot with many joints and links.

In this paper we have also shown that MPC is an effective control strategy for maintaining robustness to unmodeled forces and/or dynamics. Medium to high fidelity models (such as the one presented in this paper) are promising as a means of reducing these unmodeled disturbances, but take time and effort to develop with possibly very small gains in performance. Even equipped with a perfect model, determining soft robot model kinematic and dynamic parameters accurately is a formidable task and these parameters may also change over time. As such, our presented control strategy, MRPAC, contributes a novel approach to overcoming these challenges by adapting the dynamic model while still leveraging the benefits of MPC.

Specifically, MRPAC inherits two invaluable traits: the adaptive capabilities of MRAC and the robustness of MPC. As a result, MRPAC outperforms both MPC and MRAC on a soft continuum joint, where both parameter mismatch (such as unknown spring and damper coefficients) and structure mismatch (such as unmodeled external forces or offsets) exist. MRPAC successfully compensates for modeling errors to eliminate steady state error while also demonstrating robustness to modeling disturbances.

Future research into MRPAC should include investigation into how to identify a minimal regressor that accurately represents a system's dynamics. Although not discussed in this work, the time taken by MRAC and MRPAC to converge to steady-state adaptive parameters was notably different. For MRPAC it depended heavily on the initial model parameters. The exact differences between the transient response of each control method as well as investigation into the reasons for these differences is left to future work. While our approach has shown promising results, we also did not compare it to other adaptive MPC formulations. Nor do we make the claim that it is the best adaptive MPC formulation. Future work should likely include a comparison between our approach and other existing methods.

Although the problems of accurate soft robot modeling and control remain interesting and unsolved problems, we believe that the dynamic model and adaptive control methods presented in this work represent an important contribution as a new approach to soft robot control.

DATA AVAILABILITY STATEMENT

The data of this article will be available upon request to the corresponding author.

AUTHOR CONTRIBUTIONS

PH and CJ participated in this research as partial fulfillment of their degrees from Brigham Young University. Furthermore, CJ participated in the data collection, problem formulation, and paper writing. MK participated in this research as an advising professor. All authors contributed to the article and approved the submitted version.

FUNDING

This material is based upon work supported by the National Science Foundation under Grant no. 1935312.

REFERENCES

Abdollahi, A., and Chowdhary, G. (2019). Adaptive-optimal control under time-varying stochastic uncertainty using past learning. *Int. J. Adapt. Control Signal Process.* 33, 1803–1824. doi: 10.1002/acs.3061

Adetola, V., DeHaan, D., and Guay, M. (2009). Adaptive model predictive control for constrained nonlinear systems. *Syst. Control Lett.* 58, 320–326. doi: 10.1016/j.sysconle.2008.12.002

Allen, T. F., Rupert, L., Duggan, T. R., Hein, G., and Albert, K. (2020). "Closed-form non-singular constant-curvature continuum manipulator kinematics," in *2020 3rd IEEE International Conference on Soft Robotics* (New Haven, CT: RoboSoft), 410–416. doi: 10.1109/RoboSoft48309.2020.9116015

Best, C. M., Gillespie, M. T., Hyatt, P., Rupert, L., Sherrod, V., and Killpack, M. D. (2016). A new soft robot control method: Using model predictive control for a pneumatically actuated humanoid. *IEEE Robot. Autom. Magaz.* 23, 75–84. doi: 10.1109/MRA.2016.2580591

Bruno, S., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*. London: Springer Science & Business Media.

Bujarbaruah, M., Zhang, X., Rosolia, U., and Borrelli, F. (2018). "Adaptive mpc for iterative tasks," in *2018 IEEE Conference on Decision and Control (CDC)* (Miami Beach, FL), 6322–6327. doi: 10.1109/CDC.2018.8618694

Chowdhary, G., Mühlegg, M., How, J. P., and Holzapfel, F. (2013). "Concurrent learning adaptive model predictive control," in *Advances in Aerospace Guidance, Navigation and Control*, eds Q. Chu, B. Mulder, D. Choukroun, E. J. van Kampen, C. de Visser, G. Looye (Delft; Berlin; Heidelberg: Springer), 29–47. doi: 10.1007/978-3-642-38253-6_3

Della Santina, C., Bicchi, A., and Rus, D. (2020a). On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. *IEEE Robot. Autom Lett.* 5, 1001–1008. doi: 10.1109/LRA.2020.2967269

Della Santina, C., Katzschmann, R. K., Bicchi, A., and Rus, D. (2020b). Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment. *Int. J. Robot. Res.* 39, 490–513. doi: 10.1177/0278364919897292

Falkenhahn, V., Mahl, T., Hildebrandt, A., Neumann, R., and Sawodny, O. (2014). "Dynamic modeling of constant curvature continuum robots using the euler-lagrange formalism," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Chicago, IL), 2428–2433. doi: 10.1109/IROS.2014.6942892

Falkenhahn, V., Mahl, T., Hildebrandt, A., Neumann, R., and Sawodny, O. (2015). Dynamic modeling of bellows-actuated continuum robots using the euler–lagrange formalism. *IEEE Trans. Robot.* 31, 1483–1496. doi: 10.1109/TRO.2015.2496826

Giri, N., and Walker, I. D. (2011). "Three module lumped element model of a continuum arm section," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Francisco, CA), 4060–4065. doi: 10.1109/IROS.2011.6094909

Godage, I. S., Branson, D. T., Guglielmino, E., Medrano-Cerda, G. A., and Caldwell, D. G. (2011). "Shape function-based kinematics and dynamics for variable length continuum robotic arms," in *IEEE International Conference on Robotics and Automation* (Shanghai), 452–457. doi: 10.1109/ICRA.2011.5979607

Hyatt, P., and Killpack, M. D. (2020). Real-time nonlinear model predictive control of robots using a graphics processing unit. *IEEE Robot. Autom. Lett.* 5, 1468–1475. doi: 10.1109/LRA.2020.2965393

Hyatt, P., Williams, C. S., and Killpack, M. D. (2020). Parameterized and gpu-parallelized real-time model predictive control for high degree of freedom robots. *arXiv preprint* arXiv:2001.04931.

Hyatt, P., Wingate, D., and Killpack, M. D. (2019). Model-based control of soft actuators using learned non-linear discrete-time models. *Front. Robot. AI* 6:22. doi: 10.3389/frobt.2019.00022

Kang, R., Kazakidi, A., Guglielmino, E., Branson, D. T., Tsakiris, D. P., Ekaterinaris, J. A., et al. (2011). "Dynamic model of a hyper-redundant, octopus-like manipulator for underwater applications," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Francisco, CA), 4054–4059. doi: 10.1109/IROS.2011.6094468

Khalil, W., Member, S., and Gallot, G. (2007). Dynamic modeling and simulation of a 3-D serial Eel-Like robot. *IEEE Trans. Syst. Man Cybern. Part C* 37, 1259–1268. doi: 10.1109/TSMCC.2007.905831

Kim, J.-S. (2010). "Recent advances in adaptive mpc," in *ICCAS 2010* (Gyeonggi-do), 218–222. doi: 10.1109/ICCAS.2010.5669892

Lavretsky E., and Wise K.A. (2013) "Robust adaptive control," in *Robust and Adaptive Control. Advanced Textbooks in Control and Signal Processing* (London: Springer). doi: 10.1007/978-1-4471-4396-3_11

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., et al. (2017). Sympy: symbolic computing in python. *PeerJ Comput. Sci.* 3:e103. doi: 10.7717/peerj-cs.103

Mochiyama, H., and Suzuki, T. (2002). "Dynamical modelling of a hyper-flexible manipulator," in *Proceedings of the 41st SICE Annual Conference. SICE 2002.*, Vol. 3 (Osaka), 1505–1510. doi: 10.1109/SICE.2002.1196530

Mochiyama, H., and Suzuki, T. (2003). "Kinematics and dynamics of a cable-like hyper-flexible manipulator," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, Vol. 3 (Taipei), 3672–3677. doi: 10.1109/ROBOT.2003.1242160

Pereida, K., and Schoellig, A. P. (2018). "Adaptive model predictive control for high-accuracy trajectory tracking in changing conditions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid), 7831–7837. doi: 10.1109/IROS.2018.8594267

Renda, F., Cianchetti, M., Giorelli, M., Arienti, A., and Laschi, C. (2012). A 3D steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm. *Bioinspir. Biomimet.* 7:025006. doi: 10.1088/1748-3182/7/2/025006

Shepherd, R. F., Ilievski, F., Choi, W., Morin, S. A., Stokes, A. A., Mazzeo, A. D., et al. (2011). Multigait soft robot. *Proc. Natl. Acad. Sci.U.S.A.* 108, 20400–20403. doi: 10.1073/pnas.1116564108

Slotine, J. J. E. and Li, W. (1987). on the adaptive control of robot manipulators. *Int. J. Robot. Res.* 6, 49–59.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. (2017). OSQP: an operator splitting solver for quadratic programs. *ArXiv e-prints*.

Tatlicioglu, E., Walker, I. D., and Dawson, D. M. (2007). "New dynamic models for planar extensible continuum robot manipulators," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Diego, CA), 1485–1490. doi: 10.1109/IROS.2007.4399334

Terry, J. S., Whitaker, J., Beard, R. W., and Killpack, M. D. (2019). "Adaptive control of large-scale soft robot manipulators with unknown payloads," in *Proceedings of the ASME 2019 Dynamic Systems and Control Conference*, Vol. 3 (Park City, UT: ASME). doi: 10.1115/DSCC2019-9037

Thuruthel, T., Falotico, E., Cianchetti, M., Renda, F., and Laschi, C. (2016). "Learning global inverse statics solution for a redundant soft robot)," in *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics* (Lisbon), 303–310. doi: 10.5220/0005979403030310

Thuruthel, T. G., Falotico, E., Renda, F., and Laschi, C. (2017). Learning dynamic models for open loop predictive control of soft robotic manipulators. *Bioinspir. Biomimet.* 12:066003. doi: 10.1088/1748-3190/aa839f

Tolley, M. T., Shepherd, R. F., Mosadegh, B., Galloway, K. C., Wehner, M., Karpelson, M.,et al. (2014). A resilient, untethered soft robot. *Soft Robot.* 1, 213–223. doi: 10.1089/soro.2014.0008

Trumić, M., Jovanović, K., and Fagiolini, A. (2020). Decoupled nonlinear adaptive control of position and stiffness for pneumatic soft robots. *Int. J. Robot. Res.* doi: 10.1177/0278364920903787

Walker, I. D. (2013). Continuous backbone "Continuum" robot manipulators. *ISRN Robot.* 2013, 1–19. doi: 10.5402/2013/726506

Zhang, H., Cao, R., Zilberstein, S., Wu, F., and Chen, X. (2017). "Toward effective soft robot control via reinforcement learning," in *Intelligent Robotics and Applications. ICIRA 2017. Lecture Notes in Computer Science*, Vol. 10462, eds Y. Huang, H. Wu, H. Liu, and Z. Yin (Wuhan; Cham: Springer). doi: 10.1007/978-3-319-65289-4_17

Zhang, K., and Shi, Y. (2020). Adaptive model predictive control for a class of constrained linear systems with parametric uncertainties. *Automatica* 117:108974. doi: 10.1016/j.automatica.2020.108974

Zheng, T., Branson, D. T., Kang, R., Cianchetti, M., Guglielmino, E., Follador, M., et al. (2012). "Dynamic continuum arm model for use with underwater robotic manipulators inspired by *Octopus vulgaris*," in *2012 IEEE International Conference on Robotics and Automation* (Saint Paul, MN), 5289–5294. doi: 10.1109/ICRA.2012.6224685

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.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 05 October 2020
doi: 10.3389/frobt.2020.00115

Controlling of Pneumatic Muscle Actuator Systems by Parallel Structure of Neural Network and Proportional Controllers (PNNP)

Alaa Al-Ibadi[1,2]*, *Samia Nefti-Meziani*[1] and *Steve Davis*[1]

[1] School of Computing, Science and Engineering, University of Salford, Salford, United Kingdom, [2] Computer Engineering Department, Engineering College, University of Basrah, Basrah, Iraq

This article proposed a novel controller structure to track the non-linear behavior of the pneumatic muscle actuator (PMA), such as the elongation for the extensor actuator and bending for the bending PMA. The proposed controller consists of a neural network (NN) controller laid in parallel with the proportional controller (P). The parallel neural network proportional (PNNP) controllers provide a high level of precision and fast-tracking control system. The PNNP has been applied to control the length of the single extensor PMA and the bending angle of the single self-bending contraction actuator (SBCA) at different load values. For further validation, the PNNP has been applied to control a human–robot shared control system. The results show the efficiency of the proposed controller structure.

Keywords: controller system, PMA, neural network, P controller, human-robot shared controller

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Wei Meng, Wuhan University of Technology, China
Zool H. Ismail, University of Technology Malaysia, Malaysia

***Correspondence:**
Alaa Al-Ibadi
alaa.falah77@yahoo.fr;
a.f.a.al-ibadi@edu.salford.ac.uk

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *30 April 2020*
Accepted: *22 July 2020*
Published: *05 October 2020*

Citation:
Al-Ibadi A, Nefti-Meziani S and Davis S (2020) Controlling of Pneumatic Muscle Actuator Systems by Parallel Structure of Neural Network and Proportional Controllers (PNNP). Front. Robot. AI 7:115. doi: 10.3389/frobt.2020.00115

INTRODUCTION

Soft robotics represents a new generation of robotic research, which provides numerous advantages, such as being lightweight, safe for close contact with humans, and environmentally friendly, as well as having low cost in terms of materials, construction time, and power (Neppalli and Jones, 2007; Trivedi et al., 2008; Al-Ibadi et al., 2020). In addition to the general advantages of soft robotics, soft actuators, such as contraction and extension pneumatic muscle actuators (PMAs), have their benefits when compared with the traditional electrical and mechanical actuators. Moreover, there is a high ratio of force to the actuator weight, in most cases a 100 newtons for several 100 g (Tondu and Lopez, 2000; Al-Ibadi et al., 2017, 2018a; Yang et al., 2019), but on the other hand, due to the softness, low stiffness, and hysteresis, the PMA shows a high degree of non-linearity and adds more challenges to controlling such types of actuators (Wang et al., 2017; Giannaccini et al., 2018; Teramae et al., 2018).

The performances of soft robots provide infinite degrees-of-freedom (DoF) motions, such as elongation, contraction, bending, shrinkage, and rotation. Furthermore, different designs and actuation techniques give unique behaviors (Manti et al., 2016; Al-Ibadi et al., 2018b; George Thuruthel et al., 2018), and the value mechanism and the high rubber material non-linearity of the PMA make the control process difficult and rule out simple controllers. Therefore, to overcome these difficulties, the high robust control has to be considered (Tondu and Lopez, 2000; Leephakpreeda, 2011). Numerous types of control strategies were used to control the position and force of the PMA. Among them, a linear proportional–integral–derivative (PID) controller has been used in Andrikopoulos et al. (2011), Shen et al. (2015), and Chan et al. (2020). Four PID

controllers have been used to control the orientation of the ankle rehabilitation robot of four PMAs, one controller for each actuator (Meng et al., 2017). Adaptive pole placement techniques for positioning PID controllers were applied in Bowler (1996). A sliding mode control was used in Cai and Yamaura (1997) and Carbonell et al. (2001), and a fuzzy sliding mode controller, which is trained by a neural network for single dimensional PMA, is used in Chiang and Chen (2017) and Chiang and Chen (2018). Fuzzy PID was used in Balasubramanian and Rattan (2003), and a fuzzy PD controller and an integration controller were used in Chan et al. (2003). Tracking control with hysteresis compensation was done by PID (Schreiber et al., 2011). A series combination of PID controllers and an artificial neural network (ANN)—non-linear PID—was used in Thanh and Ahn (2006) for physical rehabilitation by using multijoint actuate based on pneumatic muscles. Similar techniques have been used in Andrikopoulos et al. (2014).

This article aims to provide an efficient, simple structure, controller system to be used for various soft robotic systems. For that purpose, a parallel controller structure is proposed by using a neural network (NN) controller and a proportional (P) controller. This structure provides a fast and accurate response to track the soft pneumatic robot systems. The proposed controller has been used to control the position of single actuators, the bending angles of the self-bending contraction actuator (SBCA), and a human–robot shared control system to show the efficiency of the proposed controller for different robot behaviors and applications.

The order of this paper has been organized as follows: Section non-linear PID controller shows the idea of non-linear PID; Section other controller approaches describes several approaches to control the soft pneumatic systems. The proposed controller structure is presented in Section parallel neural network proportional controller together with its applications.

NON-LINEAR PID CONTROLLER

The PID controller has been one of the most important strategies used in industrial applications due to its simplicity and robustness. The need for variable efficient controller performance in operating conditions or parameters in the environment is often beyond the abilities of linear PID controllers (Su et al., 2005). Moreover, the high non-linearity of the PMAs makes the PID controller insufficient to solve this complex control problem. To improve the performance of linear PID to control the performances of PMA, numerous techniques have been utilized to enhance the performance and robustness of the PID controller by using the self-tuning method of general predictive control, fuzzy logic, and neural networks (Cervantes and Alvarez-Ramirez, 2001; Duan et al., 2004). **Figure 1** shows the non-linear PID by connecting it serially to the ANN.

A multilayer ANN is used for three inputs, one neuron in one hidden layer and one output neuron, with

$$e_P\left(k\right) = \theta_{ref}\left(k\right) - \theta\left(k\right) \tag{1}$$

$$e_I\left(k\right) = e_P\left(k\right)\Delta T \tag{2}$$

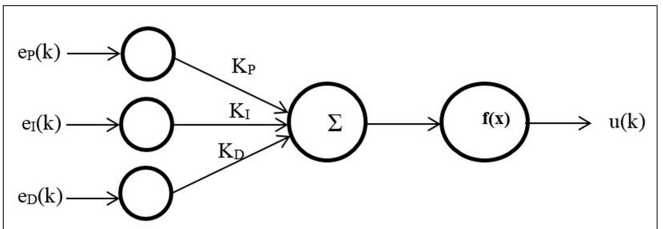


FIGURE 1 | Proportional–integral–derivative (PID)-artificial neural network (ANN) control structure.

$$e_D\left(k\right) = \frac{e_P(k)(1 - z^{-1})}{\Delta}T \tag{3}$$

where θ_{ref} and θ are the setpoints and the actual output for each joint, respectively; ΔT is the sampling time; z is the Z-transform operator; K_p, K_I, and, K_D are PID constants that have to be modified to find the optimal value; $f(x)$ is a sigmoid function; and $u(k)$ is the controller output.

Other work was done in Anh (2010) using the same PID-ANN technique with added bias input to the hidden and output neurons.

OTHER CONTROLLER APPROACHES

The inverse control strategy for PMA motion control was presented in Kang et al. (2013) and Kang et al. (2014). By using this idea, they were able to define an inverse kinematic (IK) model for control application. Furthermore, they assumed that the dynamics of the system could be ignored because the speed of these types of actuators is low. Meanwhile, Nakamura and Shinohara (2007) presented the controller system according to the mathematical model of PMA, which drives the inverse relationship between both the position and force of the PMA and the pressure input where P is the function of L and F.

The fuzzy control based on bang-bang control strategy is used in Leephakpreeda (2011) with a combination of proportional control to adjust the system output around the desired points either for the length of the contraction force. **Figure 2** shows a diagram of this control system. In this method, the author used the pulse width modulation (PWM) technique as a variable time on–off controller to adjust the air valve outlets. The model-based statics controller has been utilized in Camarillo et al. (2009) for a 5-DoF-*per-sec*tion model by formulating IK. The most frequently used of IK-based static controls uses the constant curvature (CC) approximation (Hannan and Walker, 2003).

On the other hand, model-free approaches for control of soft robots are quite a novel field and provide a wide range of possibilities. The early utilization of this method has been proposed in Giorelli et al. (2013) and for a 2- and a 3-DoF cable-driven soft manipulator (Giorelli et al., 2015a,b). The main idea for this controller system is applying a closed-loop control system with an effective sensory feedback system.

PARALLEL NEURAL NETWORK PROPORTIONAL CONTROLLER

The parallel neural network proportional (PNNP) controller is suggested in this section. The NARMA-L2 neural network control system has been utilized. The structure of nine neurons has been chosen in a single hidden layer, three delayed controlled signal outputs, and two delayed plant outputs. The NN has been trained by trainlm for 100 Epochs. The mean square error (MSE) for the training, testing, and validating data is about 10^{-7}. The NN controller system provides good performance; nonetheless, the PMA system is too slow, and it needs a fast controller to track its behavior. To enhance the speed of the controller system, a proportional controller has been used in parallel to the NN controller. While the NN controller provides high precision, the P controller offers a high-speed response. As a result, the structure of the PNNP controller provides efficient performances in terms of precision and speed. The structure of the controller is shown in **Figure 3**.

The reference model states the required target such as length, position, bending angle, and so on. Moreover, since the air pressure in single or multiple PMA defines the system performances, G1 can either be the inverse kinematics of the plant and, in this case, the proposed controller will adjust the pressure p, or be equal to 1 and, in this case, the controller system will track the error in the output y.

The error e can be defined as follows:

$$e = p_r - p \,, if\ G1\ is\ IK \tag{4}$$

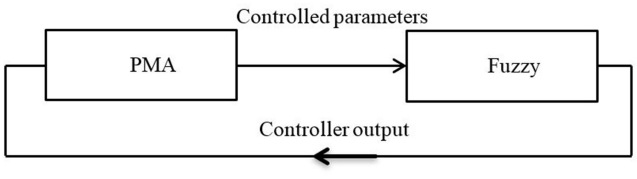


FIGURE 2 | The fuzzy logic block diagram controller system.

or

$$e = y_r - y \,, if\ G1\ is\ 1 \tag{5}$$

p_r and y_r are the reference (setpoint) for the pressure and the system's output, respectively.

The controller outputs $u1$ and $u2$ represent the duty cycle of the PWM signal for the NN controller and P controller, respectively, where

$$u = u1 + u2 \tag{6}$$

The NARMA-L2 NN controller output $u1$ can be defined as

$$u1\left(k\right) = \frac{y_r\left(k+1\right) - f[y_n\left(k\right),\ u1_m\left(k-1\right)]}{g[y_n\left(k\right),\ u1_m\left(k-1\right)]} \tag{7}$$

where $f()$ and $g()$ are approximated using neural networks, and

$$y_n\left(k\right) = [y\left(k\right),\ \ldots,\ y(k-n+1)]^T \tag{8}$$

$$u1_m\left(k-1\right) = [u1\left(k-1\right),\ u1\left(k-2\right),\ \ldots,\ u1(k-m)]^T \tag{9}$$

where n and m are equal to 2 and 3, respectively, according to the proposed controller structure.

While the proportional controller output has been defined as

$$u2\left(k\right) = k_p\,\frac{(y_r\left(k+1\right) - y(k+1))(u_{\max})}{x} \tag{10}$$

The PWM signal controls the airflow for the valve output in fill and vent directions. Therefore, two PNNP controllers are required: one to control the airflow in the fill direction and the other controls the venting process.

Depending on the error, the proposed controller activates either the filling controller (positive error) or the venting controller (negative error). On the other hand, two possible methods are used to train the NN. The first method is using an approximate function between the output and the duty cycle as in (11):

$$y^* = y_0 + \frac{x\,u^*}{98} \tag{11}$$

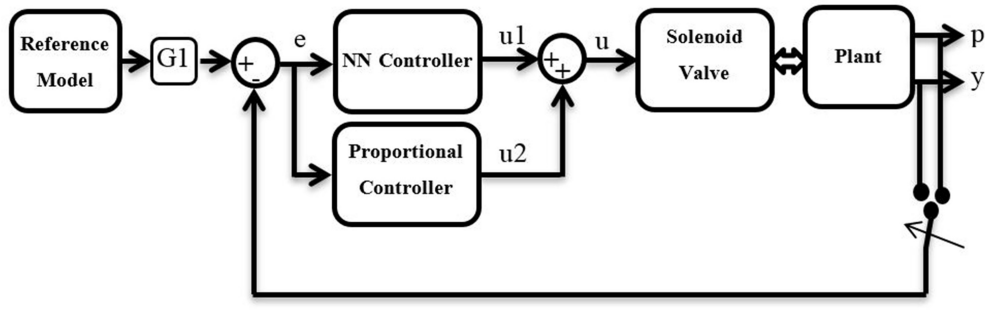


FIGURE 3 | The schematics of the suggested controller.

In the case of the pressure controller of the single or multiple actuators, y^* and y_0 represent the air pressure in the actuator p and the initial pressure in the actuator, respectively, x is the maximum applied pressure p_{max} (p_{max} is subject to the actuator size and material), and u^* is the training duty cycle of the NN controller.

In most cases,

$$p_{\max} = 500\ kPa \tag{12}$$

$$0 \leq p \leq p_{\max} \tag{13}$$

$$x = p_{\max} \tag{14}$$

$$0 \leq u^* \leq 100 \tag{15}$$

In order to prevent a continuously applied voltage (100% duty cycle) on the solenoid valve, we chose 98% as the maximum operating duty cycle. Formula (11) provides an acceptable linear performance of the PMA pressure at variable duty cycles.

Alternatively, the actual relationship between the output and the duty cycle can be found experimentally as follows.

A contraction actuator of 30 cm in length and 1.7 cm in diameter is chosen. A source of 600 kPa is used to apply air pressure to this actuator via a solenoid valve by different duty cycles ranging from 0 to 100% within 1 s. Firstly, a 10% duty cycle is selected, the air pressure is measured by a pressure sensor, and then the venting process is activated. This process is repeated for 20, 30…, and 100%, respectively. The result of this experiment is shown in **Figure 4**. The trained line to these data is utilized for training the NN.

Formula (11) is used due to the similarity in the performances of the NN and to decrease the complexity of the control system. Moreover, the PNNP controller is tracking the desired behavior online; therefore, the controller is adjusting the duty cycle to minimize the error.

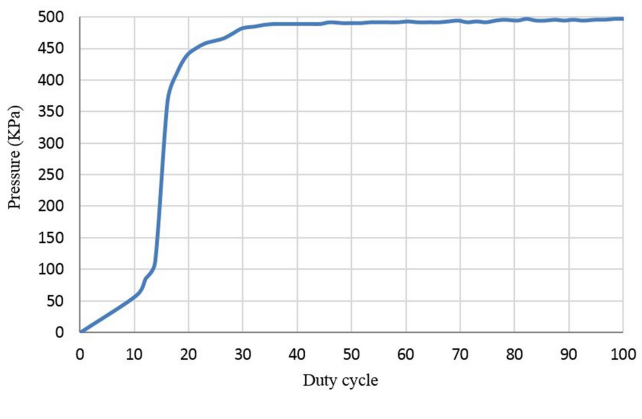


FIGURE 4 | The relation between air pressure and duty cycle.

Length Control of Single Extensor PMA

In order to validate the proposed controller, a 30-cm extensor actuator is chosen. To measure the change in length of the extensor PMA, an ultrasound HC-SR04 sensor is fixed to the end of the air muscle. According to (11), y^* is the length of the actuator L, y_0 is the initial length L_0 of the extensor PMA (30 cm), and x is the maximum extension ratio (50% of L_0).

The PNNP controller sends the controlled input u to the (3/3 Matrix MK 754.8E1D2XX) solenoid valve via Arduino Mega 2560. The Arduino acts as an interface between the PC and the valve-actuator system. It is reading the pressure and the distance from the pressure sensor and the ultrasound sensor and sending them to the Matlab *via* a USB port. Then, The PNNP controller adjusts the duty cycles for both

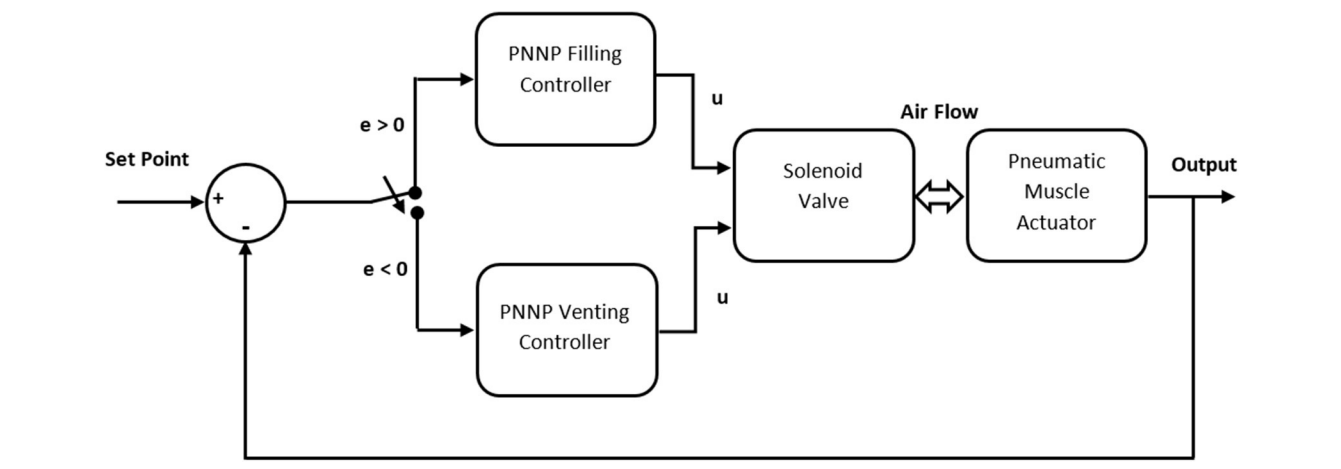


FIGURE 5 | The structure of the proposed controller for single PMA.

the filling and the venting and sends them back to the valve as follows:

While the actuator air pressure is low and the PMA length is less than the required, the error will be positive and that activates the filling PNNP controller branch (see **Figure 5**) to actuate the extensor muscle and increase its length. The controlled duty cycle u decreases gradually according to the feedback error until the error becomes zero; at this point, the filling PNNP controller is being inactivated. Due to the hysteresis behavior of the PMA, the length of the actuator will be slightly increasing, which leads to a negative error. The venting PNNP controller responds to the increment of the actuator length by decreasing the amount of pressure. The expected maximum controlled duty cycle is low because of the small error value. As a result, the length of the actuator decreases until the error reaches zero. This process of filling and venting might be repeated several times according to the sign of the feedback error. The operation at low frequencies decreases the number of the filling and venting controlling process repetitions because the whole controlling process speed operates close to the pneumatic system behaviors.

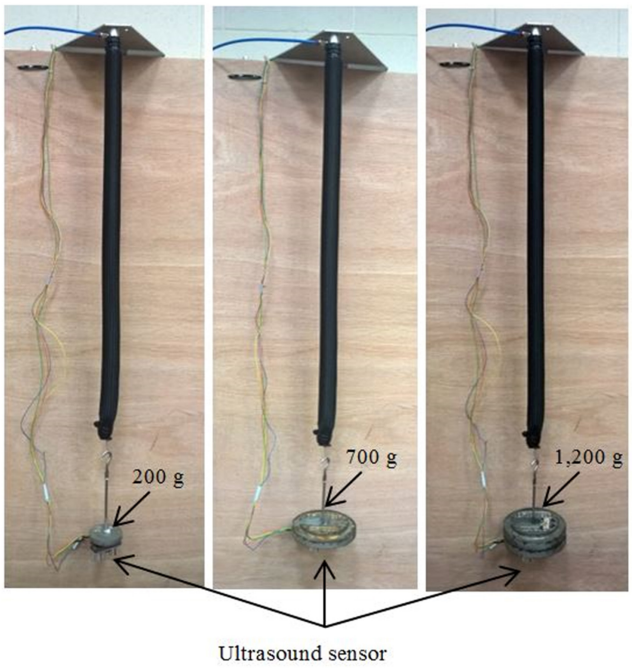


FIGURE 6 | The photograph of the extensor actuator at various loads and the position of the attached ultrasound sensor.

Attaching a load or increasing its value causes increasing in the actuator length. To keep the length as required by the reference model, the venting controller operates to reduce the length, and, of course, the filling will be activated if overcontraction occurs.

The length of the actuator is controlled under three different load values. At each time, a square wave between 30 and 45 cm is applied as a reference at 0.5 Hz. The extensor actuator and the control performance are illustrated in **Figures 6**, **7**, respectively, for 200 g.

Figure 7 shows that the venting time takes longer than the filling time due to the hysteresis of the actuator material and the air pressure difference between the environments.

Bending Angle Control of Single SBCA

The presented SBCA in Al-Ibadi et al. (2020) has been used in this section to control its bending angle. The specifications of the bending actuator are listed in **Table 1**.

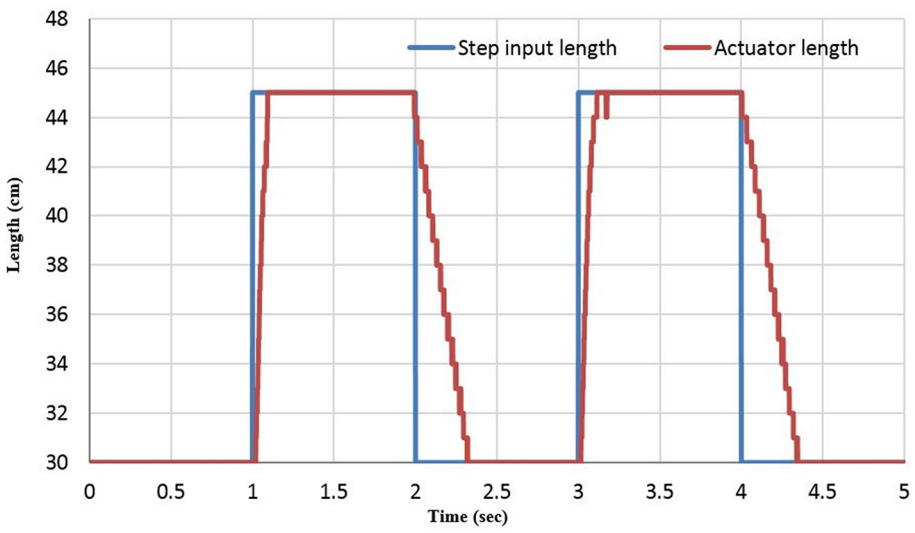


FIGURE 7 | The unit step response of the length controller system at 0.5 Hz.

TABLE 1 | The dimensions and material specifications of the SBCA.

L_0 (m)	Rubber thickness (m)	Braided thickness (m)	Inner diameter (m)	Rubber stiffness(N/m)	Rod length (m)	Rod thickness (m)	Rod width (m)
0.3	1.1×10^{-3}	0.5×10^{-3}	12×10^{-3}	363.33	0.3	0.002	0.006

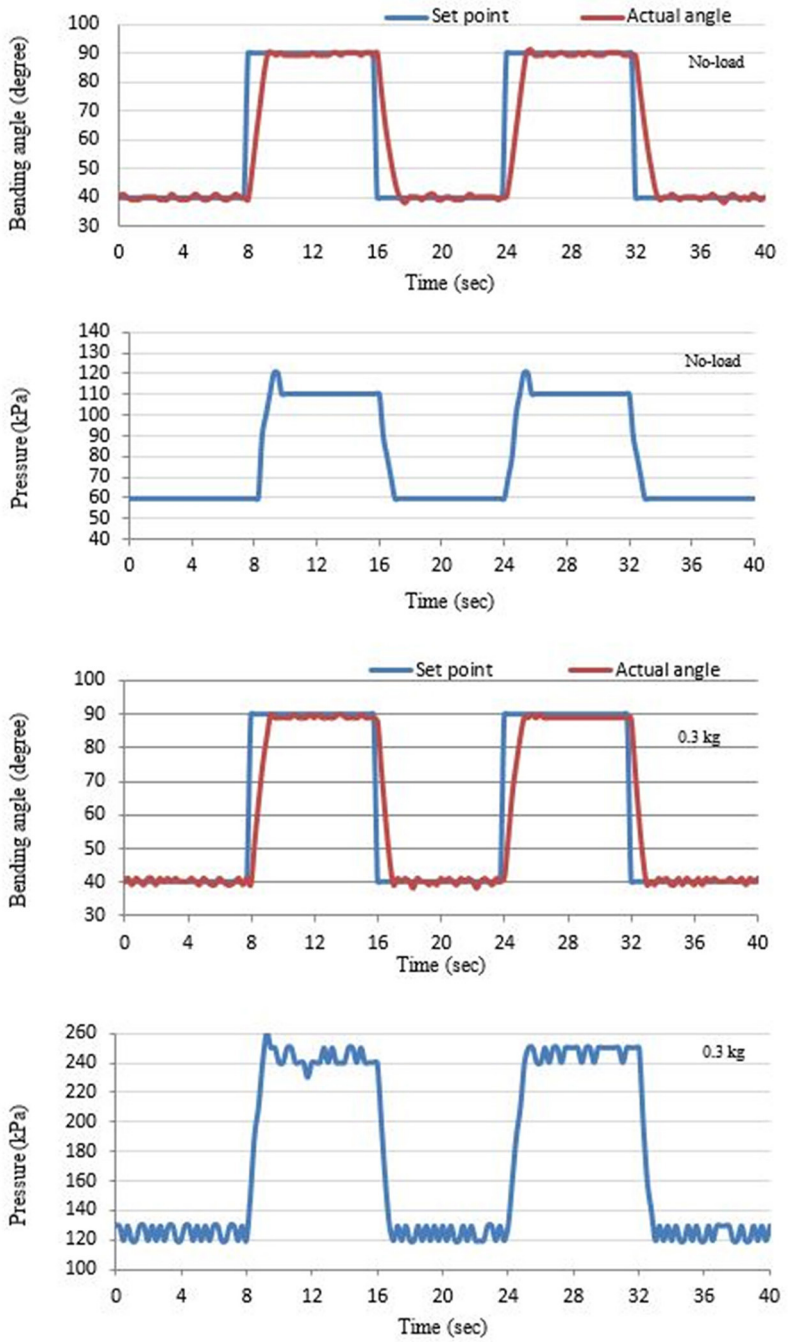


FIGURE 8 | The bending angle and the applied pressure of the SBCA at no load and 0.3 kg.

A similar controller has been used to control the bending angle at different load values, but, in this case, the initial bending angle is zero, and the maximum bending angle for the chosen specification in **Table 1** is 135° at 500 kPa. Square wave between 40 and 90 is selected as a reference model for tracking the bending angle of the bending actuator between 40 and 90°. **Figure 8** illustrates the controller response at no load and 0.3 kg. Pressure and MPU sensors are used to record the actuator air pressure and the bending angle, respectively. The MPU is mounted at the free end of the SBCA. A similar procedure to the extensor actuator can be shown here; the positive 40° feedback error triggers the filling PNNP to apply air pressure to the SBCA. The actuator bending angle increases to 40°. The zero-feedback error isolates the filling controller to avoid increasing the bending angle. Again, because of the non-linear behaviors of the PMA, the bending angle might increase by some degrees. This leads to the activation of the venting controller until the error reaches zero again. Similar processes are applied to the second value of the reference signal (90°). For the second cycle of the reference square signal, the venting controller operates first, and the filling PNNP responds to the decreasing bending angle. Furthermore, increasing the load value at any moment leads to the reduction of the bending angle, and more air pressure is required to reduce the positive error by the filling controller.

Figure 8 shows that the controller system applied more air pressure when the load is increased to reach the required bending angle.

Human–Robot (H–R) Interaction (HRI)

Unsafe workspaces for individuals force them to work from a split site. In this section, a unidirectional continuum arm and a four-finger gripper are used to work in a workspace considered to be unsafe for a human being. The MPU and the pressure sensors are used to measure the bending angle of the continuum arm and the air pressure in the finger gripper, respectively. On the other hand, another MPU sensor and a flex sensor are worn by a human hand, as shown in **Figure 9**.

The wearable MPU sensor is used to send the set bending angle to the PNNP controller to adjust the bending angle of the continuum arm, and the flex sensor controls the grasping force of the four-finger gripper by converting the resistance to pressure by mapping its data at different bending steps for the index finger. The control system controls the air pressure in the fingers to control the grasping process.

In this process, the human sends a variable reference bending and grasping force to the controller through the Arduino Mega 2560, and the controller adjusts both of them on the continuum arm. **Figure 10** shows the bending angle for both the human hand and the continuum arm at two different loads.

Since the application consists of two pneumatic systems, four PNNP controllers are required, two filling controllers and two venting controllers. Either the filling PNNP or the venting PNNP controllers are being activated for the gripper to adjust the grasping force as required by the reference value, which is sent by the human index finger. Simultaneously, the bending angle of the human arm is sending to another group of PNNP controllers to adjust the bending angle of the continuum arm.

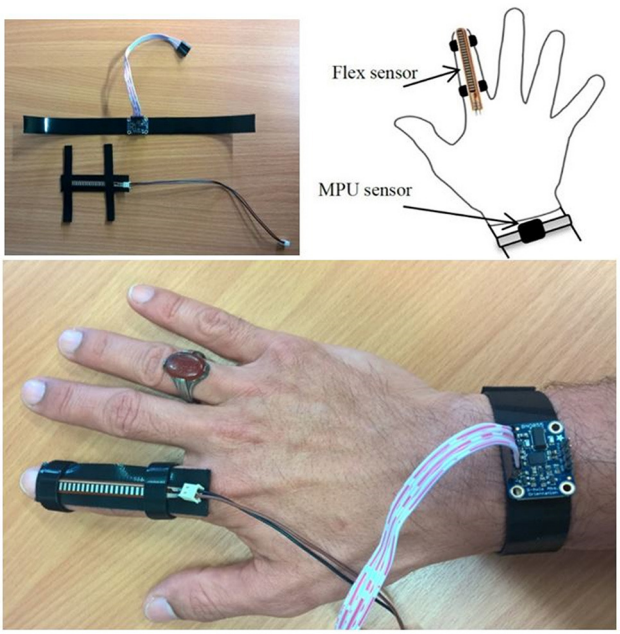


FIGURE 9 | The wearable sensors to control the bending angle and grasping force.

Figure 10 illustrates the efficiency of the PNNP controller, which provides precise tracking for the bending angle of the human arm. As previously mentioned, the tracking error for the filling process is less than the error of the venting process due to the variations between the air pressure in the two different environments.

The comparison with the literature shows that the performance error is very low for the three presented applications. While it is seen obviously at numerous previous researches, such as the performance error for the ankle rehabilitation robot in Meng et al. (2017), the possible cause for that is using the PID controller to control the high non-linear system (the PMA).

COMPARISON WITH PREVIOUS CONTROLLER APPROACHES

The PNNP controller shows efficient performances when it is applied to soft pneumatic systems. In this section, the PNNP is compared with several other controller approaches from the literature to show the advantages of the proposed controller system. **Table 2** lists the main characteristics of numerous controllers.

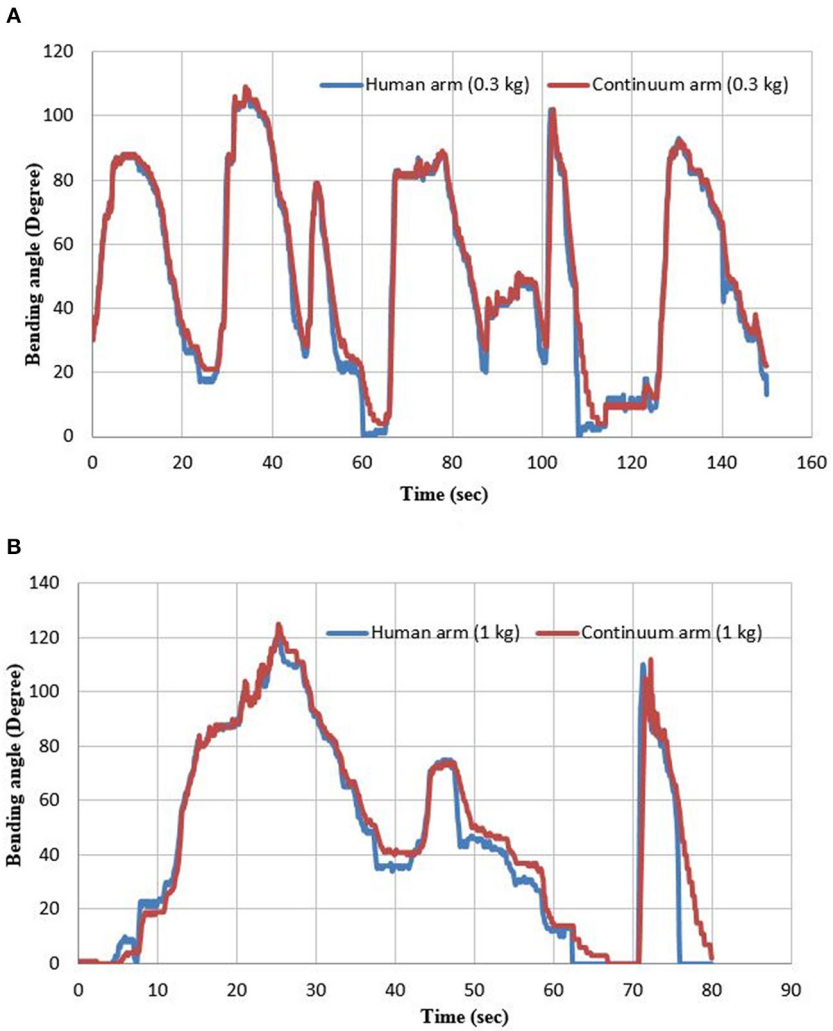


FIGURE 10 | The bending angle for both the human hand and the continuum arm at **(A)** 0.3 and **(B)** 1 kg.

Table 2 shows that the performance of the PNNP controller system in terms of speed response, accuracy, and applications (complexity of the pneumatic system) is higher than other chosen researches.

CONCLUSION

The high non-linear behaviors of the pneumatic muscle actuator require fast response and high accuracy control systems. In this article, a parallel structure of the neural network controller and the proportional controller is presented to control single extensor PMA and single SBCA, respectively, at different load values. For further validation of the PNNP controller, an interaction between a human and a bidirectional continuum arm has been designed, and the controller system shows a valuable tracking to the human hand. The results illustrated the efficiency of using the parallel structure to increase precision and decrease the tracking time.

The results show that the venting time is more than the filling time due to the non-linear behavior of the PMA such as hysteresis and the air pressure difference inside and outside the actuator. Furthermore, increasing the load for the presented pneumatic systems does not have any effect on the resulting performances. Nonetheless, the PNNP decreases the required air pressure for the extensor PMA at a higher load to decrease the extension ratio, while the proposed controller increases the applied air pressure for the single SBCA and the unidirectional continuum arm to raise the bending angle. The actuators are tested at pressure up to 600 kPa, but the maximum air pressure has been set at 500 kPa for safe working.

TABLE 2 | The dimensions and material specifications of the SBCA.

Research title	Type of controller	Linearity	Performance
The Proposed Controller (current article)	Parallel neural network proportional (PNNP)	Non-linear	High response (settling time = 0.15 s at 0.5 Hz), accurate, suitable for single and multiple actuator systems, provides local controlling for every single actuator
Andrikopoulos et al. (2011)	PID and on–off	Linear	Low response (settling time = 16 s), tested for a climbing robot of four PMAs
Chan et al. (2020)	Cascaded PID	Linear	Low response (settling time = 2 s) and tested for single PMA
Shen et al. (2015)	PID	Linear	Moderate response, tested to control a robot leg of two PMAs
Meng et al. (2017)	Iterative feedback tuning control (IF-PID)	Linear	Moderate response (high settling time, the error does not reach zero), tested for four single actuators for ankle rehabilitation system
Chiang and Chen (2017)	Neural network fuzzy sliding mode controller	Non-linear	High performance at frequencies ≤0.05 Hz and low performance at 1 Hz
Chan et al. (2003)	Fuzzy PD+I	Non-linear	Low response (1–2 s), tested for single PMA
Andrikopoulos et al. (2014)	Advanced non-linear PID	Non-linear	Moderate response (settling time = 0.5 s at 0.25 Hz), tested for single PMA

DATA AVAILABILITY STATEMENT

All datasets presented in this study are included in the article/**Supplementary Material**.

ETHICS STATEMENT

Written informed consent was obtained from the individual(s) for the publication of any potentially identifiable images or data included in this article.

AUTHOR CONTRIBUTIONS

AA-I designed, performed the experiments, and wrote the paper. AA-I and SD analyzed the data. SN-M edited the paper. All authors have read and agreed to the published version of the manuscript.

ACKNOWLEDGMENTS

Many thanks to the ministry of higher education and scientific research in Iraq and for the University of Basrah for providing the scholarship support to the first author of this article.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2020.00115/full#supplementary-material

REFERENCES

Al-Ibadi, A., Nefti-Meziani, S., and Davis, S. (2017). Efficient structure-based models for the McKibben contraction pneumatic muscle actuator: the full description of the behaviour of the contraction PM. *Actuators* 6:32. doi: 10.3390/act6040032

Al-Ibadi, A., Nefti-Meziani, S., and Davis, S. (2018b). "A circular pneumatic muscle actuator (CPMA) inspired by human skeletal muscles," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, 7–12. doi: 10.1109/ROBOSOFT.2018.8404889

Al-Ibadi, A., Nefti-Meziani, S., and Davis, S. (2020). The design, kinematics and torque analysis of the self-bending soft contraction actuator. *Actuators* 9:33. doi: 10.3390/act9020033

Al-Ibadi, A., Nefti-Meziani, S., Davis, S., and Theodoridis, T. (2018a). Novel design and position control strategy of a soft robot arm. *Robotics* 7:72. doi: 10.3390/robotics7040072

Andrikopoulos, G., Nikolakopoulos, G., and Manesis, S. (2011). *Development and control of a hybrid controlled vertical climbing robot based on pneumatic muscle actuators*. J. Control Eng. Technol. 1, 53–58.

Andrikopoulos, G., Nikolakopoulos, G., and Manesis, S. (2014). Advanced nonlinear PID-based antagonistic control for pneumatic muscle actuators. *IEEE Trans. Ind. Electron.* 61, 6926–6937. doi: 10.1109/TIE.2014.2316255

Anh, H. P. H. (2010). Online tuning gain scheduling MIMO neural PID control of the 2-axes pneumatic artificial muscle (PAM) robot arm. *Expert. Syst. Appl.* 37, 6547–6560. doi: 10.1016/j.eswa.2010.02.131

Balasubramanian, K., and Rattan, K. S. (2003). "Fuzzy logic control of a pneumatic muscle system using a linearing control scheme," in *22nd International Conference of the North American Fuzzy Information Processing Society, NAFIPS (2003)* (Chicago, IL), 432–36.

Bowler, C. J. (1996). "Pneumatic muscle actuators: musculature for an anthropomorphic robot arm," in *IEE Colloquium on Actuator Technology: Current Practice and New Developments*, 8. doi: 10.1049/ic:19960699

Cai, D., and Yamaura, H. (1997). "A robust controller for manipulator driven by artificial muscle actuator," in Proceeding of the 1996 IEEE International Conference on Control Applications IEEE International Conference on Control Applications held together with IEEE International Symposium on Intelligent Control IEEE International Symposium on Computer-Aided *Contro* (Dearborn, MI), 540–45.

Camarillo, D. B., Carlson, C. R., and Salisbury, J. K. (2009). Task-space control of continuum manipulators with coupled tendon drive. *Springer Tracts Adv. Robotics*. 54, 271–280. doi: 10.1007/978-3-642-00196-3_32

Carbonell, P., Jiang, Z. P., and Repperger, D. W. (2001). "Nonlinear control of a pneumatic muscle actuator: backstepping vs. sliding-mode," in *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01) (Cat. No.01CH37204)* (Mexico), 167–72.

Cervantes, I., and Alvarez-Ramirez, J. (2001). On the PID tracking control of robot manipulators. *Syst. Control Lett.* 42, 37–46. doi: 10.1016/S0167-6911(00)00077-3

Chan, C. Y., Chong, S. H., Loh, S. L., Alias, A., and Kasdirin, H. A. (2020). Positioning control of an antagonistic pneumatic muscle actuated system using feedforward compensation with cascaded control scheme. *Int. J. Integr. Eng.* 12:70–74. doi: 10.30880/ijie.2020.12.02.008

Chan, S. W., Lilly, J. H., Repperger, D. W., and Berlin, J. E. (2003). "Fuzzy PD+I learning control for a pneumatic muscle," in *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ*'03 (St Louis, MO), 278–283.

Chiang, C.-J., and Chen, Y.-C. (2017). Neural network fuzzy sliding mode control of pneumatic muscle actuators. *Eng. Appl. Artif. Intell.* 65, 68–86. doi: 10.1016/j.engappai.2017.06.021

Chiang, C.-J., and Chen, Y. C. (2018). Incremental fuzzy sliding mode control of pneumatic muscle actuators. *Int. J. Innov. Comput. Inf. Control.* 14: 1917–1928. doi: 10.24507/ijicic.14.05.1917

Duan, B. Y., Su, Y. X., and Zheng, C. H. (2004). Nonlinear PID control of a six-DOF parallel manipulator. *IEE Proc.* 151, 95–102. doi: 10.1049/ip-cta:20030967

George Thuruthel, T., Ansari, Y., Falotico, E., and Laschi, C. (2018). Control strategies for soft robotic manipulators: a survey. *Soft Robot.* 5, 149–163. doi: 10.1089/soro.2017.0007

Giannaccini, M. E., Xiang, C., Atyabi, A., Theodoridis, T., Nefti-Meziani, S., and Davis, S. (2018). Novel design of a soft lightweight pneumatic continuum robot arm with decoupled variable stiffness and positioning. *Soft Robot.* 5, 54–70. doi: 10.1089/soro.2016.0066

Giorelli, M., Renda, F., Calisti, M., Arienti, A., Ferri, G., and Laschi, C. (2015a). Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature. *IEEE Trans. Robot.* 31, 823–834. doi: 10.1109/TRO.2015.2428511

Giorelli, M., Renda, F., Calisti, M., Arienti, A., Ferri, G., and Laschi, C. (2015b). Learning the inverse kinetics of an octopus-like manipulator in three-dimensional space. *Bioinspir. Biomimetics.* 10:35006. doi: 10.1088/1748-3190/10/3/035006

Giorelli, M., Renda, F., Ferri, G., and Laschi, C. (2013). "A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Tokyo), 5033–5039. doi: 10.1109/IROS.2013.6697084

Hannan, M. W., and Walker, I. D. (2003). Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots. *J. Robot. Syst.* 20, 45–63. doi: 10.1002/rob.10070

Kang, R., Guglielmino, E., Branson, D. T., and Caldwell, D. G. (2013). "Kinematic model and inverse control for continuum manipulators," in *2013 10th IEEE International Conference on Control and Automation (ICCA),* 1615–1620. doi: 10.1109/ICCA.2013.6565003

Kang, R., Guo, Y., Cheng, K., and Chen, L. (2014). "Design and control of a soft actuator driven by pneumatic muscles," in *2014 International Conference on Industrial Automation, Information and Communications Technology* (Bali), 26–30. doi: 10.1109/IAICT.2014.6922097

Leephakpreeda, T. (2011). Fuzzy logic based PWM control and neural controlled-variable estimation of pneumatic artificial muscle actuators. *Expert Syst. Appl.* 38, 7837–7850. doi: 10.1016/j.eswa.2010.12.120

Manti, M., Cacucciolo, V., and Cianchetti, M. (2016). Stiffening in soft robotics: a review of the state of the art. *IEEE Robot. Autom. Mag.* 23:2582718. doi: 10.1109/MRA.2016.2582718

Meng, W., Xie, S. Q., Liu, Q., Lu, C. Z., and Ai, Q. (2017). Robust iterative feedback tuning control of a compliant rehabilitation robot for repetitive ankle training. *IEEE/ASME Trans. Mechatr.* 22, 173–184. doi: 10.1109/TMECH.2016.2618771

Nakamura, T., and Shinohara, H. (2007). "Position and force control based on mathematical models of pneumatic artificial muscles reinforced by straight glass fibers," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 4361–4366. doi: 10.1109/ROBOT.2007.364151

Neppalli, S., and Jones, B. A. (2007). "Design, construction, and analysis of a continuum robot," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Diego, CA), 1503–7. doi: 10.1109/IROS.2007.4399275

Schreiber, F., Sklyarenko, Y., Schlüter, K., Schmitt, J., Rost S., and Raatz A. (2011). "Tracking control with hysteresis compensation for manipulator segments driven by pneumatic artificial muscles," in *2011 IEEE International Conference on Robotics and Biomimetics, ROBIO* (2011*)* (Phuket). doi: 10.1109/ROBIO.2011.6181721

Shen, T. K., Lee, I. F., Lin, P. H., Lin, C. I., Lin, P. C., and Shih, W. P. (2015). *"Implementation of a PID controller for a robotic leg actuated by pneumatic artificial muscles," in 2015 IFToMM World Congress Proceedings*, Taipei: IFToMM.

Su, Y. X., Sun, D., and Duan, B. Y. (2005). *Design of an enhanced nonlinear PID controller*. Mechatronics. 15, 1005–1024. doi: 10.1016/j.mechatronics.2005.03.003

Teramae, T., Ishihara, K., Babi,č J., Morimoto, J., and Oztop, E. (2018). Human-in-The-loop control and task learning for pneumatically actuated muscle based robots. *Front. Neurorobot.* 12:71. doi: 10.3389/fnbot.2018.00071

Thanh, T. D. C., and Ahn, K. K. (2006). Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network. *Mechatronics* 16, 577–587. doi: 10.1016/j.mechatronics.2006.03.011

Tondu, B., and Lopez, P. (2000). Modeling and control of McKibben artificial muscle robot actuators. *IEEE Control Syst.* 20, 15–38. doi: 10.1109/37.833638

Trivedi, D., Rahn, C. D., Kier, W. M., and Walker, I. D. (2008). Soft robotics: biological inspiration, state of the art, and future research. *Appl. Bionics. Biomech.* 5:520417. doi: 10.1155/2008/520417

Wang, B., McDaid, A., Giffney, T., Biglari-Abhari, M., and Aw, K. C. (2017). Design modelling and simulation of soft grippers using new bimorph pneumatic bending actuators. *Cogent. Eng.* 4:1285482. doi: 10.1080/23311916.2017.1285482

Yang, H. D., Greczek, B. T., and Asbeck, A. T. (2019). Modeling and analysis of a high-displacement pneumatic artificial muscle with integrated sensing. *Front. Robot.* 5:136. doi: 10.3389/frobt.2018.00136

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.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 16 November 2020
doi: 10.3389/frobt.2020.557624

Closed-Loop Control of Electro-Ribbon Actuators

Richard Suphapol Diteesawat[1,2*†], *Aaron Fishman*[1,2†], *Tim Helps*[1,2], *Majid Taghavi*[1,2] *and Jonathan Rossiter*[1,2]

[1] *Department of Engineering Mathematics, University of Bristol, Bristol, United Kingdom,* [2] *Bristol Robotics Laboratory, Bristol, United Kingdom*

Electro-ribbon actuators are lightweight, flexible, high-performance actuators for next generation soft robotics. When electrically charged, electrostatic forces cause the electrode ribbons to progressively zip together through a process called dielectrophoretic liquid zipping (DLZ), delivering contractions of more than 99% of their length. Electro-ribbon actuators exhibit pull-in instability, and this phenomenon makes them challenging to control: below the pull-in voltage threshold, actuator contraction is small, while above this threshold, increasing electrostatic forces cause the actuator to completely contract, providing a narrow contraction range for feedforward control. We show that application of a time-varying voltage profile that starts above pull-in threshold, but subsequently reduces, allows access to intermediate steady-states not accessible using traditional feed-forward control. A modified proportional-integral closed-loop controller is proposed (Boost-PI), which incorporates a variable boost voltage to temporarily elevate actuation close to, but not exceeding, the pull-in voltage threshold. This primes the actuator for zipping and drastically reduces rise time compared with a traditional PI controller. A multi-objective parameter-space approach was implemented to choose appropriate controller gains by assessing the metrics of rise time, overshoot, steady-state error, and settle time. This proposed control method addresses a key limitation of the electro-ribbon actuators, allowing the actuator to perform staircase and oscillatory control tasks. This significantly increases the range of applications which can exploit this new DLZ actuation technology.

Keywords: control, soft robotics, actuator, electrostatic, zipping, pull-in instability, electro-ribbon, dielectrophoretic liquid zipping

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Herbert Shea, École Polytechnique Fédérale de Lausanne, Switzerland
Gianluca Rizzello, Saarland University, Germany
Samuel Rosset, University of Auckland, New Zealand

****Correspondence:***
Richard Suphapol Diteesawat
richard.diteesawat@bristol.ac.uk

† These authors have contributed equally to this work

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *30 April 2020*
Accepted: *11 September 2020*
Published: *16 November 2020*

Citation:
Diteesawat RS, Fishman A, Helps T, Taghavi M and Rossiter J (2020) Closed-Loop Control of Electro-Ribbon Actuators. Front. Robot. AI 7:557624. doi: 10.3389/frobt.2020.557624

INTRODUCTION

Soft robotics has the potential to enhance dexterous robotics tasks, such as gripping and locomotion, using structures that conform to variable and sensitive environments (Schmitt et al., 2018). While soft robotic actuators offer the high stroke and force-to-weight ratio required for such tasks (Bar-Cohen, 2000; Carpi et al., 2011), the position control of rigid actuators is generally more predictable, and better suited for complex tasks such as path planning and position sensing in sufficiently structured environments (Trivedi et al., 2008). Thus, a significant challenge for soft robotics is the development of control strategies that permit fast and complex motion within changing or unpredictable environments.

Dielectrophoretic Liquid Zipping (DLZ) is a novel actuation concept whereby amplified electrostatic attraction results in high-force electrostatic zipping (Taghavi et al., 2018). In DLZ, a pair of electrodes, electrically insulated from one another, are mechanically arranged in a zipping configuration (**Figure 1**). If the electrodes are oppositely charged, a strong electric field is developed between them, inducing a strong electrostatic attractive force that causes them to progressively zip together. Although electrostatic zipping has been demonstrated previously in zipping devices (Maffli et al., 2013; Chen et al., 2014; Li et al., 2018), DLZ features a tiny droplet of liquid dielectric (e.g., silicone oil) at the zipping point(s), which considerably amplifies electrostatic force. The electrostatic force between the two electrodes is related to Maxwell pressure, $P = \varepsilon E^2$, where ε is the permittivity of the liquid dielectric and E is the electric field (Suo, 2010). The added liquid dielectric not only has a higher permittivity but also a considerably higher breakdown strength compared with air, allowing stronger fields to be sustained with associated stronger electrostatic zipping forces. Silicone oil, for example, whose respective permittivity and breakdown strength are around 2.7 and 6.7 times greater than air, theoretically implies up to 120-fold amplification of electrostatic force (Taghavi et al., 2018). While this amplification could be achieved by submerging the entire actuator in liquid dielectric, practically only a tiny droplet at each zipping point is required, since coincidentally occurring dielectrophoretic forces (which have the effect of drawing high-permittivity materials toward strong electric fields) help to retain the liquid dielectric at the zipping point (Taghavi et al., 2018).

The simplest embodiment of DLZ is the electro-ribbon actuator (**Figure 1**). Electro-ribbon actuators are compliant artificial muscles that can be made from any combination of conducting and insulating materials. Various embodiments of this type of actuator can exhibit high tension, high contraction (>99%), or high specific power equivalent to human muscle. These performance metrics make them a promising technology for Soft Robotics, where flexible, low-mass, high-performance actuators are required to deliver useful functions.

Control of soft robots is typically a challenging task due to their continuum structure and inherent compliance when interacting with the environment (Trivedi et al., 2008).

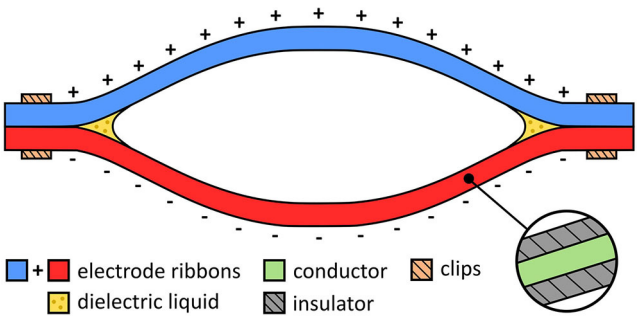


FIGURE 1 | Components of an electro-ribbon actuator.

Conventional control strategies that assume rigid joints tend to be ineffective at controlling soft robots (Rus and Tolley, 2015). Compared with traditional hard actuators, such as motors, control of soft actuators is often challenging due to their inherent compliance. Closed-loop control of dielectric elastomer actuators has been demonstrated using capacitive self-sensing (Rosset et al., 2013). Self-sensing has also been demonstrated in liquid-filled flexible fluidic actuators (Helps and Rossiter, 2018) and electrically driven HASEL (Acome et al., 2018) and Peano-HASEL (Kellaris et al., 2018) actuators, although full closed-loop control was not demonstrated. Closed-loop control of several parallel-plate electrostatic actuators, such as microelectromechanical systems (MEMS) devices, have been studied in Chu and Pister (1994), Seeger and Boser (1999), and Dong and Edwards (2010).

In this article, we investigate closed-loop control for electro-ribbon actuators. Open-loop control is limited with these actuators due to pull-in instability, resulting in an extremely small range of travel where position may be reliably controlled. However, by modulating the input voltage, a much larger region of stable positions can be achieved. We introduce a closed-loop (Boost-PI) controller, explore the effect of system gains upon output parameters, and configure gains based on a multi-objective parameter-space approach. Finally, we demonstrate system performance, including set point tracking of predetermined trajectories and sinusoidal signals, typical behaviors needed for the actuation and control of soft robots.

MATERIALS AND METHODS

The electro-ribbon actuators were made from two electrode ribbons, each of which was comprised of a 50-μm-thick, 10-cm-long, 2.5-cm-wide steel strips (1.1274 carbon steel, h+s präzisionsfolien GmbH, Germany). Each electrode ribbon was insulated using PVC tape (AT7 PVC Electrical Insulation Tape, Advance Tapes, UK). The ends of the electrode ribbons were attached to one another using custom-made plastic clips to ensure a tight zipping point (**Figure 2**). A drop of silicone oil with viscosity of 50 cSt (# 378356, Sigma-Aldrich, USA) was added to each zipping point prior to each experiment to ensure consistency. High voltage was applied to the electro-ribbon actuators using a high voltage amplifier (5HVA24-BP1, UltraVolt, USA). Inputs were controlled and data was recorded using a National Instrument device (NI USB-6343, National Instruments, USA). A laser displacement sensor (LK-G402, Keyence, Japan) was used to measure actuator displacement, by measuring the height of the suspended mass in the vertical direction at a frequency of 1,000 Hz. For closed-loop control, measured height was used as the feedback variable, with a control loop sample frequency of 32 Hz. The open-loop bandwidth of the actuator has been observed as 10 Hz (Taghavi et al., 2018), thus 32 Hz was considered sufficient.

Isotonic testing was used to investigate the controllability of an electro-ribbon actuator. A rigid acrylic frame was built for the

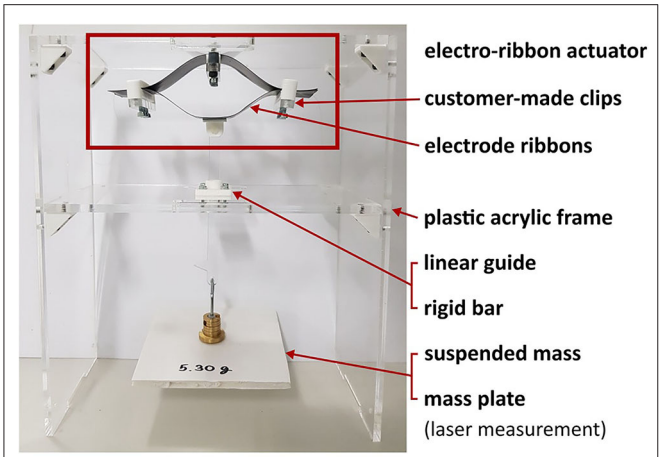


FIGURE 2 | Experimental setup of the electro-ribbon actuator for both open-loop and closed-loop controls.

experiments. The center of the upper ribbon of the electro-ribbon actuator was clamped to the top of the rigid frame (**Figure 2**). This clamp had the advantage of preventing full zipping, which can introduce temporal hysteresis due to adhesive and cohesive forces associated with the liquid dielectric (Taghavi et al., 2018). The bottom ribbon was connected to a rigid bar, prescribed to move vertically by a linear guide, ensuring symmetrical zipping of the electro-ribbon actuator. To apply load to the electro-ribbon actuator, an external mass was hung at the bottom of the rigid bar. In the following experimental results, zero height was set as the position where the two electrodes zipped such that the bottom ribbon touched the clamp, with a 5 mm gap remaining between two electrodes at the center. The initial negative height was set as the resting position of the actuator with a suspended load and without an electrostatic force.

RESULTS

Step Response

To investigate the electrical charging effects, we performed a step-response test by applying a constant voltage, $V_{constant}$, across the electrodes of the electro-ribbon actuator. In each test a constant mass was suspended from the actuator, which set its resting height. A constant voltage was applied for actuation. The vertical motion of the mass was recorded. After 10 s, the voltage was reduced to zero, and the actuator extended due to gravity, returning to its initial resting height for the next test at a higher voltage. The applied voltage was increased in 100 V increments for each test until pull-in voltage was reached. The maximum vertical displacement of the actuator achieved during 10 s of actuation at each voltage, was recorded and presented in **Figure 3A**. For example, for the actuator loaded with a constant mass of 10.18 g, when the $V_{constant}$ reached 6,300 V, pull-in instability resulted in the actuator undergoing full zipping (**Figure 3A**). In this case, pull-in voltage $V_{pull-in}$ was 6,300 V. This pull-in instability causes rapid full zipping because the generated constant electrostatic force at active moving zipping

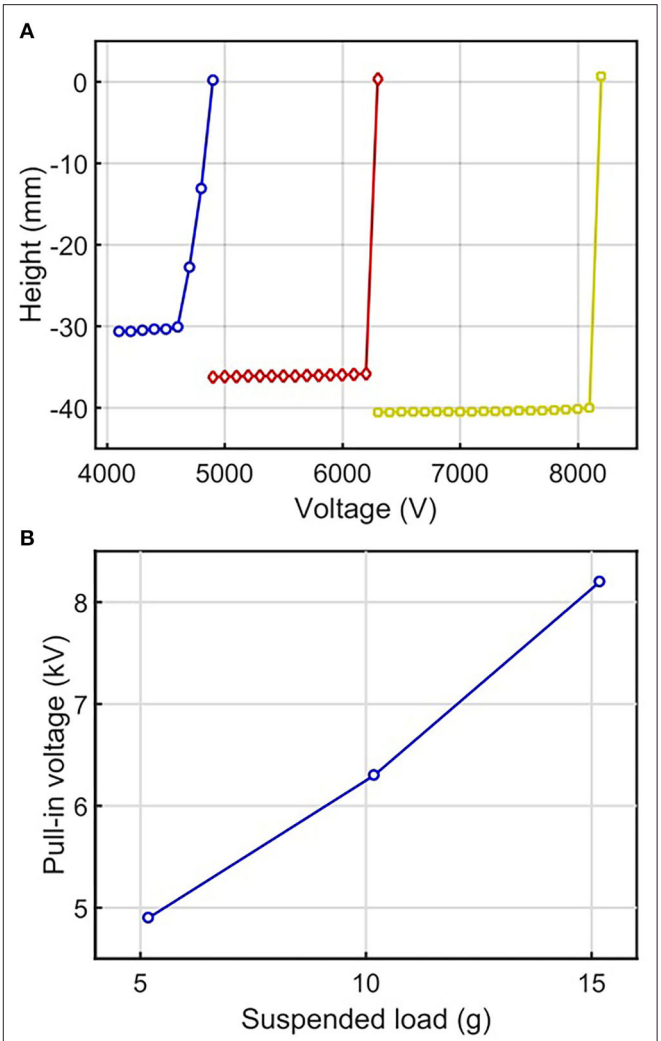


FIGURE 3 | (A) Height variation with voltage for electro-ribbon actuators with masses of 5.18 g (blue line, circular markers), 10.18 g (red line, diamond markers), and 15.18 g (yellow line, square markers). Full zipping position occurs at zero height. **(B)** The relationship between pull-in voltage and suspended load.

points consistently and increasingly overcomes the gravity force transferred from an external load, which is highest at zipping corners and decreases along the actuator to the center. As a result, when the electrostatic force at the zipping corners exceeds the gravitational load, the actuator will always fully zip. **Figure 3A** demonstrates how traditional open-loop control strategies for electro-ribbon actuators provide a very small controllable range. Since greater loads applied to the actuator require greater electrostatic force to initiate full zipping, $V_{pull-in}$ increases with an external load. The relationship between pull-in voltage and load can be shown in **Figure 3B**.

Time-Varying Voltage Profiles

A more complex approach was explored by applying a time-varying voltage profile to the actuator (**Figure 4A**). In this experiment, the actuator began to zip when applying $V \geq$

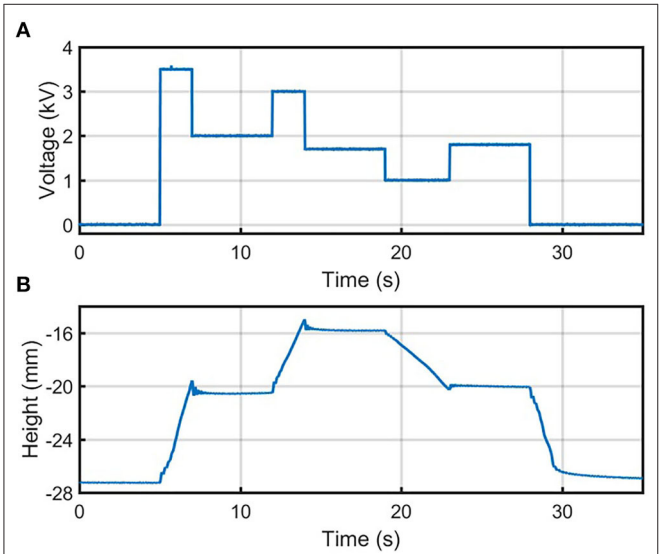


FIGURE 4 | (A) The voltage and **(B)** height output of the time-varying voltage profile under a load of 7.05 g. Full zipping position occurs at zero height.

$V_{pull-in}$. After some time, V was instantaneously reduced to a constant value below $V_{pull-in}$, which allowed the actuator to be held at a steady state height greater than the pull-in height in the previous step-response experiment. By switching V to a higher or lower value for a short time, and then setting a new constant voltage below $V_{pull-in}$, we were able to move the actuator's position to multiple heights not accessible in the step-response experiment (**Figure 4B**).

For example, as shown in **Figure 4B**, the voltage is increased from 2 to 3 kV and thus contracts the actuator by roughly 5 mm. After decreasing the voltage to 1.7 kV, this steady-state position is maintained. The existence of additional stable heights within the range at which pull-in voltage occurs is attributed to various effects not included in standard pull-in instability models. These effects include stiction forces, and fluidic forces such as surface tension. These effects can be exploited to extend the contraction range of open-loop controllers of the electro-ribbon actuators. In addition, electro-ribbon actuators exhibit voltage-displacement hysteresis due to the inverse square relationship between actuation force and displacement at a given voltage (Taghavi et al., 2018). This hysteresis has been studied in detail in Taghavi et al. (2020).

Closed-Loop Control

Having found evidence of complex non-linearities affecting actuator stability within the system, we investigated a closed-loop controlled actuation by introducing a simple proportional-integral control. When using fixed proportional and integral terms of 600 and 60 respectively and setting the voltage to initialize at $V_{pull-in}$, the actuator controllably approached to different set points while loaded different masses of 3.18, 10.18, and 15.18 g (**Figure 5**). This method allowed the electro-ribbon

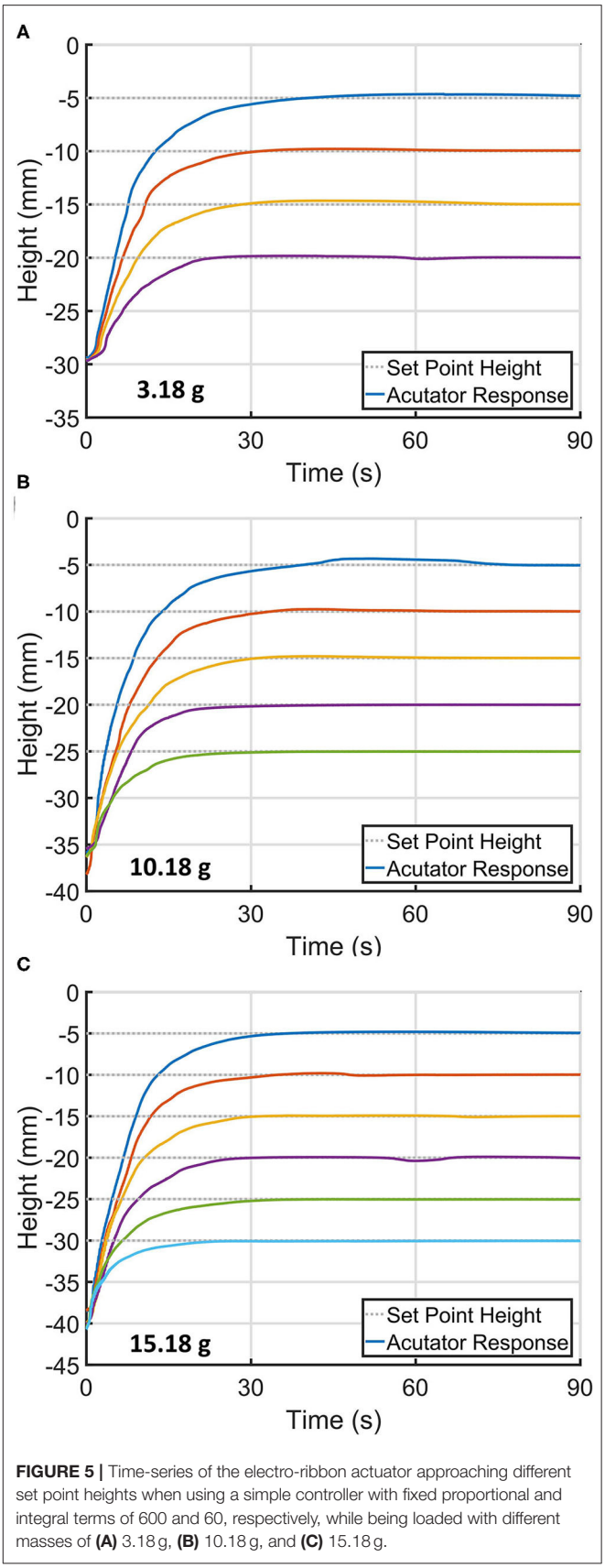


FIGURE 5 | Time-series of the electro-ribbon actuator approaching different set point heights when using a simple controller with fixed proportional and integral terms of 600 and 60, respectively, while being loaded with different masses of **(A)** 3.18 g, **(B)** 10.18 g, and **(C)** 15.18 g.

actuator to access to different intermediate heights between resting and full zipping points. However, an actuation speed decreased when set point height was set further, and the actuator suffered from long rise time to reach the desired height. Based upon these findings, we developed a modified closed-loop proportional-integrator (PI) controller, termed the Boost-PI controller. The control law took the form of the following equations:

$$E(t) = h(t) - h_s, \tag{1}$$

where E is the input error, which is the difference between the measured height h and the setpoint height h_s of the electro-ribbon actuator. The input voltage V can be derived as follows when t is time:

$$V(t) = K_p E(t) + \int K_i E(t)\, dt + V_c. \tag{2}$$

The Boost-PI controller was used to calculate the input voltage V, consisting of three parts as follows:

- K_p: a proportional term, providing a large initial voltage proportional to the error. This term acts to rapidly initiate zipping.
- K_i: an integral term, which acts to minimize steady-state error and to compensate for variable external loads.
- V_c: a constant voltage equal to 90% of $V_{pull-in}$, the voltage at which the actuator overcomes the load and begins to contract. This term acts to prime, or boost, the actuator for zipping, reducing rise time.

The voltage applied to the electro-ribbon actuator needs to exceed $V_{pull-in}$ in order to initiate zipping. If only proportional and integral terms are used (standard PI controller), the actuator experiences large integration timescales, causing a long delay to reach this critical voltage. For example, if the proportional term is much lower than $V_{pull-in}$, the integral term will take long time to accumulate until the total input voltage V reaches $V_{pull-in}$. In this regard, V_c is set as 90% of $V_{pull-in}$, ensuring the actuator is immediately almost at the point at of zipping. We also limit the maximum voltage applied to the actuator to 9,000 V, to prevent damage to the actuator or electric breakdown of the nearby air.

We used a set point height of 10 mm stroke away from the resting position to observe the performance of presented closed-loop control. **Figure 6** shows closed-loop control of an electro-ribbon actuator under an external load of 8.04 g using the Boost-PI controller. **Figure 6A** shows the proportional and integral terms of the controller, the constant voltage V_c and the input voltage. **Figure 6B** shows the actuator response, setpoint, maximum height, rise time, and settle time.

The performance in controlling the actuator was explored by varying control gains of the Boost-PI controller. We varied K_p from 0 to 1,600 with increments of 200 and K_i from 0 to 60 with increments of 20 for each K_p. V_c was set at 4,590 V (90% of $V_{pull-in}$ for an actuator loaded with a mass of 8.04 g). Step-response control tasks (as shown in **Figure 6**) were performed for different control values; the experimental results can be concluded as follows:

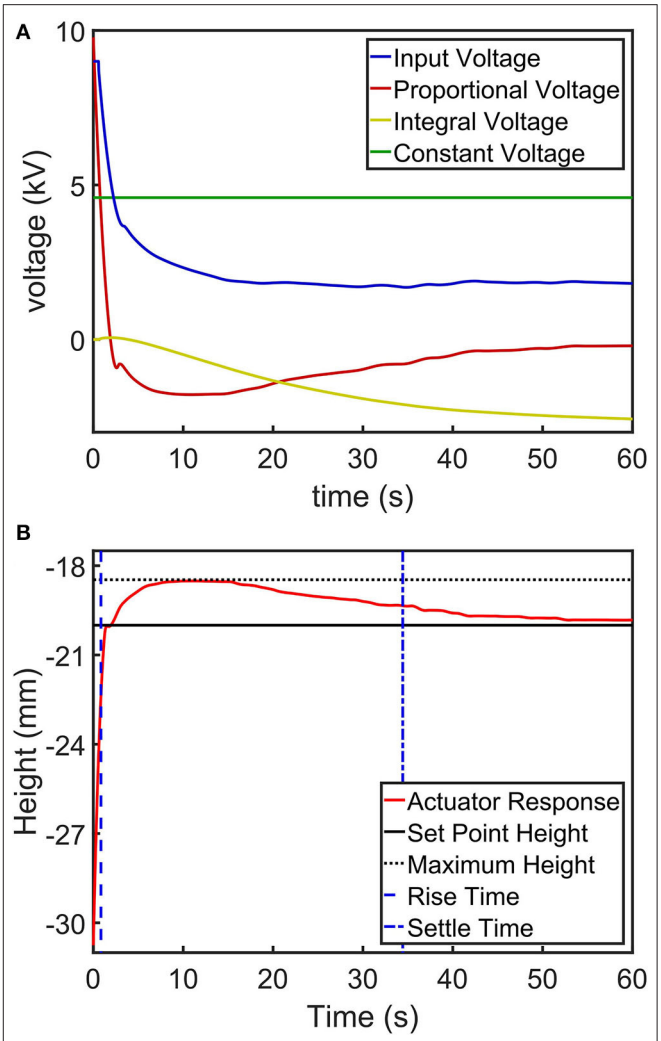


FIGURE 6 | An example step-response control task of an electro-ribbon actuator actuated using a closed-loop Boost-PI controller (K_p = 1200 and K_i = 60) while loaded with a mass of 8.04 g, showing **(A)** a controlled voltage input to the actuator (a combination of the proportional, integral and constant voltage terms of the controller) and **(B)** a height output with setpoint height, maximum height, rise time and settle time. Full zipping position occurs at zero height.

- When K_p and K_i were both equal to zero (i.e., using only V_c without a PI controller), the actuator remained at the resting height without any zipping motion since the input voltage is equal to V_c, which is less than $V_{pull-in}$.
- When $K_i = 0$ (i.e., using the controller with only K_p and V_c), when K_p was between 200 and 600, the actuator overshot the setpoint considerably, and zipped fully. With $K_p \geq 800$, the proportional term was large enough to reduce applied voltage after overshoot quickly enough to prevent full zipping. With $K_p \geq 800$, the actuator height approached the setpoint, although large steady-state error was present. Increasing K_p reduced the steady-state error but could negatively cause oscillation around the setpoint.

- When $K_p = 0$ (i.e., using the controller with only K_i and V_c), the integral term slowly increased until the sum of the integral term and V_c exceeded $V_{pull-in}$ to initialize actuation. At this point, full zipping occurred, because the integral term did not reduce applied voltage quickly enough to prevent full zipping.
- For any K_i (from 0 to 60), when K_p is 200, the actuator fully zipped (again, the proportional term was not large enough to reduce applied voltage after overshoot quickly enough to prevent full zipping). When K_p is 1400 and higher, unstable oscillations occurred.
- Using a controller with non-zero values of both K_p, K_i, and V_c allowed the actuator to converge to the setpoint at different velocities, when K_p was between 400 and 1,200 and K_i was between 20 and 60.

The performance of the Boost-PI controller was evaluated by assessing performance metrics of steady state error, overshoot, rise time and settle time as benchmarks to select Boost-PI gains (**Figure 7**), considering only the case, where the actuator converges to set point position. Steady state error is defined as the difference between steady state height and setpoint height, while overshoot is the difference between steady state height and maximum height. Rise time is the time at which current height h reaches 90% of the setpoint height h_s; settle time is the time at which h remains within 5% of the steady-state height.

According to **Figure 7**, rise time considerably decreased when $K_p \geq 600$, down to between 1 and 2 s. Overshoot significantly decreased mainly by increasing K_p from 6.01 mm ($K_p = 400$, $K_i = 20$) to 1.77 mm ($K_p = 1{,}200$, $K_i = 60$), whereas steady-state error dramatically reduced with increasing K_i although it slightly increased with increasing K_p. Settle time fluctuated across control gains, varying between 27 and 50 s with an average of 40 s. It is not possible to select gains which are ideal for all performance metrics, instead gains should be chosen that provide an appropriate compromise. In practice, the relative importance of each metric is problem-dependent and thus the Boost-PI gains should be selected according to the task at hand. This could be approached, for example, with a weighted-sum method using our performance metrics. Lower overshoot and faster rise time were prioritized for closed-loop control since lower overshoot enables higher controllable range close to full zipping position, and faster rise time increases the range of applications for this actuator. Hence, $K_p = 1{,}200$ and $K_i = 60$ were selected as task-appropriate general-use gains for this controller.

We tested the versatility of the Boost-PI controller with two setpoint tracking tasks: a staircase task and an oscillatory task (**Figure 8**). $K_p = 1{,}200$ and $K_i = 60$ were used to control the actuator for these two tasks, which were set to perform within a 20 mm stroke range from a resting position to maintain high actuation performance (±10 mm stroke from where these control

		Rise time (s)			Overshoot (mm)		
		K_i			K_i		
		20	40	60	20	40	60
K_p	400	3.767	3.583	1.636	6.013	5.852	5.083
	600	1.678	1.509	1.438	4.507	4.076	3.849
	800	1.907	1.843	1.723	2.688	2.613	2.442
	1000	1.595	1.609	1.113	2.106	2.004	1.952
	1200	1.459	1.387	1.453	1.797	2.254	1.771

		Steady-state error (mm)			Settle time (s)		
		K_i			K_i		
		20	40	60	20	40	60
K_p	400	0.278	0.120	0.074	49.618	35.378	43.177
	600	0.840	0.147	0.133	46.916	38.168	27.050
	800	1.057	0.196	0.018	46.369	44.984	33.960
	1000	1.225	0.390	0.028	38.222	44.206	42.784
	1200	1.185	0.344	0.087	33.808	39.473	37.968

FIGURE 7 | Key performance metrics of the electro-ribbon actuator when varying proportional gain K_p and integral gain K_i: rise time, overshoot, steady-state error and settle time. Color scale is used to present the performance outcomes, where green and red colors indicate the best and the worst value, respectively.

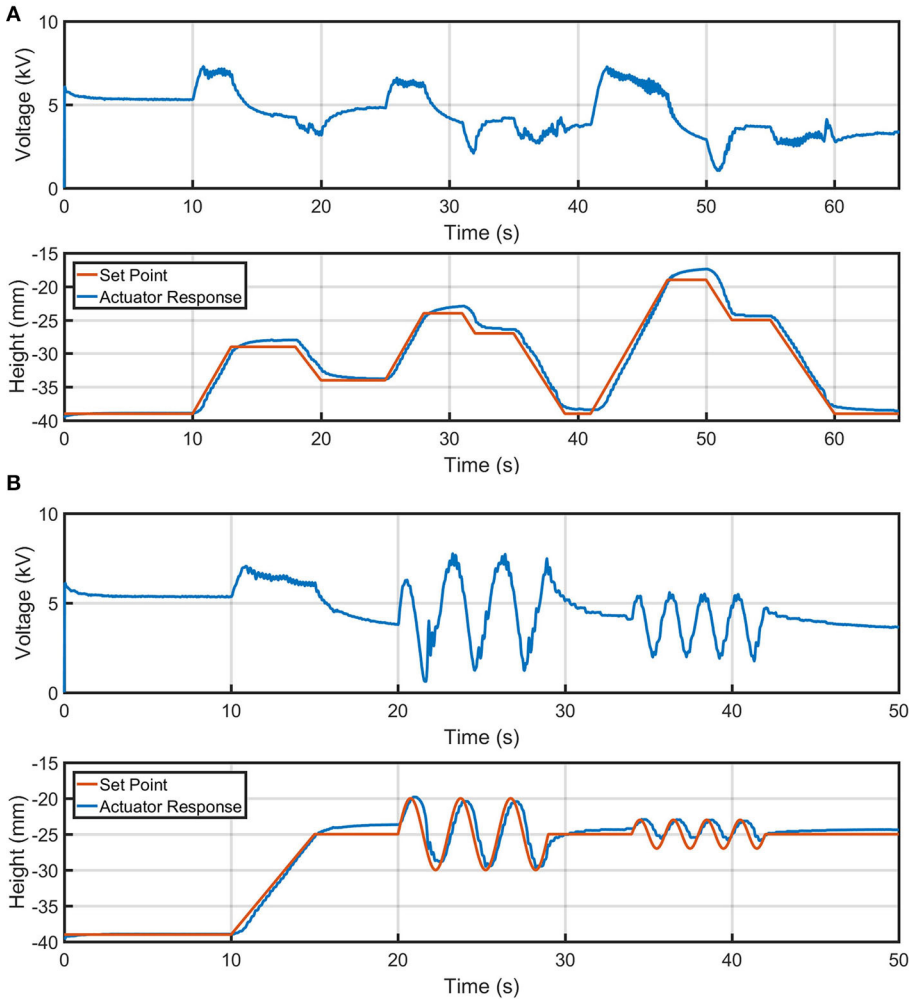


FIGURE 8 | Controlled height output of an electro-ribbon actuator actuated using a closed-loop Boost-PI controller (K_p = 1,200 and K_i = 60) while loaded with a mass of 18.04 g, for different setpoint tracking tasks: **(A)** mountain and **(B)** sinewave. Full zipping position occurs at zero height.

gains were analyzed). Comparing between the multiple cycles of sinusoidal set point control at 0.5 Hz as shown in **Figure 8B**, the maximum standard deviation for recorded actuator height between four repeated cycles was 0.5 mm. Although complex non-linearities clearly exist for the electro-ribbon actuator, it can be effectively controlled using the presented Boost-PI controller.

DISCUSSION

In this article, we explored an approach to control the contraction of an electro-ribbon actuator, which exhibits complex non-linear behavior. Initially, the actuator was tested by applying increasing static voltages to determine the $V_{pull-in}$, where the actuator experienced pull-in instability and performed full zipping. $V_{pull-in}$ depends on the external load; higher loads induce larger extensions and require higher $V_{pull-in}$. If voltages below $V_{pull-in}$ are applied, there exists a very narrow open-loop-controllable range of contractions. Application of a time-varying voltage approach, that is initially above $V_{pull-in}$ but subsequently steps down to a lower voltage, enables a much wider range of accessible steady-state contractions. However, this approach is challenging because the steady-state contraction reached depends on not only on the applied voltage profile but also the previous steady-state contraction.

We modified a closed-loop PI controller—the Boost-PI controller—with an additional constant voltage term (V_c) to control the actuator. V_c reduces the time taken for the integral term to ramp-up to the voltage required to initialize zipping, resulting in lower rise time. It ensured the controlled voltage was close to $V_{pull-in}$, which is dependent to a suspended load (**Figure 3B**). The Boost-PI controller was studied by varying proportional and integral terms while setting V_c to 90% of $V_{pull-in}$.

To select appropriate Boost-PI gains, we implemented a multi-objective parameter-space approach, analyzing rise time, overshoot, steady-state error and settle time of the actuator response as benchmarks. As a result, K_p = 1,200 and K_i = 600 were selected since the resulting actuator was capable of

fast rise time (1.45 s or 5.89 mm/s), low overshoot (1.77 mm or 18.6%) and acceptable settle time (38.0 s) for a 9.49 mm setpoint distance step-response task. The Boost-PI controller was used to control an electro-ribbon actuator to perform different control tasks: a staircase and oscillatory task, showing versatility and controllability of electro-ribbon actuators.

The contraction rate of the electro-ribbon actuator increases with increasing supplied voltage (Taghavi et al., 2018). In contrast, its extension rate increases with decreasing supplied voltage and is lower than contraction due to stiction of dielectric liquid and surface tension between two electrodes. Applying high voltage results in high contraction speed but can result in full zipping. Using the presented PI controller enables the actuation speed up to 8.7 mm/s when using $K_p = 1{,}000$ and $K_i = 60$ and holds the actuator at stable height.

While our Boost-PI method demonstrates controllability of the electro-ribbon actuators, we note several limitations based on load, actuator length and the current height. At higher loads, increasing gravitational forces reduce the acceleration of the actuator when traveling upward (against gravity) and increase acceleration when traveling downward (with gravity) and would likely require retuning of controller gains. Furthermore, the sensitivity to variations in load increases when decreasing the bending stiffness of the actuator (a long beam is more sensitive due to the longer moment arm). An additional limitation of Boost-PI is set-point sensitivity. While we show good performance over a large range of travel for our actuator, electrostatic forces increasingly affect controller performance close to full zipping position.

To address these limitations, the gains could be configured to automatically respond to changes in load and height. While height dependent gains could be developed using feedback already present in the system, load dependency could be implemented with a load cell. This approach would allow electro-ribbon actuators to perform over a wider range of loads and set-points, decrease settle-time and mitigate the need to manually reconfigure gains. The electro-ribbon actuator was observed to access the minimum distance of 5 mm away from full zipping position (**Figure 5**). Automatically tuning control gains based on the current actuator shape could enable the actuator to access a smaller distance closer to the full zipping position.

Alternatively, to improve the actuator to counteract pull-in instability, the controllable range could be increased by implementing a direct charge control strategy that actively controls the level of electrostatic charge rather than voltage, as has been done for other MEMS devices (Bochobza-Degani et al., 2003; Zhang et al., 2014). Due to the novelty of electro-ribbon actuators, analytical models are not yet available, therefore we investigated the performance of our controller using a multi-objective parameter-space approach. We plan to use a model-based approach to control electro-ribbon actuators in future work.

Traditional closed-loop control using self-sensing has been demonstrated in electro-ribbon actuators over a small displacement range (Bluett et al., 2020). Using the capacitance of the actuator measured by a self-sensing unit, which increases with zipping, as a feedback variable for the proposed Boost-PI controller could result in a controlled electro-ribbon actuator without additional sensors, as required for many soft robotics applications.

The demonstrated Boost-PI controller enables closed-loop, high-accuracy, high-working-range displacement control of electro-ribbon actuators. This addresses one limitation of the electro-ribbon actuators and considerably extends the range of applications for this type of DLZ actuator, allowing it to be included in a wide range of soft robotic systems including wearables assist devices, autonomous rescue robots and soft robots for space exploration.

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 at: University of Bristol Research Data Repository (https://data.bris.ac.uk/data) at https://doi.org/10.5523/bris.2sv1xtrsmhoro2qu3io2u244ig.

AUTHOR CONTRIBUTIONS

RD and AF contributed equally throughout developing the presented controller, designing and performing experiments, and collecting and analyzing the data. TH, MT, and JR provided advices along the research project. All authors wrote the manuscript, read, and approved the submitted version.

FUNDING

This research was supported by Royal Society Grant (TA\R1\170060) and EPSRC Impact Acceleration Funding. RD was supported by EPSRC Centre for Doctoral Training in Future Autonomous and Robotic Systems (FARSCOPE, grant EP/L015293/1) and EPSRC grant EP/S021795/1. AF was supported by Royal Society Grant (TA\R1\170060), EPSRC Impact Acceleration Funding, and The James Dyson Foundation. TH was funded by the Royal Academy of Engineering and the Office of the Chief Science Adviser for National Security under the UK Intelligence Community Postdoctoral Fellowship Programme. MT was supported by EP/R02961X/1. JR was supported by EPSRC grants EP/L015293/1, EP/M020460/1, EP/S026096/1, P/S021795/1, and EP/R02961X/1 and by the Royal Academy of Engineering as Chair in Emerging Technologies.

REFERENCES

Acome, E., Mitchell, S. K., Morrissey, T. G., Emmett, M. B., Benjamin, C., King, M., et al. (2018). Hydraulically amplified self-healing electrostatic actuators with muscle-like performance. *Science* 359, 61–65. doi: 10.1126/science.aao6139

Bar-Cohen, Y. (2000). "Electroactive polymers as artificial muscles: capabilities, potentials and challenges," in *Robotics 2000* (Reston, VA: American Society of Civil Engineers), 188–196. Available online at: https://ascelibrary.org/doi/abs/10.1061/40476(299)24

Bluett, S., Helps, T., Taghavi, M., and Rossiter, J. (2020). Self-sensing Electro-ribbon Actuators. *IEEE Robot. Autom. Lett*. 5, 3931–3936. doi: 10.1109/LRA.2020.2983677

Bochobza-Degani, O., Elata, D., and Nemirovsky, Y. (2003). A general relation between the ranges of stability of electrostatic actuators under charge or voltage control. *Appl. Phys. Lett*. 82, 302–304. doi: 10.1063/1.1536251

Carpi, F., Kornbluh, R., Sommer-Larsen, P., and Alici, G. (2011). Electroactive polymer actuators as artificial muscles: Are they ready for bioinspired applications? *Bioinspir. Biomim*. 6:045006. doi: 10.1088/1748-3182/6/4/045006

Chen, A. S., Zhu, H., Li, Y., Hu, L., and Bergbreiter, S. (2014). "A paper-based electrostatic zipper actuator for printable robots," in *2014 IEEE International Conference on Robotics and Automation* (Hong Kong: IEEE), 5038–5043. doi: 10.1109/ICRA.2014.6907597

Chu, P. B., and Pister, S. J. (1994). "Analysis of closed-loop control of parallel-plate electrostatic microgrippers," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation* (San Diego, CA: IEEE Comput. Soc. Press), 820–825.

Dong, L., and Edwards, J. (2010). "Closed-loop voltage control of a parallel-plate MEMS electrostatic actuator," in *Proceedings of the 2010 American Control Conference* (Baltimore, MD: IEEE), 3409–3414.

Helps, T., and Rossiter, J. (2018). Proprioceptive flexible fluidic actuators using conductive working fluids. *Soft Robot*. 5, 175–189. doi: 10.1089/soro.2017.0012

Kellaris, N., Venkata, V. G., Smith, G. M., Mitchell, S. K., and Keplinger, C. (2018). Peano-HASEL actuators: muscle-mimetic, electrohydraulic transducers that linearly contract on activation. *Sci. Robot*. 3:eaar3276. doi: 10.1126/scirobotics.aar3276

Li, J., Godaba, H., Zhang, Z. Q., Foo, C. C., and Zhu, J. (2018). A soft active origami robot. *Extrem. Mech. Lett*. 24, 30–37. doi: 10.1016/j.eml.2018.08.004

Maffli, L., Rosset, S., and Shea, H. R. (2013). Zipping dielectric elastomer actuators: Characterization, design and modeling. *Smart Mater. Struct*. 22:104013. doi: 10.1088/0964-1726/22/10/104013

Rosset, S., O'Brien, B. M., Gisby, T., Xu, D., Shea, H. R., and Anderson, I. A. (2013). Self-sensing dielectric elastomer actuators in closed-loop operation. *Smart Mater. Struct*. 22:104018. doi: 10.1088/0964-1726/22/10/104018

Rus, D., and Tolley, M. T. (2015). Design, fabrication and control of soft robots. *Nature* 521, 467–475. doi: 10.1038/nature14543

Schmitt, F., Piccin, O., Barbé, L., and Bayle, B. (2018). Soft robots manufacturing: a review. *Front. Robot. AI* 5:84. doi: 10.3389/frobt.2018.00084

Seeger, J. I., and Boser, B. E. (1999). Dynamics and control of parallel-plate actuators beyond the electrostatic instability. *Transducers* 99, 474–477.

Suo, Z. (2010). Theory of dielectric elastomers. *Acta Mech. Solida Sin*. 23, 549–578. doi: 10.1016/S0894-9166(11)60004-9

Taghavi, M., Helps, T., and Rossiter, J. (2018). Electro-ribbon actuators and electro-origami robots. *Sci. Robot*. 3:eaau9795. doi: 10.1126/scirobotics.aau9795

Taghavi, M., Helps, T., and Rossiter, J. (2020). "Characterisation of self-locking high-contraction electro-ribbon actuators", in *In 2020 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE), 5856–5861.

Trivedi, D., Rahn, C. D., Kier, W. M., and Walker, I. D. (2008). Soft robotics: Biological inspiration, state of the art, and future research. *Appl. Bionics Biomech*. 5, 99–117. doi: 10.1155/2008/520417

Zhang, W.-M., Yan, H., Peng, Z.-K., and Meng, G. (2014). Electrostatic pull-in instability in MEMS/NEMS: a review. *Sens. Actuat. A Phys*. 214, 187–218. doi: 10.1016/j.sna.2014.04.025

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.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 27 November 2020
doi: 10.3389/frobt.2020.586216

Design of a High-Speed Prosthetic Finger Driven by Peano-HASEL Actuators

Zachary Yoder[1], Nicholas Kellaris[1,2], Christina Chase-Markopoulou[1,3], Devon Ricken[4], Shane K. Mitchell[1], Madison B. Emmett[1], Richard F. ff. Weir[3], Jacob Segil[3,4] and Christoph Keplinger[1,2]**

[1] Paul M. Rady Department of Mechanical Engineering, University of Colorado Boulder, Boulder, CO, United States, [2] Materials Science and Engineering Program, University of Colorado Boulder, Boulder, CO, United States, [3] Biomechatronics Development Laboratory, Rocky Mountain Regional VA Medical Center, Aurora, CO, United States, [4] Engineering Plus Program, University of Colorado Boulder, Boulder, CO, United States

Current designs of powered prosthetic limbs are limited by the nearly exclusive use of DC motor technology. Soft actuators promise new design freedom to create prosthetic limbs which more closely mimic intact neuromuscular systems and improve the capabilities of prosthetic users. This work evaluates the performance of a hydraulically amplified self-healing electrostatic (HASEL) soft actuator for use in a prosthetic hand. We compare a linearly-contracting HASEL actuator, termed a Peano-HASEL, to an existing actuator (DC motor) when driving a prosthetic finger like those utilized in multi-functional prosthetic hands. A kinematic model of the prosthetic finger is developed and validated, and is used to customize a prosthetic finger that is tuned to complement the force-strain characteristics of the Peano-HASEL actuators. An analytical model is used to inform the design of an improved Peano-HASEL actuator with the goal of increasing the fingertip pinch force of the prosthetic finger. When compared to a weight-matched DC motor actuator, the Peano-HASEL and custom finger is 10.6 times faster, has 11.1 times higher bandwidth, and consumes 8.7 times less electrical energy to grasp. It reaches 91% of the maximum range of motion of the original finger. However, the DC motor actuator produces 10 times the fingertip force at a relevant grip position. In this body of work, we present ways to further increase the force output of the Peano-HASEL driven prosthetic finger system, and discuss the significance of the unique properties of Peano-HASELs when applied to the field of upper-limb prosthetic design. This approach toward clinically-relevant actuator performance paired with a substantially different form-factor compared to DC motors presents new opportunities to advance the field of prosthetic limb design.

Keywords: prosthesis, prosthetic hand, HASEL, electrohydraulic actuator, soft robotics, bioinspired, modeling

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Wenwei Yu, Chiba University, Japan
Lucas Gerez, The University of Auckland, New Zealand
R. A. R. C Gopura, University of Moratuwa, Sri Lanka

****Correspondence:***
Jacob Segil
Jacob.Segil@colorado.edu
Christoph Keplinger
Christoph.Keplinger@colorado.edu

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *22 July 2020*
Accepted: *29 October 2020*
Published: *27 November 2020*

Citation:
Yoder Z, Kellaris N, Chase-Markopoulou C, Ricken D, Mitchell SK, Emmett MB, Weir RF ff, Segil J and Keplinger C (2020) Design of a High-Speed Prosthetic Finger Driven by Peano-HASEL Actuators. Front. Robot. AI 7:586216. doi: 10.3389/frobt.2020.586216

INTRODUCTION

The field of upper limb prosthetic design seeks to recreate what was lost after amputation. In order to accomplish this feat, prosthetic devices require compact, stable, and clinically robust materials which integrate actuation to interact with the external environment (Childress and Weir, 2004). These types of actuators are more important now, in light of recent progress across a wide range of related fields like neural interfaces and osseointegration (Ortiz-Catalan et al., 2014; Tan et al., 2015).

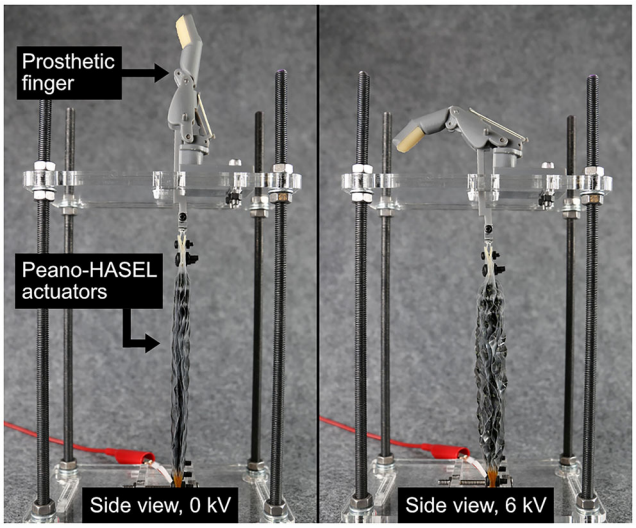


FIGURE 1 | An array of Peano-HASEL actuators, a new type of high-performance, electrohydraulic artificial muscle, driving a custom prosthetic finger. The actuators linearly contract under an applied voltage of 6 kV, which causes flexion of the prosthetic finger.

However, a remnant actuator technology—the DC electric motor—has been used for generations and has constrained the design of prosthetic devices. Here we investigate if hydraulically amplified self-healing electrostatic (HASEL) actuators, a new type of high-performance artificial muscles, will be able to improve upon the existing DC motor to facilitate the creation of lifelike prosthetic limbs with enhanced functionality. The proposed design can be seen in **Figure 1**.

The first use of DC motors in prosthetic limbs occurred in the 1940s and 1950s (Childress, 1985). These systems required substantial development before an actuated prosthesis was able to withstand the rigors of everyday use. In 1968 at Northwestern University, the first self-contained, externally-powered proportional myoelectric prosthetic limb was fit by Childress and others (Childress, 1985). Since then, miniaturization of motors, their associated electronics, and an increase in battery energy density caused rapid development of single degree-of-freedom prehensors/hands and subsequently multi-functional hands (Childress, 1985; Childress and Weir, 2004; Belter et al., 2013).

The advantages of multi-functional prosthetic hands include: the ability to create unique grasps/postures, production of clinically-appropriate forces/speeds, and an anatomically appropriate structure with all five digits. However, the use of DC motors has also constrained the device in ways that have caused disappointment among upper limb amputees. Biddiss et al. outlined the design priorities among upper limb amputees and highlighted the primary concern among powered prosthetic hand users: distribution of weight (Biddiss et al., 2007). Other design priorities listed in the top ten among powered prosthetic hand users also directly relate to the use of DC motors including cost, heat, reliability, and size. In all cases, the use of DC motors in the distal elements of a prosthetic system causes numerous concerns among upper-limb amputees such as weight, heat, reliability, and mechanical compliance (Cordella et al., 2016). Many of these concerns have not been addressed even with updated releases of popular commercial hands because the same fundamental element (DC motor) is still being used.

In multifunctional prosthetic hands, the location of the motor defines the design of the prosthetic fingers and thumb. Commercially available devices integrate the motor in the digit (iLimb by Touch Bionics, Vincent by Vincent Systems) or the palm (Bebionic by RSL Steeper, Michelangelo by Otto Bock) (Belter et al., 2013). The location of the DC motor then informs the type of transmission necessary for the digit. Various transmission designs are used throughout the industry including planetary gearheads, spur gear trains, ball screws, and other custom devices (Belter et al., 2013); new approaches that integrate DC motors with soft transmissions have been explored in academia (O'Brien et al., 2018). Finally, the kinematic design of the digit translates the force/torque produced by the actuator to the force/displacement of the digit. Typical kinematic designs of the commercially available prosthetic fingers include two phalanges where the distal interphalangeal (DIP) joint is fixed, and the metacarpal phalangeal (MCP) and proximal interphalangeal (PIP) joints are free to rotate.

In several thorough reviews, (Cura et al., 2003; Biddiss and Chau, 2008; Controzzi et al., 2014) prototype actuators with varied principles of operation are studied for use in prosthetic hands. Pneumatics, hydraulics, shape memory allows, and dielectric elastomer actuators (DEAs) have been studied, but have not been translated into clinical solutions due to difficulties with their implementation as a stable clinical system. Biddiss and Chau (2008) detail the challenges and opportunities of DEAs for use in upper limb prostheses and highlight several outstanding challenges including: unreliable temporal control, insufficient force production, and lack of anthropomorphic size/weight. While numerous artificial muscles have been proposed or explored as actuators in prostheses and wearable assistive devices, all have pitfalls in either strength, speed, reliability, complexity, and/or controllability which impedes their use outside of a laboratory setting (Biddiss et al., 2007; Park et al., 2014; Wu et al., 2015). Therefore, traditional robotic components like gears and electric motors are still the preferred method of linear actuation for prosthetic devices. Biddiss and Chau's thorough review was published in 2008 with the request to reevaluate "soft actuators" in the future.

Over 10 years later, the field of soft actuators has dramatically advanced, as part of a larger push toward "soft robotics" (Kim et al., 2013). Soft robotics incorporates compliant, lightweight, and multifunctional components into machines to mimic the adaptability and robustness of biological organisms. Biomimetic actuators are seen as desirable since their compliance should translate to robustness toward external loading events during everyday use in the field. However, the capabilities of soft robots continue to be limited by the lack of soft actuators with sufficient all-around performance in areas such as force production, speed, and efficiency.

A new class of soft actuators, or artificial muscles, termed hydraulically amplified self-healing electrostatic (HASEL)

actuators (Acome et al., 2018; Kellaris et al., 2018; Mitchell et al., 2019; Wang et al., 2020), merge the design freedom of soft fluidic actuators (Polygerinos et al., 2017) with the biomimetic, muscle-like performance and portability of dielectric elastomer actuators (Anderson et al., 2012; Koh et al., 2017; Duduta et al., 2019; Ji et al., 2019). One type of HASEL, termed the Peano-HASEL actuator, is depicted in **Figure 2** (Kellaris et al., 2018). The actuator consists of a flexible, yet inextensible polymer shell which is filled with a liquid dielectric and partially covered by electrodes. When a voltage is applied across the electrodes, an electrostatic force causes the electrodes to controllably zip together, thereby forcing the fluid into the volume of the shell which is not covered by the electrodes. This local displacement of the fluid causes the cross-section of the uncovered portion of the shell to change from a flatter cross-section to a more circular one. Since the shell is inextensible, this shape change results in a linear contraction of the actuator (Kellaris et al., 2018). Peano-HASEL actuators offer advantages over existing artificial muscles in that they feature a voltage-controlled linear contraction without the need for rigid components. Peano-HASELs have been shown to achieve muscle-like power densities of 160 W/kg and are capable of self-sensing their deformation via built-in capacitive sensing (Kellaris et al., 2018). In contrast to soft fluidic actuators, Peano-HASELs locally redistribute a hydraulic fluid, which reduces viscous losses [HASEL actuators have shown full-cycle efficiencies of 20% (Acome et al., 2018; Mitchell et al., 2019)] and enables high-speed operation with cut-off frequencies above 50 Hz and strain rates over 800%/s (Kellaris et al., 2018).

To date, the fundamental benefits and drawbacks of using electrohydraulic Peano-HASEL actuators to drive prosthetic fingers have not been identified by a systematic experimental study that compares performance to traditionally used DC motor actuators. This work describes a first attempt to integrate a Peano-HASEL actuator into a prosthetic device with the goal of informing future development of multifunctional prosthetic hands. A direct comparison between an existing Peano-HASEL and a commercially available DC motor actuator is presented. A kinematic model of the prosthetic finger is developed and used to modify the four-bar linkage used in the prosthetic finger design to better suit the characteristics of the Peano-HASEL actuators. An analytical model of the Peano-HASEL is then used to inform design modifications to the Peano-HASEL actuators in order to improve the force output while maintaining similar weight. The result is a system with promising force output over a wide range of flexion angles that is capable of both controllable and rapid response (**Supplementary Video 1**). We discuss the significance of the unique properties of Peano-HASELs in the context of prosthetic limb design.

MATERIALS AND METHODS

Commercially Available Prosthetic Hand and DC Motor

The Bebionic v2 Prosthetic Hand (RSL Steeper) is a widely available multi-functional prosthetic hand (2011). The hand has five degrees-of-freedom (DoF, individual actuators for each digit and thumb) and a passively positionable thumb abduction joint. Each digit is underactuated, meaning a single input controls the position of both the metacarpal phalangeal (MCP) joint and the proximal interphalangeal (PIP) joint through a four-bar linkage. A linear displacement of the pin in a slot in the proximal phalange causes a coupled flexion of the MCP and PIP joints. The finger is compliant at both the MCP and PIP joints in that external loads can collapse the digit in flexion. A torsional spring is located in the PIP joint to cause passive extension while the pin within the slot is pulled by the actuator to cause powered flexion [see Belter et al. (2013) for further discussion].

The actuator for each digit in the Bebionic consists of a DC motor, a planetary gearhead, a spur gear train, and a lead screw. The motor is a 1,524 Faulhaber DC motor with a 14:1 plantary gearhead and an IE2-16 motor encoder. The output of the planetary transmission is a plastic spur gear pair from Reliance Precision Mechatronics. This linear drive causes a nut to translate vertically and interfaces with the digit through the pin in the curved slot. The linear drive is a non-backdrivable element, ensuring that electric power can be turned off while maintaining a stable grasp. As used, the DC motor and necessary transmission components weigh a total of 37 grams. The DC motor itself weighs 18 grams; the array of transmission components make up roughly 50% of the entire weight of the actuating system.

The Bebionic v2 Prosthetic Hand was chosen as a basis for this design effort because it is considered a standard of care device in clinical care of people with upper limb amputation. The Bebionic contains well-founded electromechanical design elements including motors for each digit, kinematic linkages to couple multiple joints together, and the production of clinically viable forces and speeds from each digit. These standards of prosthetic hand design serve as a baseline to make comparisons and draw conclusions on the fundamental trade-offs when using the Peano-HASEL actuators instead of DC motor actuators.

Design and Fabrication of Peano-HASEL Actuators

A thorough description of the Peano-HASEL actuator is provided in Kellaris et al. (2018) and basic operating principles are depicted in **Figure 2A**. Here we briefly describe the actuator designs used in this experimental work. We refer to multiple Peano-HASEL actuators combined in parallel as stacks; two types of actuator stacks were manufactured for this work. The first stack consisted of four Peano-HASEL actuators, each with six pouches in series. Each pouch was 5 cm wide by 2 cm high, and the electrodes were 5 cm wide by 1 cm high. Eighteen-micrometer-thick biaxially oriented polypropylene (BOPP) film (70 gauge, 5020 film, Multi-Plastics) was used as the dielectric shell for this set of actuators. The second stack consisted of seven Peano-HASEL actuators, each with 12 pouches in series as seen in **Figures 2B,C**. Each pouch was 4 cm wide by 1 cm high, and the electrodes were 4 cm wide by 0.5 cm high. 12-μm-thick Mylar film (Mylar 850H, DuPont Teijin) was used for this set of actuators.

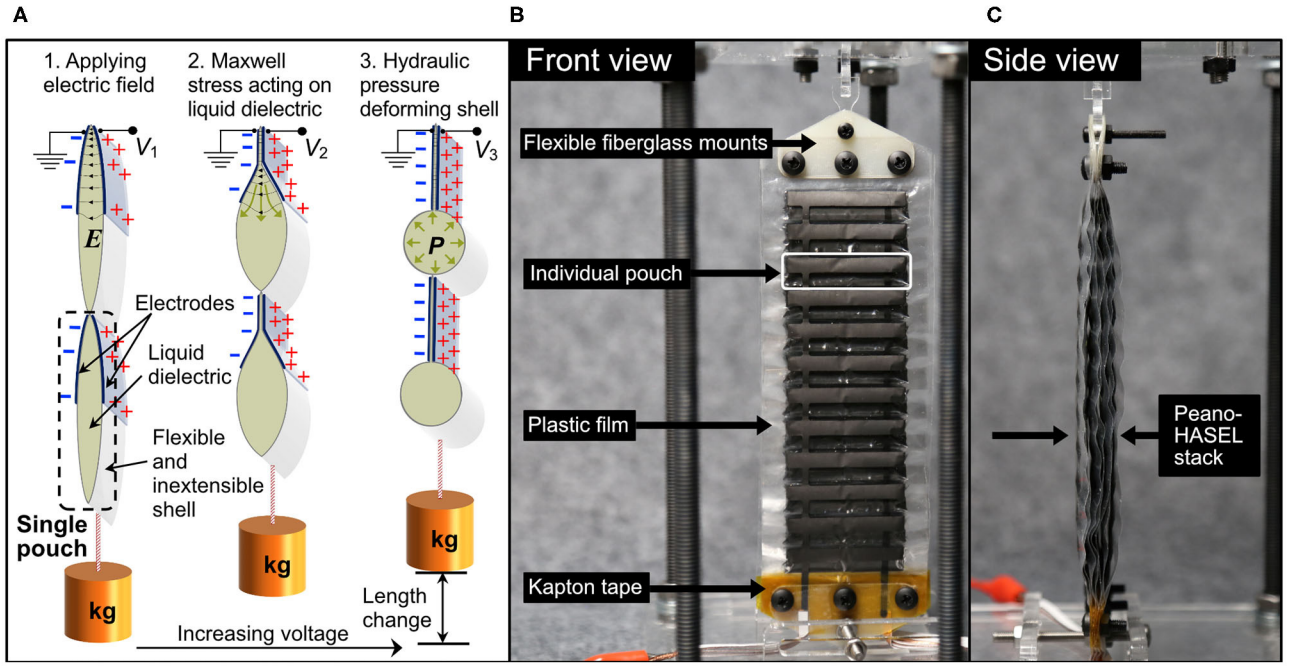


FIGURE 2 | (A) Schematic of a Peano-HASEL actuator showing basic structure and principles of operation. When voltage is applied, the resulting electric field *E* draws the electrodes together and locally displaces the liquid dielectric. The resulting hydraulic pressure *P* deforms the inextensible thermoplastic shell and causes linear contraction. **(B,C)** Photos of a stack of 7 Peano-HASEL actuators, with ancillary components labeled.

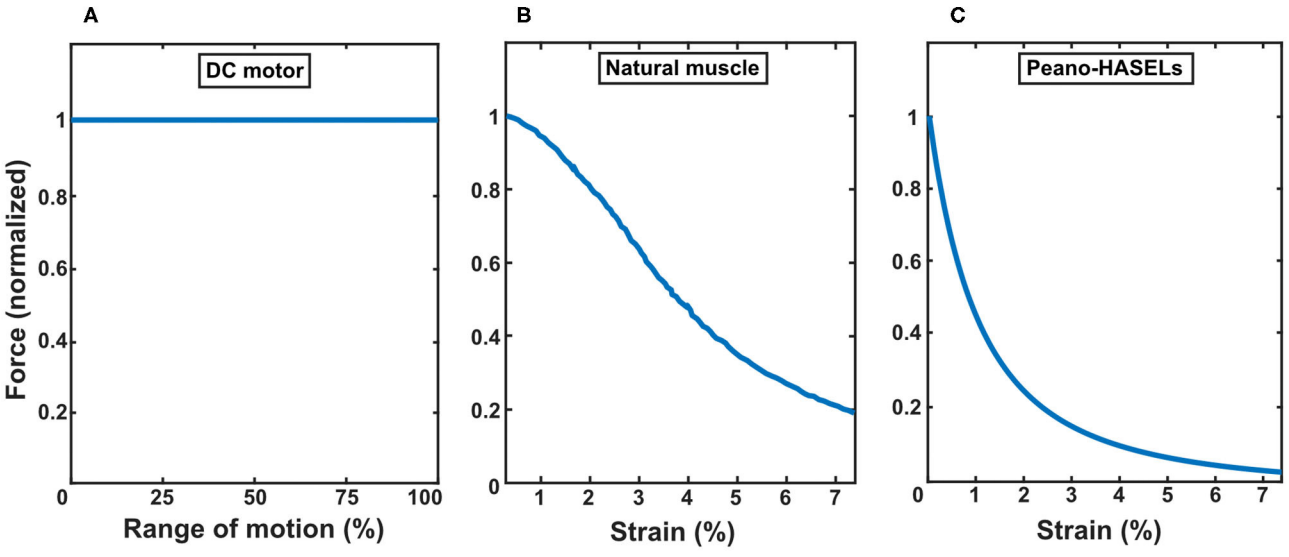


FIGURE 3 | Comparing characteristics of force-displacement curves for three types of actuators: **(A)** DC motor, **(B)** natural muscle [derived from a study of tuna muscle (Shadwick and Syme, 2008)], and **(C)** Peano-HASELs. The force profile of each actuator is normalized to its blocked force. The Peano-HASEL actuators demonstrate behavior more closely resembling natural muscle than the DC motor.

The Mylar actuators were designed for improved force output vs. the BOPP stack, based on previous work by Kellaris et al. (2019) as described in detail in Sections Kinematic Model for Prosthetic Finger/Actuator System and Improving Actuator Force Output. The pouches were fabricated following the process described in previous work (Mitchell et al., 2019). Both sets of actuators used conductive carbon ink for the electrodes (CI-2051, Engineering Materials Systems, Inc) and

were filled with a dielectric transformer oil (Cargill Envirotemp FR3). Both stacks were mounted to laser-cut 0.125-mm-thick FR-4 fiberglass mounts using adhesive transfer tape (3M 924 tape). On the end closest to the high voltage leads, 0.001" thick Kapton tape (Dupont) was placed in between dielectric film layers in the region contained by the FR-4 fiberglass mounts based on observations that an additional insulating layer decreased the likelihood of dielectric breakdown in that region. These ancillary components are depicted in **Figure 2B**. The stacks were designed to replace the DC motor and entire transmission system; therefore, the weight of the stacks plus ancillary components represent the weight of the system it was designed to replace. The BOPP actuator system weighed 43.6 grams while the Mylar version weighed 38.8 grams; this weight includes all components necessary for mounting and connecting the stacks to the prosthetic finger. The difference in weight between the Peano-HASEL stacks and the original DC motor system represented the closest possible weight match using existing Peano-HASEL architecture.

Kinematic Model for Prosthetic Finger/Actuator System

Peano-HASELs provide a high blocking force that decreases over their linear stroke similar to the behavior of natural muscle. This behavior is unlike DC motors which produce torque that is proportional to the current provided and independent of the rotational position of the actuator output. A comparison of the force output of these three types of actuators can be seen in **Figure 3** (Shadwick and Syme, 2008). The significant difference in actuator behavior drives the desire for an accurate model of the finger/actuator system that will allow us to study the changes in system behavior as various parameters, such as actuator force output, are modified. To appropriately integrate Peano-HASEL actuators with the prosthetic finger system, we developed a kinematic model to describe the prosthetic finger and actuator system.

A kinematic system of an eight-bar linkage, which was derived by Murali et al. (2019) was adapted in this work to suit the four-bar linkage system used in the Bebionic v2 Prosthetic Hand (RSL Steeper). **Figure 4** shows a crank-rocker system that is composed of binary linkages r_1, d_1, l_1, r_2, d_2, d_3 (Norton, 2000). r_{pull} represents the location from which the actuators (DC motor or Peano-HASELs) pull. φ_1, φ_2, φ_3, φ_4 are fixed, meaning they do not change value throughout flexion.

The full kinematic model was developed using several equations derived by Norton (2000) and, in context of prosthetic finger design, in Murali et al. (2019). **Figure 5** details how various parameters of interest behave over the MCP joint (θ_1) flexion range which is constrained from 0 to 90°. The following derivation explains how these parameters of interest were determined.

First, link length l_1 is obtained based on known initial conditions using Equation 1 (Murali et al., 2019).

$$l_1 = \sqrt{(d_1 \cos(\theta_1) + r_2 \cos(\theta_3 + \varphi_3) - r_1 \cos(\varphi_1))^2 + (d_1 \sin(\theta_1) + r_2 \sin(\theta_3 + \varphi_3) - r_1 \sin(\varphi_1))^2} \quad (1)$$

Next, various joint angles are derived. We utilized Equation 2 to determine the behavior of the PIP joint angle θ_3 with respect to the MCP joint angle θ_1 as it changes linearly over its flexion range. This is shown in **Figure 5A**.

$$\theta_3 = 2 \arctan \frac{-B \pm \sqrt{A^2 + B^2 - C^2}}{C - A} - \varphi_3 \quad (2)$$

Where

$$A = 2d_1 r_2 \cos\theta_1 - 2r_1 r_2 \cos\varphi_1$$
$$B = 2d_1 r_2 \sin\theta_1 - 2r_1 r_2 \sin\varphi_1$$
$$C = d_1^2 + r_1^2 + r_2^2 - l_1^2 - 2d_1 r_1(\cos\varphi_1 \cos\theta_1 + \sin\varphi_1 \sin\theta_1)$$

Finally, we use Equation 3 to find joint angle θ_2.

$$\theta_2 = 2 \arctan \frac{d_1 \sin(\theta_1) + r_2 \sin(\theta_3 + \varphi_3) - r_1 \sin(\varphi_1)}{d_1 \cos(\theta_1) + r_2 \cos(\theta_3 + \varphi_3) - r_1 \cos(\varphi_1)} \quad (3)$$

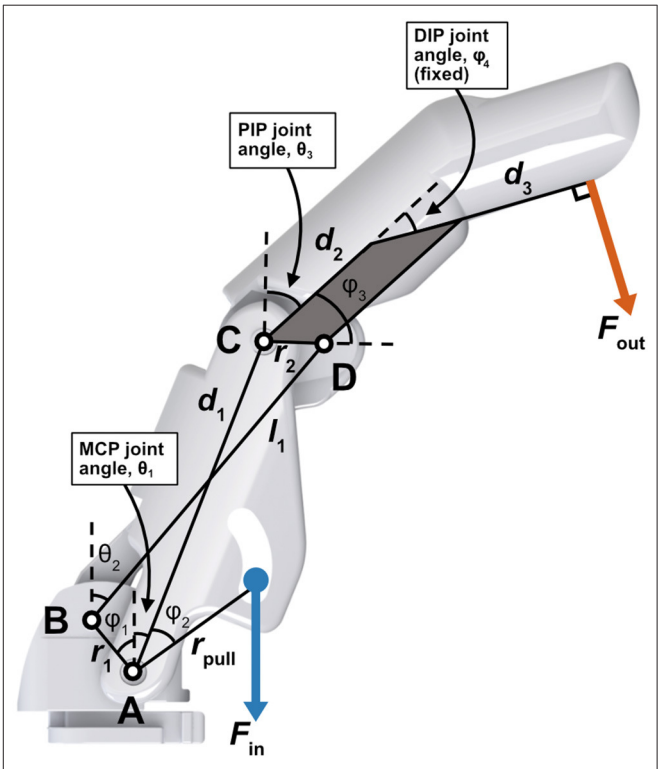


FIGURE 4 | Schematic describing the prosthetic finger (based off of commercially available prosthetic hand) represented by a crank-rocker four-bar linkage as used in the kinematic model (Murali et al., 2019). The proximal and distal phalanges are fused, causing the distal interphalangeal (DIP) joint to be fixed, while the metacarpal phalangeal (MCP) and proximal interphalangeal (PIP) joints are free to rotate. Actuators apply load F_{in} at r_{pull}. The pinch force is taken perpendicular to d_3 and is represented by F_{out}.

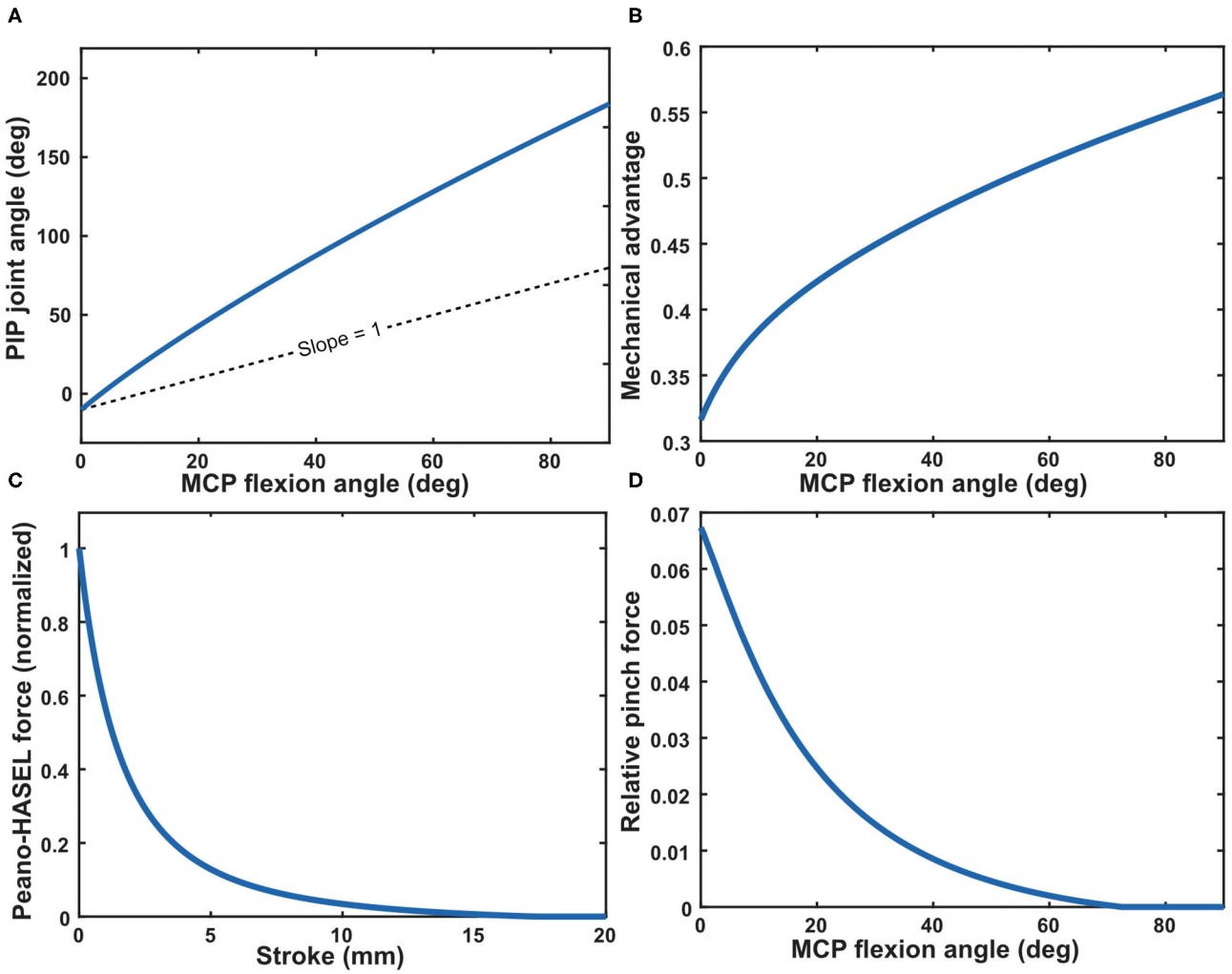


FIGURE 5 | (A) Model prediction of the PIP joint angle as a function of the MCP joint angle through its flexion range (0–90°). **(B)** Mechanical advantage over MCP joint flexion range. In this case, the mechanical advantage is determined by the inverse of the slope of the PIP joint angle with respect to the MCP joint angle as shown in **(A)**. The mechanical advantage is equivalent to the ratio of torque about the PIP joint to torque about the MCP joint (Norton, 2000). **(C)** Force output for Peano-HASEL actuators as predicted by the analytical model (Kellaris et al., 2019), normalized by the theoretical blocking force of the actuator. **(D)** The mechanical advantage **(B)** of the kinematic system of the finger translates the force of the Peano-HASEL actuator **(C)** to the fingertip, resulting in the predicted fingertip pinch force **(D)**. The pinch force is depicted relative to the blocking force of the actuator.

We can now describe the theoretical orientation of the finger at any instance throughout its flexion range and subsequently the mechanical advantage of the system. Mechanical advantage N is used to determine how force output will translate from the actuators to the fingertip (Norton, 2000). This value is found by comparing the angular velocities ω of two points of interest, which is inversely proportional to the ratio of torque T about those same points (Norton, 2000). The mechanical advantage of point C with respect to point A at each instance over its flexion range is described in Equation 4 (Norton, 2000; Murali et al., 2019) and is plotted in **Figure 5B**.

$$N = \frac{T_C}{T_A} = \frac{\omega_A}{\omega_C} = \frac{-r_2 \sin\left(\theta_2 - (\theta_3 + \varphi_3)\right)}{d_1 \sin\left(\theta_2 - \theta_1\right)} \tag{4}$$

Next, we integrated a previously developed analytical model for force output as a function of stroke of the Peano-HASEL actuators (Kellaris et al., 2019). We used this as the force input to the system, which allowed us to predict the theoretical force output at the fingertip when using Peano-HASELs as the actuation method, rather than the DC motor. Equation 5 describes the predicted force output of the Peano-HASEL actuators while Equation 6 predicts the corresponding stroke, represented by ϵ. Both are used to arrive at the force-stroke curve, presented with respect to MCP flexion angle, as seen in **Figure 5C**. The physical parameter represented by each variable can be seen in **Figure 6**.

$$F = \frac{w}{4t} \frac{\cos(\alpha)}{1 - \cos(\alpha)} \varepsilon_0 \varepsilon_r V^2 \tag{5}$$

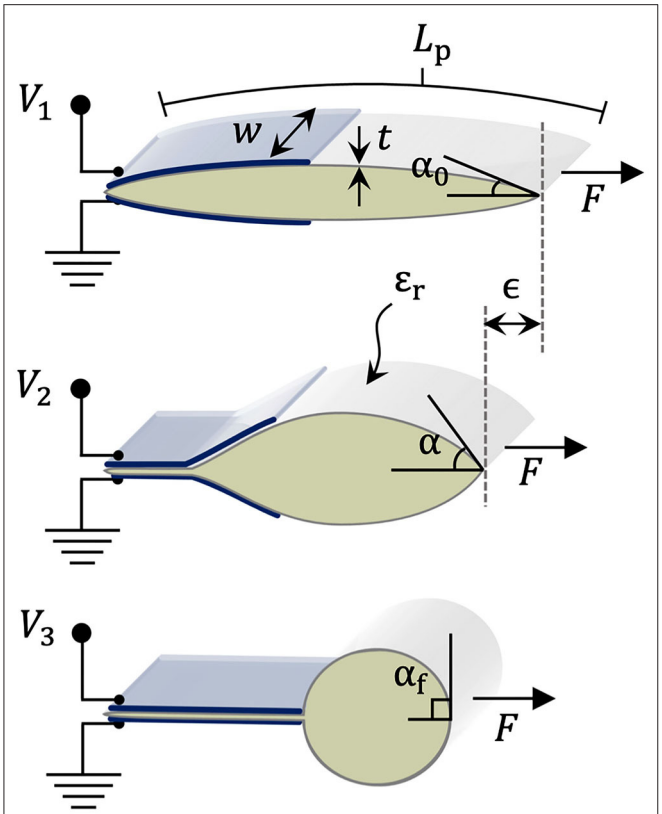


FIGURE 6 | Schematic describing geometric and material parameters for Peano-HASEL actuators at different stages of the actuation process, as voltage increases from V_1 to V_3 (Kellaris et al., 2019).

$$\epsilon = 1 - \frac{\alpha_0}{\sin(\alpha_0)}\left(1 + \frac{\sqrt{2A}}{L_p}\frac{\sin(\alpha) - \alpha}{\sqrt{\alpha - \sin(\alpha)\cos(\alpha)}}\right) \quad (6)$$

With the above, we can next obtain the force output of the fingertip over the MCP flexion range. We calculated the torque input to the system by taking the cross product of the actuator force vector and the moment arm about point A (r_{pull}). We multiplied this torque value with the mechanical advantage N to translate from torque about point A to torque about point C.

$$T_4 = N^* T_2$$

Finally, we geometrically translated the torque about point C to the normal (pinch) force out at the fingertip F_{out} using Equation 7. The predicted pinch force can be seen in **Figure 5D**.

$$F_{out} = \frac{T_4}{|\overrightarrow{T_{Tip}}| \cos\left(\frac{d_2 \sin(\theta_3+\varphi_3+\varphi_4)}{|\overrightarrow{T_{Tip}}|}\right)} \quad (7)$$

Where $\overrightarrow{T_{Tip}}$ is the position vector of the fingertip as measured from point C, as shown in Equation 8.

$$\overrightarrow{T_{Tip}} = \left(d_2\cos(\theta_3) + d_3\cos(\theta_3+\varphi_4)\right)\hat{i} + \left(d_2\sin(\theta_3) + d_3\sin(\theta_3+\varphi_4)\right)\hat{j} \quad (8)$$

Combining the full kinematic finger model with the Peano-HASEL model allows us to systematically alter finger parameters to change finger performance. This will be discussed further in the results section below.

Experimental Setup and Protocol

To compare the DC motor actuator and Peano-HASEL actuators, as well as to validate model predictions, testing was performed over a series of experimental protocols.

General Experimental Design

Form factor measurements were recorded using hand-held calipers and a bench-top scale (Optima Scales OPK-S500). The Peano-HASEL was powered by a voltage signal with reversing polarity (Kellaris et al., 2018). A custom Matlab script sent the signal to a NI DAQ (Model USB6212), which interfaced with a high voltage amplifier (TREK 50/12). The TREK was limited to 6 mA maximum current. An 8-kV signal was used for the BOPP actuators, while 6 kV was used for the Mylar actuators based upon previously determined safe operating voltages for each material. The DC motor actuator was powered by a benchtop DC power supply at 9 V, and the current limit was set at 5 A. A custom LabView interface controlled the motor through a custom motor controller board (Sigenics Inc, Chicago IL). The current and voltage input to the DC motor actuator was monitored from the LCD readout of the desktop power supply (Keysight U8002A) to measure the electrical power consumption during experimental trials.

For all experiments, the prosthetic digit was mounted on an acrylic plate supported by 4 threaded rods. Two holes were laser-cut through the plate, one allowing the DC motor or Peano-HASELs to pull down and another to prevent the acrylic plate from interfering with flexion of the fingertip. For the Peano-HASEL actuators, the base of the stand included a mounting point to interface with the actuators and can be seen in **Figure 1**.

Characterization of Peano-HASEL Force-Stroke Curve

The force-stroke curves of the Peano-HASEL actuators were collected using a custom Matlab program that interfaced with both the TREK and an artificial muscle tester (Aurora Muscle Tester 310C-LR). The length of the actuators was recorded under a pretension of 40 N from the artificial muscle tester. From that point, the muscle tester arm moved in the direction of contraction in set increments, and the Peano-HASELs were actuated with a ramped square wave voltage signal 4 times at each increment. The average force was then calculated for each displacement step.

Force Measurement

For force measurements, a variable-position load cell (Phidget CZL616C) was mounted to the plate allowing for collection of fingertip force over a wide angular range. The load cell was calibrated using a set of calibration weights before the experimental protocol was completed. Using a SolidWorks model of the prosthetic finger, calibration blocks were 3D printed to ensure proper position of the load cell for each MCP angle of

interest. The Peano-HASEL actuators were affixed in the test fixture as shown in **Figure 1** and pre-tensioned by raising the height of the stand until the MCP angle reached 0°.

Force values were collected over two experimental protocols by varying the voltage signal sent to the TREK. To measure pinch force, the average pinch force value was collected over a 1 s grasp and repeated 5 times at each angle of interest. To measure continuous grasp, the force was collected over a 10 s grasp and repeated 4 times. Both tests included a 0.25 s ramp up and down during voltage transitions to minimize kinematic effects, though any spikes in force attributed to kinematics were ignored.

Power Consumption

A custom Matlab program acquired voltage data from the internal monitors of the Trek and current data from a current sensor (μCurrent GOLD) simultaneously with force testing to determine the energy consumption and power drawn during pinch and continuous grasp.

Dynamic Behavior

To measure free stroke, the load cell was removed and the finger was allowed to move freely through its maximum flexion range. The angle of the MCP joint was tracked optically (Canon EOS 6D DSLR) and measured using an open-source software program (Tracker version 5.1.3). The voltage signal was ramped over 0.5 s to minimize kinematic effects.

The dynamic specifications of the actuators were also tested. A high-speed camera (Model Phantom v710) was used to measure the temporal properties of the Peano-HASEL and a Canon EOS 6D DSLR was used with the DC motor actuator. The impulse speed was quantified using a square wave input signal at the appropriate voltage or pulse-width (100%) for the Peano-HASELs and DC motor, respectively. The bandwidth was studied by applying various frequencies between 0.5 and 50 Hz, using a reversing polarity sine wave at the appropriate voltage for the Peano-HASELs and a square-wave with 100% pulse-width for the DC motor. The angle of the MCP joint was tracked optically and processed, also using the software program Tracker. A resonant frequency in the Peano-HASEL actuator system was observed between 10 and 25 Hz. For those trials, small acrylic side constraints were added to minimize large motion perpendicular to the direction of actuation. For the Mylar stack of actuators, a 5.5 kV voltage signal was used (as opposed to the 6 kV amplitude for force testing), to reduce risk of electrical failure during the high cycling frequency required for dynamic testing.

During dynamic testing, no external loads were applied to the prosthetic finger system. However, the spring element (torsional spring for the original finger and elastic band for the custom finger) was present in the prosthetic finger system. This element provided a variable load proportional to the flexion magnitude which acted against the actuators, therefore all force and dynamic data was collected under a variable load condition.

EXPERIMENTAL RESULTS

Validation of Kinematic Model

The original design, based on the commercially available Bebionic finger, was analyzed with the intention of validating the kinematic model. The torsional spring was removed in order to isolate the kinematic behavior of the finger. Without the torsional spring, the finger rested on the load cell at various angles. Force data was collected based off the described protocol at a range of MCP flexion angles.

The finger and test stand demonstrated elastic behavior during testing. In the finger, imperfect rotational joints contributed to unwanted deflection, and the thermoplastic used to 3D print the finger bent under load. The acrylic stand and load cell mount also deflected under load. This unwanted elastic behavior was modeled as a linear spring between the actuators and the prosthetic digit. The software program Tracker was used to measure the elasticity of the system- at a known force, the deflection of the fingertip was measured, allowing an estimate for the spring constant of the system to be obtained. As shown in **Figure 7**, when we incorporated this experimentally-measured spring constant in the system, we found that the model matched the experimental data very well.

While configuring the finger without the torsional spring is useful for validating the kinematic model, removing the torsional spring results in a prosthetic finger with no restoring force. In the original finger design that includes the torsional spring, the Peano-HASEL actuators did not provide any grip force past an MCP angle of 30° which is insufficient for gripping objects.

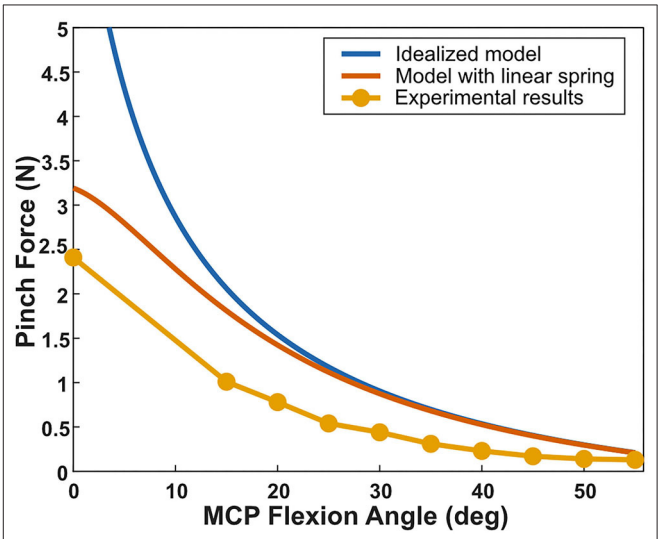


FIGURE 7 | Comparing the experimental results for the fingertip pinch force as a function of MCP angle with the idealized model and the model that includes a linear spring. The linear spring constant was measured experimentally and accounts for unwanted deformation in both the prosthetic digit and the test setup as well as the elasticity of the Peano-HASEL actuators. The experimental results for the pinch force match well with the model that includes the linear spring.

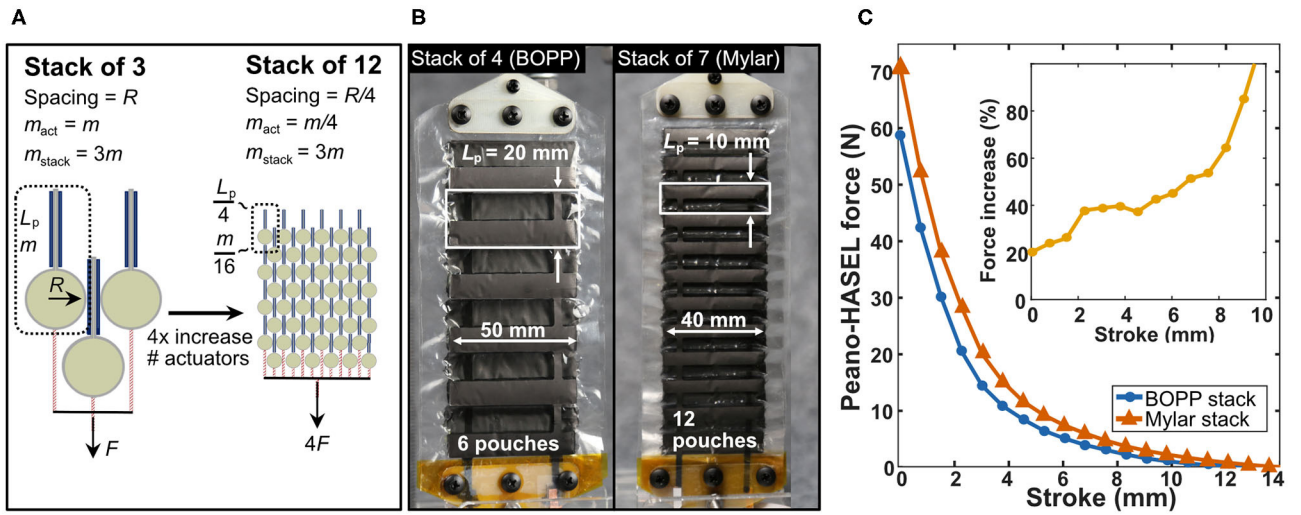


FIGURE 8 | Designing weight-matched stacks of Peano-HASEL actuators with higher force output. **(A)** By fabricating actuators with decreased pouch length L_p and using a series of smaller pouches in place of a single larger pouch, the mass of the actuator is reduced without affecting its force-strain performance (Kellaris et al., 2019). In this way, more actuators can be stacked in parallel to produce a weight-matched stack with increased force output. **(B)** Photos of the actuators used in this work. The first stack of actuators tested used 4 actuators made from BOPP with 6 pouches each (43.6 g total). A second stack was produced with decreased pouch length using 7 actuators made from Mylar with 12 pouches each (38.8 g total). **(C)** Force-stroke curves for both stacks of Peano-HASEL actuators. The stack of Mylar Peano-HASEL actuators with shorter pouch length produced higher force than the original design while maintaining similar weight. The inset shows the relative force increase of the Mylar Peano-HASELs with respect to the BOPP Peano-HASELs.

Improving Actuator Force Output

First, we attempted to increase the force output of the finger by improving the force output from the Peano-HASEL actuator stack. In Equation 5, we observe that the force output F of a Peano-HASEL is independent of its pouch length L_p. As a result, Kellaris et al. showed that actuators comprised of a series of shorter pouches have a smaller overall mass m_{act} than actuators comprised of fewer but longer pouches, while maintaining the same force-stroke characteristics (**Figure 8A**; Kellaris et al., 2019); thus, fabricating actuators with shorter pouches allows us to stack more actuators in parallel, while maintaining the same weight for the stack overall.

Following this strategy, actuators with decreased pouch length were constructed from Mylar film, with pouch geometries shown in **Figure 8B**. The reduced weight of these actuators allowed seven Mylar actuators to be stacked in parallel for improved force output while maintaining the same overall weight as the stack of four BOPP actuators. This film also has a higher permittivity, which acts to increase force output, according to Equation 5. Mylar film with a thickness of 12-μm was used, rather than the 18-μm BOPP, to help minimize losses due to bending stiffness of the film (Kellaris et al., 2019) and to reduce necessary operating voltages. The width of the pouches was decreased from 5 to 4 cm to reduce instabilities in actuation, as reported by Rothemund et al. (2019). While decreasing film thickness and pouch width act to lower the force output according to Equation 5, the larger number of actuators and the higher permittivity film offset these effects and increased the overall force output of the stack. **Figure 8C** shows the experimental force-stroke curves for the BOPP stack of four and the Mylar stack of seven, demonstrating the improved force output for the Mylar actuator stack.

Improvement of Finger Design

A second way to improve the force output of the prosthetic finger is to modify the finger kinematic design for actuation with Peano-HASELs rather than a DC motor. The previously discussed kinematic model was used to evaluate how modifications to the four-bar linkage finger system would affect the fingertip force output at various grip angles when actuated with Peano-HASELs.

The Peano-HASELs provided forces up to 2.57 N at very low flexion angles (0–2°), but many gripping tasks fall within a range of 24–55° (Lee and Jung, 2016). Parameters were systematically changed with the goal of increasing force output within this common grip range. Without additional constraints many finger designs resulted in non-lifelike actuation or behavior very different from the original design, based on the Bebionic finger (e.g., no flexion in the PIP joint across the entire range of flexion in the MCP joint). Although these designs predicted higher force output in the desired range, they were deemed unacceptable as the prosthetic device must mimic the intact biological system it resembles.

Parameters were constrained such that (1) the relaxed position of the finger would not change, (2) the fingertip itself would be at a similar location in space as compared to the original design, when the MCP angle was flexed to 35° (ensuring appropriate flexion of the PIP joint), and (3) the proximal and distal phalanges' size and shape remained the same as the original design, maintaining the

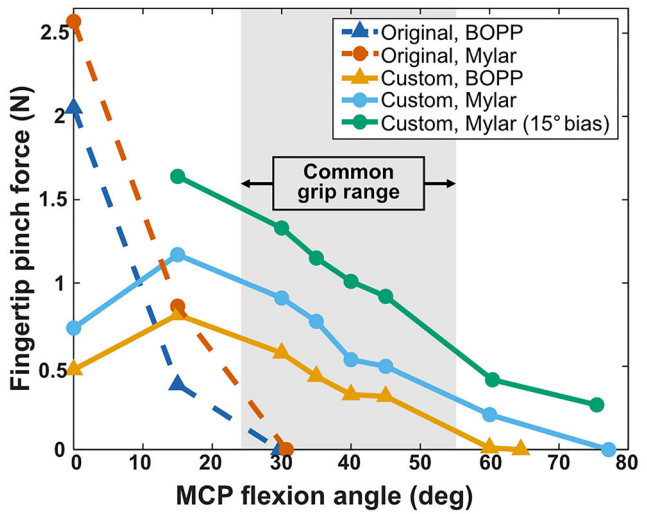


FIGURE 9 | Fingertip pinch force across original and custom prosthetic finger designs and BOPP and Mylar stacks of Peano-HASEL actuators. The original finger (dashed lines) provides a high pinch force that rapidly decreases, providing no pinch force over 30°. The custom prosthetic finger sacrifices higher blocked force for increased pinch force within the common grip range (24–55°) (Lee and Jung, 2016) and provides over twice the flexion range as the original design. The resting MCP angle of a human finger can be over 30°, so an initial bias of 15° was applied to further increase the fingertip force within the common grip range (Lee et al., 2008; Lee and Jung, 2016).

lifelike appearance of the finger. Under these conditions, new finger parameters were selected. The custom finger components were printed with a 3D printer (Formlabs Form 2). At each joint two members were connected with a metal pin, with one member of each joint designed for a press-fit while the other member was designed for a freely rotating fit.

We also removed the torsional spring in the custom finger design and replaced it with an elastic band (1/4″ diameter 2.5 oz force, Prairie Horse Supply) spanning from the knuckle joint to the proximal phalanx on the dorsal side of the finger. The elastic band was added to serve as a restoring force which pulled the finger back to its resting position.

The results of force vs. MCP angle are presented for the original and custom finger designs in **Figure 9**. For each design, force data was collected using both the BOPP and Mylar Peano-HASELs. In all cases, the Mylar Peano-HASELs increased the fingertip force output as compared to the BOPP Peano-HASELs, with an average increase of 62%. At a blocked finger position (0°) the original finger design provides the highest force, but decreases rapidly to its limit at ∼30°. The custom finger design sacrifices pinch force at the lowest angles for drastically increased fingertip force within a common grip range (24–55°) (Lee and Jung, 2016). We further increased the pinch force by an average 68% within this range by adding an initial bias of 15°. This accounted for the fact that the MCP angle of a relaxed human finger can be over 30° (Lee et al., 2008; Lee and Jung, 2014).

The custom finger also greatly increased the range of motion of the prosthetic finger. Various experiments have indicated the natural maximum angle of MCP flexion is between 60 and 90° (Lee and Jung, 2014). While the original design provided no pinch force past 30°, the custom finger (without initial bias) reached 77° of flexion.

Further Kinematic Characterization

Energy Consumption in Pinch

Driving a prosthetic finger with Peano-HASELs provides many additional benefits that are relevant to the user. We characterized energy consumption behavior based on the previously described protocol for grip and 10 s hold. The applied voltage and corresponding force output across four testing cycles can be seen in **Figure 10A**. As seen in **Figure 10B**, the BOPP consumes 66.4 mJ over the 10 s hold, while Mylar consumes 14.3 mJ over the same hold. The DC motor actuation system does not require additional energy to hold its position, but only through the use of additional transmission components. Without those components, the DC motor requires 2,100 mJ to hold its position. The HASEL actuators do not require any additional components to display this behavior. This property—known as having a catch state—is typical for electrostatic actuator systems.

The power draw for both types of actuators was determined from the collected data and is presented in **Figure 10C**. The Mylar stack consisted of 7 actuators, compared to the 4 BOPP actuators, resulting in a higher power draw during flexion (**Figure 10C**, inset left). However, during continuous grip, the BOPP consumed an average of 5.6 mW while the Mylar consumes an average of just 0.94 mW. This is due to the lower leakage current the Mylar film displays. Overall, Peano-HASEL actuators consume power primarily during transitions between actuation states, with very little power consumption during the hold state.

Dynamic Characterization of Actuators

The kinematic properties of the actuators include the range of motion of the prosthetic finger (in degrees), the maximum angular speed of the prosthetic finger (in degrees/second), and the bandwidth of the prosthetic finger system (in Hertz).

Stroke

The range of motion of the original prosthetic finger when actuated with the BOPP Peano-HASELs (29.7°) corresponded to 35% of the maximum range of motion when actuated with the DC motor (85°). With the custom finger design and Mylar Peano-HASELs, the range of motion increased to 77.17°, corresponding to 91% of the range of motion of the original design (DC motor system).

Step-Voltage Response

A major advantage of Peano-HASEL actuators is their use of a fast electrohydraulic mechanism (Kellaris et al., 2018). The dynamics of Peano-HASEL actuators under different loads and operating conditions were studied in detail by Rothemund et al. (2020). To highlight the fundamental benefits and drawbacks

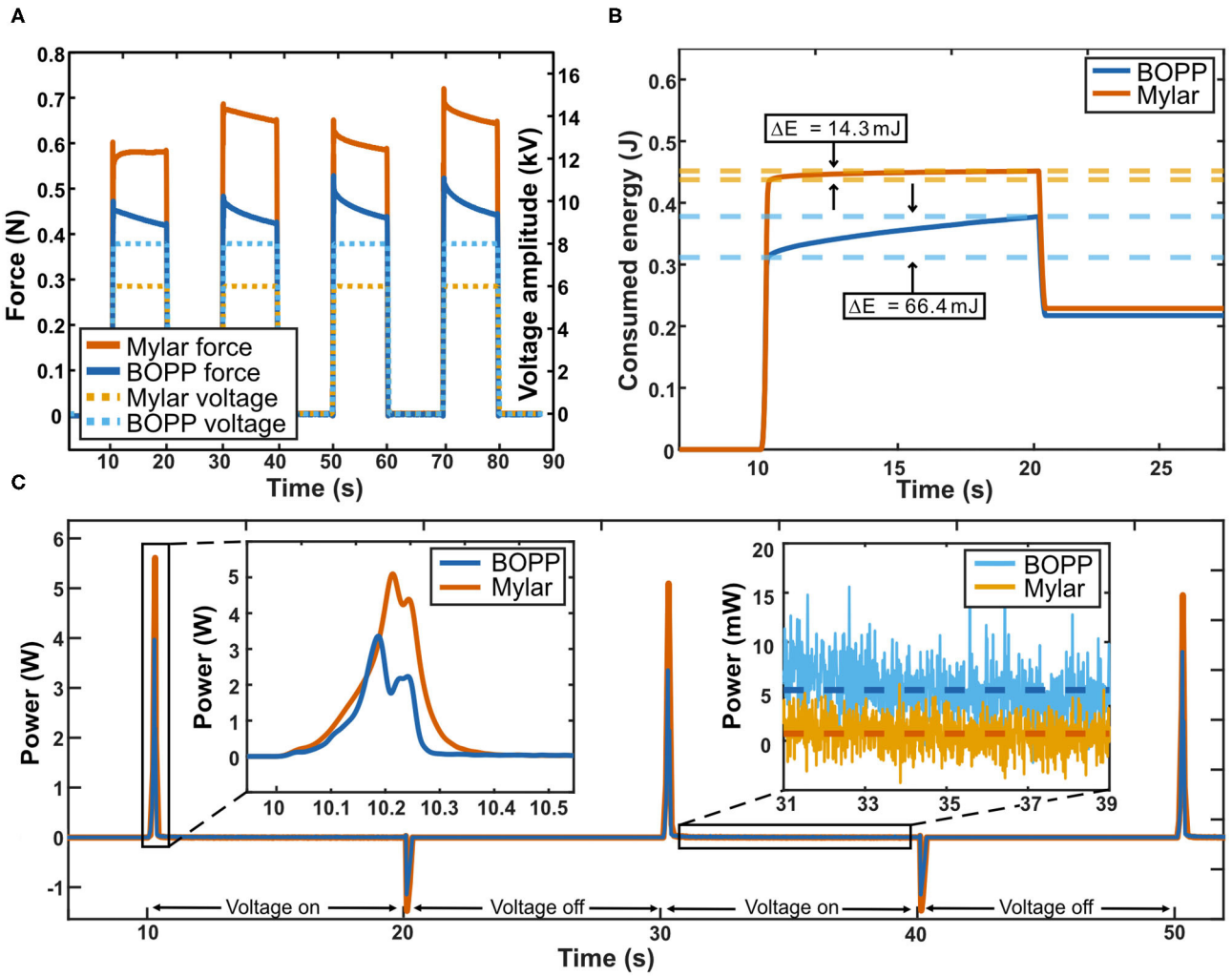


FIGURE 10 | Characterizing energy consumption of Peano-HASEL actuators during grasping. **(A)** Voltage signals (dashed line) applied for 4 repeated 10 s grasps. The corresponding pinch force is also shown (solid line). **(B)** Energy consumed during 10 s continuous grasp. Both BOPP and Mylar actuators used very little energy to maintain their state during continuous grasp due to their "catch" state. Mylar demonstrated lower energy consumption (14.3 mJ) during a continuous grasp than BOPP (66.4 mJ). During actuator relaxation, energy can be recovered via additional circuitry that harnesses the discharge current. **(C)** Power draw over multiple 10 s continuous grasp cycles. Power drawn by the actuators over the 10 s continuous grasp was largely contained to the 0.25 s voltage ramp causing initial flexion of the finger (inset, left). During continuous grasp the BOPP Peano-HASELs drew an average of 5.6 mW while the Mylar Peano-HASELs drew an average of just 0.94 mW (inset, right).

of the Peano-HASEL and DC motor actuators, the dynamics of the prosthetic finger system in this work were studied under no load.

Therefore, in our experiments the maximum flexion speed was measured about the axis of rotation at the MCP joint and is plotted in **Figure 11A**. The original finger actuated by BOPP Peano-HASELs demonstrated an average angular speed of flexion (738 deg/s) that was 4.9 times greater than the original finger design actuated by the DC motor actuator (150 deg/s). With the custom finger and Mylar Peano-HASELs, the average angular speed of flexion was 10.6 times greater (1,587 deg/s) than the original finger design actuated by the DC motor (**Supplementary Video 1**). The slope of the line in **Figure 11A** indicates the instantaneous angular velocity of the finger.

Frequency Response

Peano-HASEL actuators display a very high bandwidth (Kellaris et al., 2018). The frequency response of the system was characterized, with the cutoff frequency considered when the displacement is −3 dB of the maximum angular displacement. The bandwidth of the original finger when actuated by the BOPP Peano-HASELs is 21 times greater (15.9 Hz) than the original finger actuated by the DC motor (0.75 Hz), and the bandwidth for the custom finger actuated by Mylar Peano-HASELs was 11.1 times greater (8.3 Hz) than the original finger actuated by the

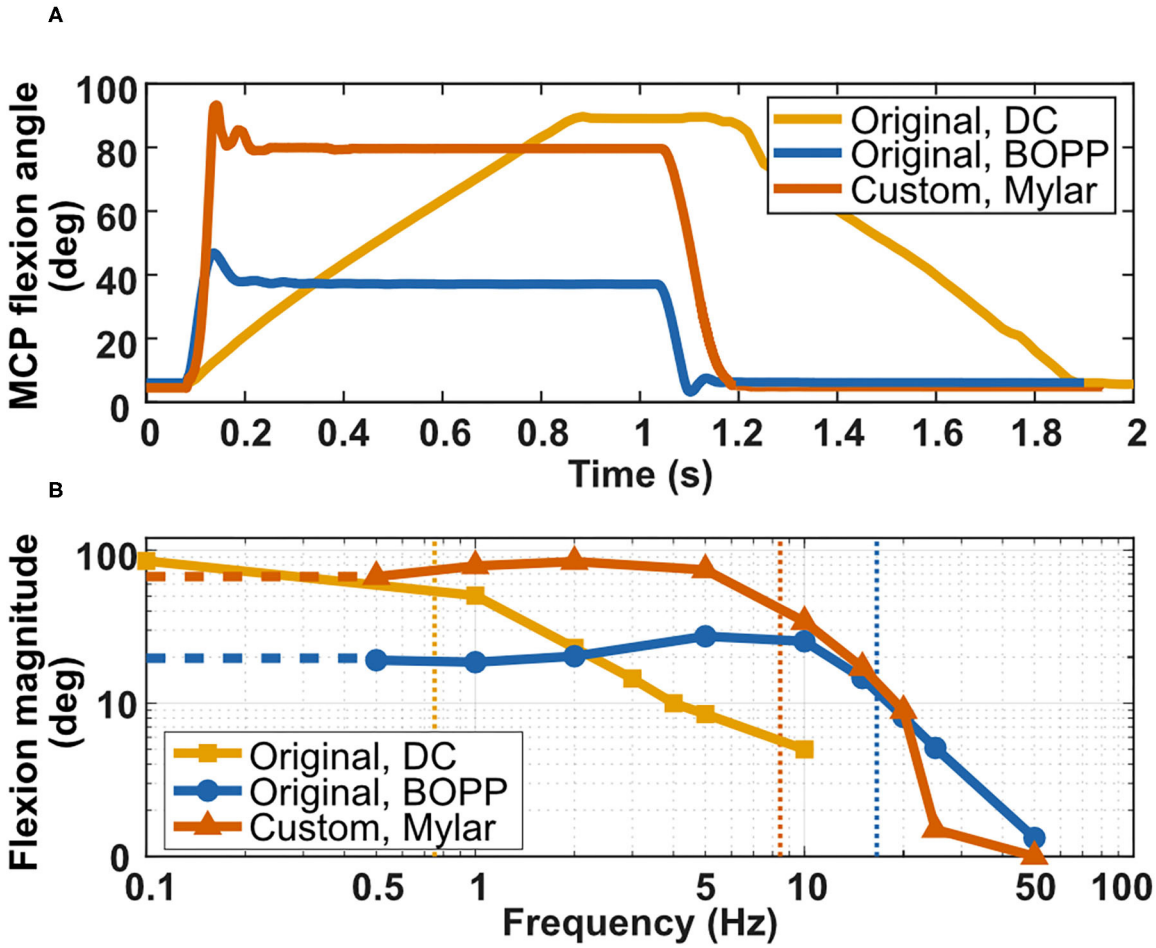


FIGURE 11 | Characterizing the dynamic performance of the prosthetic fingers. **(A)** Comparing impulse response for (1) the original finger actuated by DC motor, (2) the original finger actuated by BOPP Peano-HASELs, and (3) the custom finger actuated by Mylar Peano-HASELs. **(B)** Bode plot comparing roll-off frequencies. The roll-off frequency was considered at −3 dB of the low-frequency amplitude.

DC motor. **Supplementary Video 2** demonstrates actuation of this system at 1 and 5 Hz for a sinusoidal input voltage signal at 5.5 kV.

When the custom finger was actuated with the Mylar Peano-HASELs, there is a rapid decrease in amplitude that can be attributed to significant resonance in the actuator system from 10–25 Hz. Rather than actuating the finger, the Peano-HASELs moved side-to-side creating large oscillations perpendicular to the desired direction of actuation of the Peano-HASELs (**Supplementary Video 3**). Real-world applications of this technology would include mechanical constraints to prevent this out-of-plane behavior. The Bode plot for both systems can be seen in **Figure 11B**.

DISCUSSION AND OUTLOOK

This work is a first attempt to drive a prosthetic finger with Peano-HASEL actuators. Performance is compared to existing DC motor actuators used today. The benefits and drawbacks to this new actuator technology are elucidated by the results summarized in **Table 1**.

The custom finger actuated by Peano-HASEL actuators provided many benefits as compared to the original finger design powered by the DC motor. The custom finger system is 10.6 times faster, has 11.1 times higher bandwidth, and consumes 8.7 times less electrical energy to grasp. It reaches 91% of the maximum range of motion of the original finger.

However, the force production of these Peano-HASELs is substantially less than the DC motor actuator (∼10× decrease). Smaby et al. identified average pinch force values for every-day tasks which range from 1.4 N (push remote button) to 31.4 N (insert slippery plug into wall outlet) (Smaby et al., 2004). The current fingertip pinch force from Peano-HASEL actuators is still insufficient for many of these tasks. Two properties of the custom prosthetic finger system primarily affect this result: (1) specific energy of the Peano-HASEL actuators and (2) design of the prosthetic finger.

TABLE 1 | Summary of collected data comparing the original finger design actuated by both the DC motor and BOPP Peano-HASELs with the custom finger actuated by Mylar Peano-HASELs.

	Original finger design, DC motor	Original finger design, BOPP Peano-HASELs	Custom finger design, Mylar Peano-HASELs
Actuator weight (g)	37.0	43.6	38.8
Length × width × depth (cm)	5 × 2 × 1.5	15.5 × 5.4 × 0.65	16 × 7 × 0.64
Volume (cm^3)	15	54.4	71.7
Range of motion (deg)	85	29.7	77.17
Force at 30° (N)	13	0	0.91 (1.33)
Energy consumed to grasp (mJ)	3,800	311.4	437.5
Average power consumption during grasp (mW)	9,200*	5.6	0.94
Flexion speed (deg/s)	150	738	1,587
Bandwidth (Hz)	0.75	15.9	8.34
Flexion magnitude at 10 Hz (deg)	5.0	25.4	34.1

*Values in parentheses are those measured using a 15° offset in initial MCP angle. *Assumes continuous current draw to maintain force during grasp when using DC motor actuator.*

Specific Energy of Peano-HASEL Actuators

The analytical model presented by Kellaris et al. (2019) details how we can produce a Peano-HASEL stack with higher specific energy that consequently results in higher force output while keeping the total weight constant. Production of these actuators is currently limited by available plastic films and manufacturing techniques. As the pouch size becomes smaller, the bending stiffness of the plastic plays a more prominent role in limiting actuation (Kellaris et al., 2019). In the future, Peano-HASEL stacks with a substantial increase in specific energy may be possible using thinner plastic films and borrowing multi-layer soft lithography or precision laser micromachining processes that have been shown to produce fluidic actuators on the 100-μm scale (Ho and Jow, 2009; Moretti et al., 2018). Pouches on these length scales would allow for actuator stacks with drastically improved force output. In addition to scaling down pouch lengths, driving voltage can be increased to further increase the specific energy of the Peano-HASEL actuators. The force production of the Peano-HASEL is proportional to the square of the excitation voltage, so finding a reliable way to increase the excitation voltage without damaging the actuators (e.g., different materials, better manufacturing techniques) would cause a quadratic increase in force production (Suo, 2010; Acome et al., 2018; Kellaris et al., 2018).

Design of the Prosthetic Finger System

Altering the kinematics of the prosthetic finger increased the fingertip force at common grip angles. However, there are still opportunities for further improvements to the kinematic design. The finger was changed under many constraints; modifying these constraints could lead to more design freedom allowing for higher force production at various angles. As manufactured, the custom prosthetic finger demonstrated elastic behavior. A desktop 3D printer was used with resolution set to 100 μm, and the metal pin joints were modified and assembled by hand. Better manufacturing techniques with stiffer materials would lead to less elastic behavior and less frictional losses in the system, resulting in higher fingertip force output (Ngo et al., 2018). In this work a four-bar linkage design was used, but other designs currently in use could be adapted and optimized for actuation with Peano-HASELs, including other four-bar linkage conformations (Vincent by Vincent Systems), tendon-roller systems (iLimb by Touch Bionics) and intrinsic finger actuation designs like those presented by Murali et al. (2019). Overall, the results presented here show the first steps toward improving upon the standard design of prosthetic devices and indicates that further research in inventing prosthetic mechanisms specifically designed for the Peano-HASEL actuators is warranted.

The dynamic performance of the Peano-HASELs exceeds the abilities of the DC motor actuator. With the custom finger, maximum speed of flexion was 10.6 times greater and the bandwidth was 11.1 times greater than the DC motor actuator. Reduced pouch dimensions and lower viscosity dielectric fluids could lead to even faster actuation (Rothemund et al., 2020). This dynamic performance will become even more important as improved myoelectric control algorithms become available for prosthetic hands. Furthermore, the lifetime of currently available Peano-HASEL actuators is less than that of well-established DC motors. Peano-HASELs produced over 20,000 cycles when operated at their upper voltage limit (Kellaris et al., 2018). Kellaris et al. modeled Peano-HASEL actuators that could maintain their force output while operating at lower voltages, thereby resulting in longer lifetimes (Kellaris et al., 2019).

There are several additional aspects of the Peano-HASEL actuators that make them attractive for further development toward practical implementation in upper-limb prosthetic design, including: (1) bio-mimetic form factor and weight distribution, (2) the series-elastic nature of the actuators, (3) their self-sensing capability, (4) their compatibility with novel myoelectric control systems, (5) the availability of miniaturized high-voltage power electronics.

Bio-Mimetic Form Factor and Weight Distribution

The tradeoff between force and weight has plagued prosthetic device design, since prosthetic users consider weight to be one

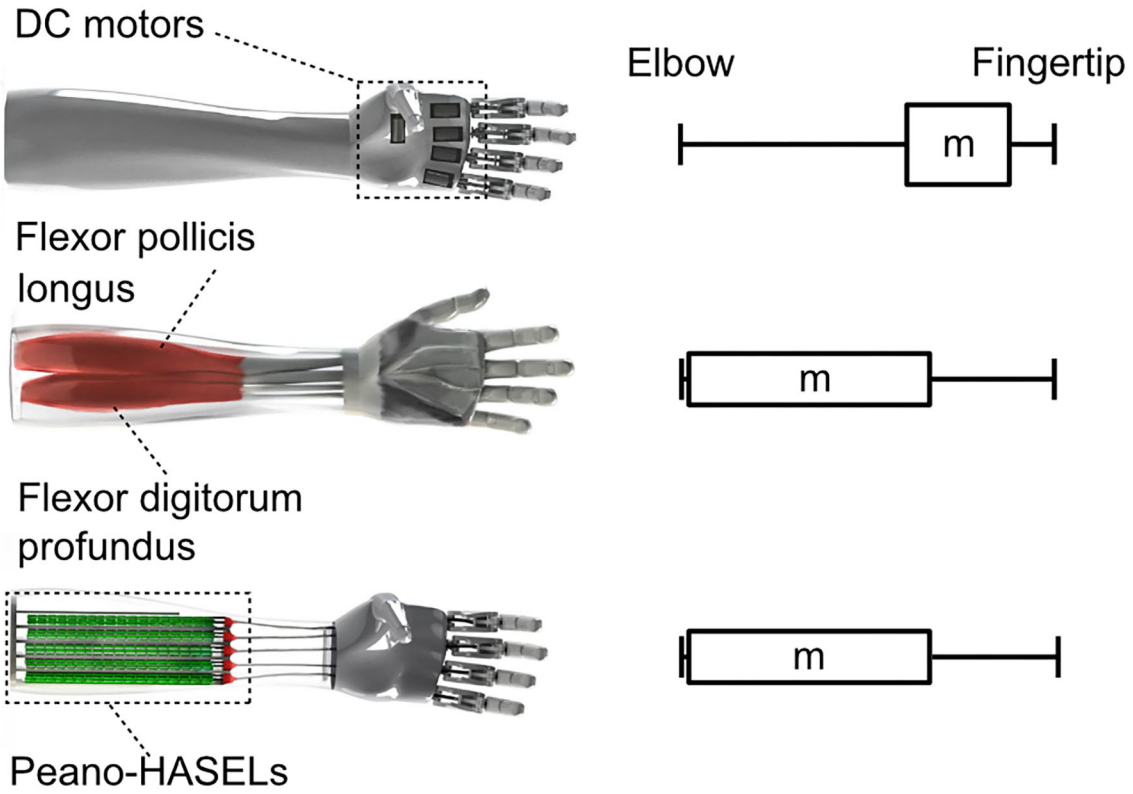


FIGURE 12 | Peano-HASEL actuators simplify the design of prosthetic limbs that have a more proximal weight distribution, which decreases the load experienced by the user. The overall design of a prosthetic limb based on Peano-HASEL actuators can better mimic the physiology of the human arm.

of the major deficits of upper-limb prosthetic devices today (Biddiss et al., 2007; Cordella et al., 2016). The distribution of weight in a prosthetic socket can have an outsized effect due to the moment arm between the residual limb and prosthetic hand. In effect, additional weight on the distal end is amplified by these large moment arms and promotes actuator design which can be distributed away from the distal end of the prosthesis. The flat form factor of Peano-HASEL actuators, coupled with their inherent linear contraction on activation—without the need for bulky gears and transmission systems—means they can more easily apply tension to a tendon/cable from afar. The actuators could then be located more proximally in the prosthetic socket which would more closely resemble the distribution of weight in an intact human limb (**Figure 12**). This design change would reduce the load experienced by the user and could lead to additional benefits like compliant digits/palm and robustness to environmental hazards like water, dirt, etc.

Series-Elastic Nature of the Actuators

The series-elastic nature of Peano-HASEL actuators is based on compliance of the electrohydraulic structure while exposed to varying external loads. The benefit of a series elastic actuator in a prosthetic device includes improved robustness to perturbation and safety when interacting with people and/or delicate objects (Sensinger and Weir, 2008). Peano-HASELs would enable these features without any additional componentry (like springs, sensors, and controllers) typically required to create a series elastic actuator when using a DC motor.

Self-Sensing Capability

The self-sensing nature of deformable capacitors stems from changes in capacitance during actuation (Acome et al., 2018), which can be mapped to the state of the actuator and used in a closed-loop position control system (Schunk et al., 2018; Ly et al., 2020). The ability of Peano-HASELs to inherently self-sense position could result in space/time/cost savings compared to DC motors, which require additional ancillary components in order to determine their rotational position in space (typically a motor encoder).

Compatibility With Novel Myoelectric Control Systems

Myoelectric control systems are the most commonly used methodology to interface the amputee to the prosthetic limb (Childress and Weir, 2004). The measurement of muscle activity in the residual limb is used as a control signal to determine the position/speed of an actuator in the device. Significant research has focused on the best algorithms for this mapping

(Scheme and Englehart, 2011). In all cases, the use of a Peano-HASEL can be incorporated into a myoelectric control system. As shown in this work, control signals can be sent from a controller to high-voltage power electronics to actuate the Peano-HASELs. Therefore, new myoelectric control algorithms could be developed that take advantage of the stacked design of Peano-HASELs, which mimic the structure of our intact muscular anatomy based on hierarchical muscle anatomy. The force that a muscle creates is dependent on the firing rate of the action potentials (similar to the excitation voltage of the Peano-HASEL) and the number of motor units that are recruited (similar to the number of pouches/actuators in the Peano-HASEL stack that are activated) (Enoka, 1995). The modulation of the number of Peano-HASEL pouches/actuators that are active at any given time permits soft, delicate motion as well as powerful grasps. A similar control methodology is not possible using established DC motor technology.

Availability of Miniaturized High-Voltage Power Electronics

Finally, commercially available high-voltage amplifiers and switches are already available in small form factors which could be fitted inside typical prosthetic sockets (Schlatter et al., 2018). **Supplementary Table 1** details such miniature electronics that could be used, with size, weight and voltage operating limits. Existing lithium ion batteries (FlexCell, Infinite Biomedical Technologies) could be used to power these circuits and actuate Peano-HASELs. The safety of the amputee is of concern when high-voltage components are used in prosthetic systems. Insulating materials such as rubber can be used to shield the user (Pourazadi et al., 2017), and actuator stacks can be designed such that the outermost electrodes in the stack are grounded, further shielding the user from high voltage to allow safe incorporation into a prosthetic limb.

This paper presents the remaining challenges and highlights the strong potential of Peano-HASEL actuators to realize the next generation of multi-functional and lifelike prosthetic devices.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Materials**, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

CK and JS conceived and supervised the research. ZY and NK conceived the design of experiments and fabricated actuators. ZY collected and analyzed data with help from CC-M. CC-M and ZY developed the kinematic model of the finger and actuator system with guidance from JS and NK. ZY, CC-M, and JS designed the prosthetic fingers used in this work. DR and CC-M designed the testing fixture for force output tests. MBE helped design early prototypes of the experimental setup. SKM and JS collected and analyzed DC motor data. ZY, NK, CC-M, and JS drafted and revised the manuscript and figures, with help from DR and guidance from CK. All authors contributed to the article and approved the submitted version.

FUNDING

This work was sponsored by a Packard Fellowship from The David and Lucile Packard Foundation, startup funds from the University of Colorado Boulder, a GAANN (Graduate Assistantships in Areas of National Need) Fellowship in Soft Materials, an NSF grant for Cyber-Physical Systems (Grant No. 1739452), and an NSF EFRI Research Projects grant (Grant No. 1830924). The authors also acknowledge funding from a seed grant at the University of Colorado Boulder in Interdisciplinary Research Themes (IRT) for Multi-functional Materials as well as an Army Research Office DURIP grant (Grant No. W911NF-18-1-0203). JS acknowledges financial support from a US Department of Veterans Affairs Rehabilitation and Development Service Program Career Development Award 1 (CDA-1 award number IK1 RX002011-01A2). CC-M acknowledges financial support from the Undergraduate Research Opportunity Program (UROP) at the University of Colorado Boulder.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Sean Humbert for the use of high-speed camera equipment used in the characterization of Peano-HASEL actuators (Phantom v710 and high-power tungsten flood lights).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2020.586216/full#supplementary-material

REFERENCES

Acome, E., Mitchell, S. K., Morrissey, T. G., Emmett, M. B., Benjamin, C., King, M., et al. (2018). Hydraulically amplified self-healing electrostatic actuators with muscle-like performance. *Science* 359, 61–65. doi: 10.1126/science.aao6139

Anderson, I. A., Gisby, T. A., McKay, T. G., O'Brien, B. M., and Calius, E. P. (2012). Multi-functional dielectric elastomer artificial muscles for soft and smart machines. *J. Appl. Phys.* 112, 41101. doi: 10.1063/1.4740023

Belter, J. T., Segil, J. L., Dollar, A. M., and Weir, R. F. (2013). Mechanical design and performance specifications of anthropomorphic prosthetic hands: A review. *J. Rehabil. Res. Dev. Wash.* 50, 599–618. doi: 10.1682/JRRD.2011.10.0188

Biddiss, E., Beaton, D., and Chau, T. (2007). Consumer design priorities for upper limb prosthetics. *Disabil. Rehabil. Assist. Technol.* 2, 346–357. doi: 10.1080/17483100701714733

Biddiss, E., and Chau, T. (2008). Dielectric elastomers as actuators for upper limb prosthetics: challenges and opportunities. *Med. Eng. Phys.* 30, 403–418. doi: 10.1016/j.medengphy.2007.05.011

Childress, D., and Weir, RF ff. (2004). "Control of Limb Prostheses," in *Atlas of Amputations and Limb Deficiencies*, 3rd ed., eds D. G. Smith, J. W. Michael, and J. H. Bowker (Rosemont, IL: American Academy of Orthopedic Surgeons).

Childress, D. S. (1985). Historical aspects of powered limb prosthetics. *Clin. Prosthet. Orthot.* 9, 2–13.

Controzzi, M., Cipriani, C., and Carrozza, M. C. (2014). "Design of artificial hands: a review," in *The Human Hand as an Inspiration for Robot Hand Development*, eds R. Balasubramanian, and V. J. Santos (Cham: Springer International Publishing), 219–246. doi: 10.1007/978-3-319-03017-3_11

Cordella, F., Ciancio, A. L., Sacchetti, R., Davalli, A., Cutti, A. G., Guglielmelli, E., et al. (2016). Literature review on needs of upper limb prosthesis users. *Front. Neurosci.* 10:209. doi: 10.3389/fnins.2016.00209

Cura, V. O. D., Cunha, F. L., Aguiar, M. L., and Cliquet, A. Jr. (2003). Study of the different types of actuators and mechanisms for upper limb prostheses. *Artif. Organs*. 27, 507–516. doi: 10.1046/j.1525-1594.2003.07000.x

Duduta, M., Hajiesmaili, E., Zhao, H., Wood, R. J., and Clarke, D. R. (2019). Realizing the potential of dielectric elastomer artificial muscles. *Proc. Natl. Acad. Sci. U.S.A.* 116, 2476–2481. doi: 10.1073/pnas.1815053116

Enoka, R. M. (1995). Morphological features and activation patterns of motor units. *J. Clin. Neurophysiol.* 12, 538–559. doi: 10.1097/00004691-199511000-00002

Ho, J., and Jow, R. (2009). *Characterization of High Temperature Polymer Thin Films for Power Conditioning Capacitors.* Adelphi, MD: Army Research Lab Adelphi MD Sensors and Electron Devices Directorate.

Ji, X., Liu, X., Cacucciolo, V., Imboden, M., Civet, Y., El Haitami, A., et al. (2019). An autonomous untethered fast soft robotic insect driven by low-voltage dielectric elastomer actuators. *Sci. Robot.* 4:eaaz6451. doi: 10.1126/scirobotics.aaz6451

Kellaris, N., Venkata, V. G., Rothemund, P., and Keplinger, C. (2019). An analytical model for the design of Peano-HASEL actuators with drastically improved performance. *Extreme Mech. Lett.* 29:100449. doi: 10.1016/j.eml.2019.100449

Kellaris, N., Venkata, V. G., Smith, G. M., Mitchell, S. K., and Keplinger, C. (2018). Peano-HASEL actuators: muscle-mimetic, electrohydraulic transducers that linearly contract on activation. *Sci. Robot.* 3:eaar3276. doi: 10.1126/scirobotics.aar3276

Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends Biotechnol.* 31, 287–294. doi: 10.1016/j.tibtech.2013.03.002

Koh, S. J. A., Keplinger, C., Kaltseis, R., Foo, C.-C., Baumgartner, R., Bauer, S., et al. (2017). High-performance electromechanical transduction using laterally-constrained dielectric elastomers part I: Actuation processes. *J. Mech. Phys. Solids* 105, 81–94. doi: 10.1016/j.jmps.2017.04.015

Lee, K. S., and Jung, M.-C. (2016). Three-dimensional finger joint angles by hand posture and object properties. *Ergonomics* 59, 890–900. doi: 10.1080/00140139.2015.1108458

Lee, K. S., and Jung, M. C. (2014). Flexion and extension angles of resting fingers and wrist. *Int. J. Occup. Saf. Ergonom.* 20, 91–101. doi: 10.1080/10803548.2014.11077038

Lee, K. S., Mo, S. M., Hwang, J., Wang, H., and Jung, M. C. (2008). Relaxed hand postures. *Jpn. J. Ergonom.* 44, 436–439. doi: 10.5100/jje.44.Supplement_436

Ly, K., Kellaris, N., McMorris, D., Johnson, B. K., Acome, E., Sundaram, V., et al. (2020). Miniaturized circuitry for capacitive self-sensing and closed-loop control of soft electrostatic transducers. *Soft Robotics*. doi: 10.1089/soro.2020.0048

Mitchell, S. K., Wang, X., Acome, E., Martin, T., Ly, K., Kellaris, N., et al. (2019). An easy-to-implement toolkit to create versatile and high-performance HASEL actuators for untethered soft robots. *Adv. Sci.* 6:1900178. doi: 10.1002/advs.201900178

Moretti, G., Duranti, M., Righi, M., Vertechy, R., and Fontana, M. (2018). "Analysis of dielectric fluid transducers," in *Electroactive Polymer Actuators and Devices (EAPAD) XX, vol. 10594* (Denver: International Society for Optics and Photonics), 105940W.

Murali, B., Huddle, S., and Weir, R. F. (2019). Design and evaluation of a distally actuated powered finger prosthesis with self-contained transmission for individuals with partial hand loss. *Adv. Mech. Eng.* 11:168781401983411. doi: 10.1177/1687814019834114

Ngo, T. D., Kashani, A., Imbalzano, G., Nguyen, K. T. Q., and Hui, D. (2018). Additive manufacturing (3D printing): a review of materials, methods, applications and challenges. *Compos. Part B Eng.* 143, 172–196. doi: 10.1016/j.compositesb.2018.02.012

Norton, R. (2000). *Machine Design (5th Edition): Amazon.com: Books.* Available online at: https://www.amazon.com/Machine-Design-5th-Robert-Norton/dp/013335671X (accessed April 24, 2020).

O'Brien, K. W., Xu, P. A., Levine, D. J., Aubin, C. A., Yang, H.-J., Xiao, M. F., et al. (2018). Elastomeric passive transmission for autonomous force-velocity adaptation applied to 3D-printed prosthetics. *Sci. Robot.* 3:eaau5543. doi: 10.1126/scirobotics.aau5543

Ortiz-Catalan, M., Håkansson, B., and Brånemark, R. (2014). An osseointegrated human-machine gateway for long-term sensory feedback and motor control of artificial limbs. *Sci. Transl. Med.* 6, 257re6. doi: 10.1126/scitranslmed.3008933

Park, Y. L., Chen, B. R., Pérez-Arancibia, N. O., Young, D., Stirling, L., Wood, R. J., et al. (2014). Design and control of a bio-inspired soft wearable robotic device for ankle–foot rehabilitation. *Bioinspir. Biomim.* 9:016007. doi: 10.1088/1748-3182/9/1/016007

Polygerinos, P., Correll, N., Morin, S. A., Mosadegh, B., Onal, C. D., Petersen, K., et al. (2017). Soft robotics: review of fluid-driven intrinsically soft devices; manufacturing, sensing, control, and applications in human-robot interaction. *Adv. Eng. Mater.* 19:1700016. doi: 10.1002/adem.201700016

Pourazadi, S., Shagerdmootaab, A., Chan, H., Moallem, M., and Menon, C. (2017). On the electrical safety of dielectric elastomer actuators in proximity to the human body. *Smart Mater. Struct.* 26:115007. doi: 10.1088/1361-665X/aa89b1

Rothemund, P., Kellaris, N., and Keplinger, C. (2019). How inhomogeneous zipping increases the force output of Peano-HASEL actuators. *Extre. Mech. Lett.* 31:100542. doi: 10.1016/j.eml.2019.100542

Rothemund, P., Kirkman, S., and Keplinger, C. (2020). Dynamics of electrohydraulic soft actuators. *Proc. Natl. Acad. Sci. U.S.A.* 117, 16207LP−16213LP. doi: 10.1073/pnas.2006596117

Scheme, E., and Englehart, K. (2011). Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *J. Rehabil. Res. Dev.* 48, 643–659. doi: 10.1682/JRRD.2010.09.0177

Schlatter, S., Illenberger, P., and Rosset, S. (2018). Peta-pico-voltron: an open-source high voltage power supply. *HardwareX* 4:e00039. doi: 10.1016/j.ohx.2018.e00039

Schunk, C., Pearson, L., Acome, E., Morrissey, T. G., Correll, N., Keplinger, C., et al. (2018). "System identification and closed-loop control of a hydraulically amplified self-healing electrostatic (HASEL) actuator," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE).

Sensinger, J. W., and Weir, R. F. (2008). User-modulated impedance control of a prosthetic elbow in unconstrained, perturbed motion. *IEEE Trans. Biomed. Eng.* 55, 1043–1055. doi: 10.1109/TBME.2007.905385

Shadwick, R. E., and Syme, D. A. (2008). Thunniform swimming: Muscle dynamics and mechanical power production of aerobic fibres in yellowfin tuna (*Thunnus albacares*). *J. Exp. Biol.* 211, 1603–1611. doi: 10.1242/jeb.013250

Smaby, N., Johanson, M. E., Baker, B., Kenney, D. E., Murray, W. M., and Hentz, V. R. (2004). Identification of key pinch forces required to complete functional tasks. *J. Rehabil. Res. Dev.* 41, 215–224. doi: 10.1682/JRRD.2004.02.0215

Suo, Z. (2010). Theory of dielectric elastomers. *Acta Mech. Solida Sin.* 23, 549–578. doi: 10.1016/S0894-9166(11)60004-9

Tan, D. W., Schiefer, M. A., Keith, M. W., Anderson, J. R., and Tyler, D. J. (2015). Stability and selectivity of a chronic, multi-contact cuff electrode for sensory stimulation in human amputees. *J Neural Engineering* 12. doi: 10.1088/1741-2560/12/2/026002

Wang, X., Mitchell, S. K., Rumley, E. H., Rothemund, P., and Keplinger, C. (2020). High-strain peano-HASEL actuators. *Adv. Funct. Mater.* 30:1908821. doi: 10.1002/adfm.201908821

Wu, L., de Andrade, M. J., Rome, R. S., Haines, C., Lima, M. D., Baughman, R. H., et al. (2015). "Nylon-muscle-actuated robotic finger," in *Active and Passive Smart Structures and Integrated Systems 2015, Vol. 9431* (San Diego, CA: International Society for Optics and Photonics), 94310I.

Conflict of Interest: NK, SKM, and CK are listed as inventors on patent applications PCT/US18/023797 and PCT/US19/020568 that cover fundamentals and basic designs of HASEL actuators. NK, SKM, and CK are co-founders of Artimus Robotics, a start-up company commercializing electrohydraulic HASEL actuators.

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

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 29 January 2021
doi: 10.3389/frobt.2020.586490

Model-Based Control and External Load Estimation of an Extensible Soft Robotic Arm

Xiaojiao Chen[1], *Dehao Duanmu*[1] *and Zheng Wang*[2]*

[1]Department of Mechanical Engineering, The University of Hong Kong, Hong Kong, China, [2]Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen, China

Soft robotics has widely been known for its compliant characteristics when dealing with contraction or manipulation. These soft behavior patterns provide safe and adaptive interactions, greatly relieving the complexity of active control policies. However, another promising aspect of soft robotics, which is to achieve useful information from compliant behavior, is not widely studied. This characteristic could help to reduce the dependence of sensors, gain a better knowledge of the environment, and enrich high-level control strategies. In this paper, we have developed a state-change model of a soft robotic arm, and we demonstrate how compliant behavior could be used to estimate external load based on this model. Moreover, we propose an improved version of the estimation procedure, further reducing the estimation error by compensating the influence of pressure deadzone. Experiments of both methods are compared, displaying the potential effectiveness of applying these methods.

Keywords: modeling, control, soft robot application, soft robot, soft arm

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Alejandro Suarez, Sevilla University, Spain
Luis Fernando Nagua, Universidad Carlos III de Madrid, Spain

***Correspondence:**
Zheng Wang
zheng.wang@ieee.org

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *23 July 2020*
Accepted: *14 December 2020*
Published: *29 January 2021*

Citation:
Chen X, Duanmu D and Wang Z (2021) Model-Based Control and External Load Estimation of an Extensible Soft Robotic Arm. Front. Robot. AI 7:586490. doi: 10.3389/frobt.2020.586490

1 INTRODUCTION

The realm of soft robotics is an ideal safe solution when dealing with collision and interaction due to compliant behavior Laschi et al. (2016); Majidi (2013); Kim et al. (2013). The properties of compliant behavior include intrinsic deformable structures Yi et al. (2018); Suarez et al. (2018), soft materials Yi et al. (2017); Polygerinos et al. (2015b); Wang et al. (2017), and backdrivable actuation methods. Various ways of achieving softness have been studied, including methods relying on compliant elements like SEA Pratt and Williamson (1995), memory effects like SMA Mohd Jani et al. (2014), dielectric elastomers like DEA O'Halloran et al. (2008), and pneumatic driven methods like PAMs Tondu and Lopez (2000) and pneu-nets Mosadegh et al. (2014). The realm of soft robotics has been actively inventing all kinds of soft machines to exploit their compliant nature in many aspects, such as soft arms that are safe to interact with Chen et al. (2017): Chen et al. (2018); Malzahn and Bertram (2014), soft fishes that swim naturally Marchese et al. (2014), soft gloves for rehabilitation Polygerinos et al. (2013, Polygerinos et al. (2015a), and soft hands that are versatile for handling objects Zhou et al. (2018); Zhou et al. (2019); Zhou et al. (2020).

The other potential use of softness is to gain valuable information from compliant behavior. There exist several examples that utilize compliant behavior to gain environmental information in the real world. For example, a human could estimate the weight of an object based on visual information of the deformation. The soft robots are also intelligence-embedded agents. They could not only handle local interaction compliantly but also store process information that may be helpful Laschi and Cianchetti (2014). One important aspect is the ability to estimate the force or load under interaction. However, it is not easy for soft robots to extract useful information from compliant behavior.

One way to achieve estimation from compliant behavior is to learn from data. In Wang and Wang (2020), the pressure information of a soft gripper was used to learn the external force. In Fang et al. (2019), local Gaussian regression was used to control and compensate for the external disturbance. However, the difficulty of using learning methods is a dependency on large data sets. Another limitation is that this method is specific to the design and structure of the soft arm, which makes it difficult for purposes of extension.

Another way is to establish models of the soft arm that involves external forces. However, it is not easy to achieve an accurate model for soft arms due to the softness of materials and complex description of the compliant body curves. Previously, most research has focused on kinematic models for controlling the soft arm statically based on the Constant Curvature assumption Jones and Walker (2006); Webster and Jones (2010); Bajo et al. (2011). Recently, there has been much improvement to the evolution of developing dynamic models for the soft arm. In Santina et al. (2019); Della Santina et al. (2020); Katzschmann et al. (2019); Wang et al. (2020), a dynamically-equivalent rigid robot is used to develop the dynamic model of the soft arm. Traditional rigid robot control methods could be well suited to this method. However, the difficulty of using this method is building an equivalent rigid robot in three-dimensional space faithfully. The resultant rigid counterpart is a hyper-redundant robot, and it is difficult to tackle. In Falkenhahn et al. (2015); Falkenhahn et al. (2017), the Euler-Lagrange method was used to derive the full dynamics. However, this method is quite complex and demands the accurate modeling of every part of the arm. In neither of these cases have these methods tackled the problem of estimating the external payload using their models.

In this paper, we have proposed a simplified analytical model and show how it could be used to control the arm and to extract loading information from compliant behavior. Our model captures the essential relationship between the pressure and the posture, establishing a preliminary relationship between the actuation space and the configuration space and providing a feed-forward control part. Based on this model, a state-change model is also derived by eliminating common modeling errors, and it is capable of estimating the external load from the change of bending angle. Furthermore, an improved method is given, accounting for the realistic pressure control deadzone and achieving a better estimation result with reduced error.

This paper does the following:

- We demonstrate the effectiveness of using a simplified model to control the soft arm in open loop.
- We propose a state-change model that avoids the negative mass problem.
- We are the first to consider the pressure deadzone effect, and we propose an improved method for the compensation, greatly improving the estimation result.
- We experimentally shown the performance of the state change model and the improved method.

The paper is organized as follows. First, the model and control of the soft robotic arm are given in **Section 2**. In **Section 3**, the state-change model and the improved method is derived and discussed. The experiments are analyzed in **Section 4**.

2 EXTENSIBLE SOFT ARM

2.1 Design of the Soft Arm

The soft robotic arm used in this paper is made up of six long bellows that have been installed in parallel. The relative positions of the bellows are constrained by two thin carbon-fiber plates, which avoids the potential buckling problem. Two acrylic plates are used as the connecting plates that force the bellows to share common starting and ending planes, as seen in **Figure 1A**.

The soft arm is actuated by inflating and vacuuming through on-off valves. The pressure distribution inside the six bellows controls the posture of the arm. When the pressures of the bellows was not equal, the arm would bend toward the direction of the smaller pressure sum. The greater the difference, the greater the degree of the bend, as seen in **Figure 1B**. The rotation around the vertical axis is achieved based on adjusting the direction of the pressure difference, and a full circle range of 360° could be achieved, as seen in **Figure 1C**. As a backbone-less arm, this extensible soft arm has a very large elongation ratio. In the free state, the soft arm has a length of 400 mm. When the six bellows are pressurized equally, the arm would elongate to a maximum length of more than 500 mm; when depressurized equally, the arm would contract to a minimal length of around 100 mm, as shown in **Figure 1D**.

A possible application for this soft arm is to lift heavy weights for people or act as a piece of massage equipment due to its large force and great compliance.

2.2 Modeling the Soft Arm

A simplified static model of the soft arm is given here to provide a feed-forward control part for preliminary control. The method considers the force balance equations of the three general coordinates, which are the elongation, the bending, and the rotations. The bellow actuators are modeled as cylinders with internal spring terms. The damping term is not considered due to the quasi-static motion assumption, and the mass terms are neglected because of the relatively small value in this soft arm. A detailed derivation process can be seen in our previous paper Chen et al. (2019). Here, we give a brief description of the modeling result since this will be the basis of the following external load estimation method.

In the elongation direction, the output force is simply the sum of all the six pressure-generated forces and the spring forces written as

$$F = \sum_{i=1}^{N} [P_i A - k(l_i - l_0)], \tag{1}$$

where F is the net output force, P_i is the bellow's internal gauge pressure, A is the cross section area of the bellow, k is the spring coefficient, l_0 is the original free length of the bellow, and l_i is the actual length of each bellow.

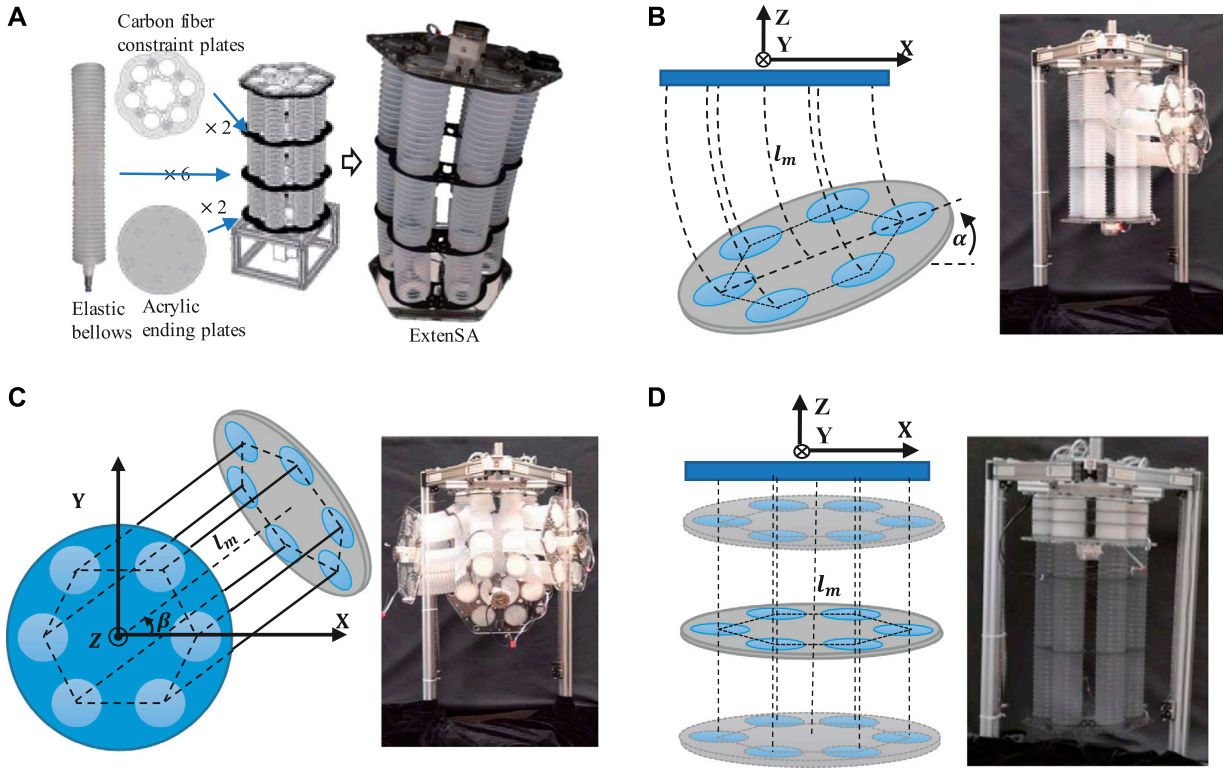


FIGURE 1 | (A) The Design concept of ExtenSA. **(B)** The bending motion of ExtenSA $\alpha = [0, 90°]$. **(C)** The Rotation Motion $\beta = [0, 360°]$. **(D)** The Elongation and contraction of ExtenSA $l = [100\,\text{mm}, 500\,\text{mm}]$.

In the bending and rotation direction, the torque generated by a certain bellow is given by its output force multiplied by its effective radius, given as

$$T_{\text{ext}}^{\beta} = \sum_{i=1}^{N} [F_i R \sin(\theta_i - \beta)], \tag{2}$$

$$T_{\text{ext}}^{\alpha} = \sum_{i=1}^{N} [F_i R \cos(\theta_i - \beta)], \tag{3}$$

where T is the torque, with the subscript α and β to represents the bending and rotation respectively. R is the distance between the bellows' center axis and the soft arm's axis, θ_i is the installation angle of the bellow in the X-Y plane, with the value of $\left[0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}\right]$ for the six bellows.

Combining **Eqs. 1, 2, 3**, we could express the configuration state of the soft arm by pressure information, which is given by

$$\begin{bmatrix} \alpha \\ \beta \\ L \end{bmatrix} = \begin{bmatrix} \dfrac{\frac{T_{ext}^{\alpha}}{AR} + \sqrt{\Phi_c^2 + \Phi_s^2 - \left(\frac{T_{ext}^{\beta}}{AR}\right)^2}}{C_1} \\ a\tan 2(\Phi_s, \Phi_c) - a\tan 2\left(\frac{T_{\text{ext}}^{\beta}}{AR}, \frac{T_{\text{ext}}^{\alpha}}{AR} - C_1\alpha\right) \\ \dfrac{A\Phi_p - F}{Nk} + l_0 \end{bmatrix}. \tag{4}$$

On the right side of **Eq. 4**, α represents the bending angle, β represents the rotation angle, and L represents the length of the central line of the soft arm. On the left side, Φ_c, Φ_s, and Φ_p are three weighted pressure sums related to the installation positions, and C_1 is a constant, given by

$$\Phi_c \overset{\text{def}}{=} \sum_{i=1}^{N} [P_i \cos\theta_i], \Phi_s \overset{\text{def}}{=} \sum_{i=1}^{N} [P_i \sin\theta_i], \Phi_p \overset{\text{def}}{=} \sum_{i=1}^{N} P_i, C_1 = \frac{NkR}{2A}.$$

In the case of no external load at the plate, the expression could be further simplified into

$$\begin{bmatrix} \alpha \\ \beta \\ L \end{bmatrix} = \begin{bmatrix} \dfrac{\sqrt{\Phi_c^2 + \Phi_s^2}}{C_1} \\ a\tan 2(\Phi_s, \Phi_c) - \pi \\ \dfrac{R\Phi_p}{2C_1} + l_0 \end{bmatrix}. \tag{5}$$

One important usage of the above model is to control the soft arm. The right side of the equation is pressure-related information, representing the actuation space of the soft arm, while the left side of the equation is the bending, rotation, and elongation of the soft arm, representing the configuration space. Therefore, this model relates the actuation space to the configuration space of the soft arm, providing feed-forward terms to the control algorithms.

However, as this model could predict the soft arm's movement to a certain degree, it is dangerous to use this model directly to estimate the external loads. This is because the unknown modeling error would be directly involved in the calculation, amplifying the estimation error, and, even worse, may cause the estimated mass to be negative.

We will therefore develop a state-change model based on this static model in **Section 3**, which would reduce the effect of the modeling error in the load estimation process.

2.3 Control of the Over-Actuated Soft Arm

The model could help to control the soft arm to the desired posture given commands like α, β, and L. However, for the ExtenSA with six actuation units, we currently only have three equations with elongation, bending, and rotation. Although we could add more constraints, such as adjusting the bending and rotating stiffness, the related equations would introduce unnecessary complexities. It is therefore meaningful to use only three input commands, that is, the length of the centerline L, bending angle α, and rotating angle β, to derive all the necessary pressure commands that we need.

Elongation movement is related to the pressure sums, and the bending and rotation are related to weighted pressure differences; given these constraints, we would like all the pressure commands to be as near the atmosphere as possible, without causing too much inflation or deflation.

Reorganize **Eq. 5**, we express the pressure related terms $[\Phi_c, \Phi_s, \Phi_p]$ in terms of the configuration states α, β, L, given by

$$\begin{bmatrix} \Phi_p \\ \Phi_c \\ \Phi_s \end{bmatrix} = \begin{bmatrix} \dfrac{2C_1(L-l_0)}{R} \\ -C_1 a \cos\beta \\ -C_1 \alpha \sin\beta \end{bmatrix}. \tag{6}$$

We formulated the procedure of solving the pressure commands from configuration commands as a quadratic optimization problem:

$$\begin{aligned} &\underset{X}{\text{minimize}} \ \boldsymbol{X}^T\boldsymbol{X} \\ &\text{subject to } A\boldsymbol{X} = b \end{aligned} \tag{7}$$

where $\boldsymbol{X} = [P_1, P_2, P_3, P_4, P_5, P_6]^T$ represent the pressure inside the bellows, and

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \cos\theta_1 & \cos\theta_2 & \cos\theta_3 & \cos\theta_4 & \cos\theta_5 & \cos\theta_6 \\ \sin\theta_1 & \sin\theta_2 & \sin\theta_3 & \sin\theta_4 & \sin\theta_5 & \sin\theta_6 \end{bmatrix},$$

$$b = \begin{bmatrix} \dfrac{2C_1(L-l_0)}{R} \\ -C_1\alpha\cos\beta \\ -C_1\alpha\sin\beta \end{bmatrix}.$$

This quadratic programming problem with linear constraints could be solved with a closed form result, which is given by

$$\boldsymbol{X} = A^{-1}\left(AA^{-1}\right)^{-1}b. \tag{8}$$

According to **Eq. 8**, given desired commands of the configuration space of the soft arm with L, α and β, the corresponding pressure commands in the actuation space could be obtained. By regulating the pressure commands, the soft arm could be controlled to the target position.

3 EXTERNAL LOAD ESTIMATION

When external loads are exerted, the posture of ExtenSA would be changing passively to a balanced new state, as shown in From **Figure 2**. The *alpha* and L would be changed and β unchanged.

The torque T^{α}_{ext} generated by the mass in the bending direction is written as

$$T^{\alpha}_{\text{ext}} = -\frac{mgL_L(1-\cos\alpha_L)}{\alpha_L}. \tag{9}$$

Assume the arm is unloaded at a certain configuration and that a mass m is then attached at the end of the ending plate. The mass exerts a bending torque and a pulling force affecting both α and L. In the following context, we will denote the modeling values as α_0, β_0, L_0 in unloaded situation and α_L, β_L, L_L in the loaded situation, respectively. We will also denote the true configuration states as $\alpha_0^T, \beta_0^T, L_0^T$ in the unloaded situation and $\alpha_L^T, \beta_L^T, L_L^T$ in the loaded situation, respectively, from measured values.

3.1 External Load Estimation with Original Model

Although the model **Eq. 4** provides a preliminary relation between the actuation and configuration space, it is unacceptable to estimate the external payload directly using the model because the modeling error may render the estimated mass to be negative.

Together with **Eq. 9**, to get the mass estimation we need to solve the equation

$$-\frac{mgL_L(1-\cos\alpha_L)}{\alpha_L} = AR\left(C_1\alpha_L - \sqrt{\Phi_c^2+\Phi_s^2}\right). \tag{10}$$

Giving the estimation as

$$m = \frac{AR\alpha_L\left(\sqrt{\Phi_c^2+\Phi_s^2} - C_1\alpha_L\right)}{gL_L(1-\cos\alpha_L)} \tag{11}$$

As modeling errors exist, it is possible that the term $\sqrt{\Phi_c^2+\Phi_s^2}$ from pressure feedback may be greater than the bending angle term $C_1\alpha_L$, and, if so, the resulting estimated mass may be a negative value. This method is highly sensitive to the sign and magnitude of the modeling error, which are both unknown in real cases. It is therefore dangerous to directly use the model to estimate the external payload.

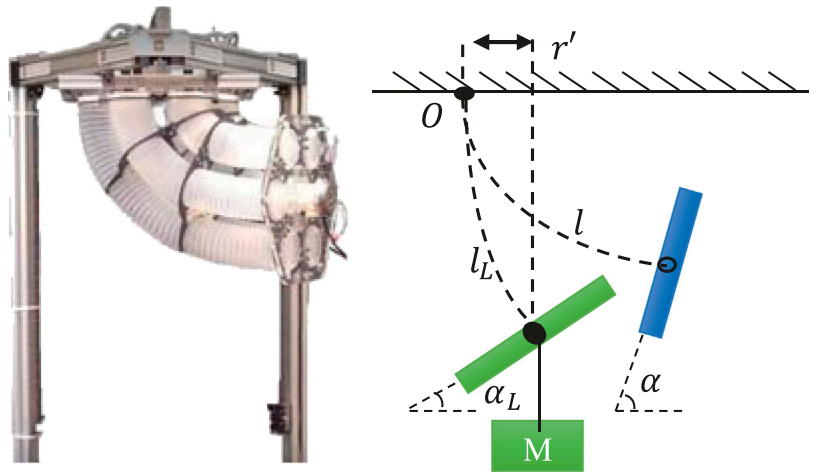


FIGURE 2 | Loading Geometry of ExtenSA. The external load would exert a bending torque around the center fixing point of the arm as well as a pulling force along the center line of the arm, affecting the bending angle α and the length l_m.

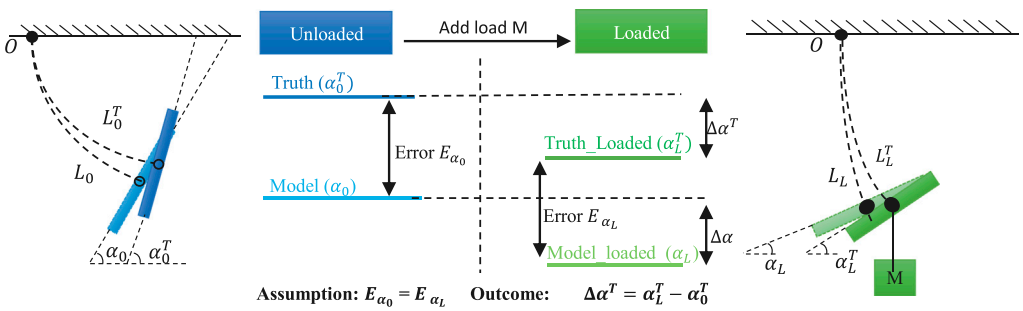


FIGURE 3 | Payload Estimation Illustration of ExtenSA. With the information of angle measurement, the loaded model could be used to approximate the external loads.

3.2 External Load Estimation with State-Change Model

Although we could not predict the exact model errors during every task, they all had the same model error in common, either from the friction or the characteristics of the materials. This means it would be more accurate to predict the state change rather than to directly predict the state. In other words, a model predicting the state change instead of predicting the state may lead to more trustworthy results through reducing the effect of common errors, as shown in **Figure 3**.

Based on this assumption that the mean modeling error of the model is constant during a periodic task, we would like to derive a state-change model to estimate the external load, that is, to use the change of bending angle $\Delta\alpha^T$ to estimate the payload, where

$$\Delta\alpha^T = \alpha_L^T - \alpha_0^T \stackrel{\text{def}}{=} \Delta A.$$

Here, we define the value of $\Delta\alpha^T$ obtained from this method as ΔA to distinguish it from the following improved method.

Looking into the process of loading, while the bending angle and length are changing, the pressures inside the bellows would not change due to low-level pressure feedback control if all the pressure commands are well controlled within a relatively small error region. Then, the three quantities, Φ_c, Φ_s, and Φ_p, could be considered unchanged during the process. Then we get the following equation:

$$-\frac{mgL_L(1-\cos\alpha_L)}{AR\alpha_L} - c_1\alpha_L = -c_1\alpha_0. \tag{12}$$

To simplify the calculation, the approximation of $1-\cos\alpha_L \approx (\alpha_L^2/2)$ and $L_L \approx L_0$ were applied to obtain α_L, leading to

$$\alpha_L = \frac{2ARC_1}{2ARC_1 + mgL_0}\alpha_0. \tag{13}$$

On the other hand, the approximation of $\alpha_0 \approx \alpha_L$ was used to obtain the L_L, leading to

$$L_L = \frac{mg\cos\alpha_0}{Nk} + L_0 \tag{14}$$

These two equations describe the state change of the ExtenSA, which could be used to compensate for the change of α and L due to external loads.

According to **Eq. 13**, the equation could be rewritten as

$$\frac{\alpha_L}{\alpha_0} = \frac{2ARC_1}{2ARC_1 + mgL_0}. \tag{15}$$

Giving m as

$$m = \frac{2ARC_1}{gL_0}\left(\frac{\alpha_0 - \alpha_L}{\alpha_L}\right) = -\frac{2ARC_1}{gL_0\alpha_L}\Delta\alpha^T. \tag{16}$$

This equation avoids the involvement of pressure information, guaranteeing the acquisition of a positive estimation of the payload because the $\Delta\alpha^T$ is always smaller than zero in real cases due to the external load. Therefore, using this equation, we are able to obtain an approximation of the external loads with a more trustworthy result.

Moreover, this equation not only applies to the situation where the soft arm is in a steady state. For a soft arm in a cyclic motion, such as when following a sinusoidal trajectory, this equation also works, with the value of *alpha* referring to the moving mean value within at least one motion period.

3.3 External Load Estimation with Improved Method

However, in reality, previous payload estimation methods may suffer from the ExtenSA's pressure changing during the loading process, which is due to the pressure control deadzone, rendering decreased accuracy of the payload estimation. In this section, a modified method for payload estimation is proposed to improve the accuracy.

3.3.1 Control Dilemma of Pressure Deadzone

The previous payload estimation method is based on the ideal assumption that pressures could be accurately regulated by the controllers. However, in real applications, most pressure feedback control has a control deadzone to avoid oscillation within which the pressure is regarded as unchanged. If the deadzone is set too large, the tracking performance would deteriorate, and the steady error would be large. If the deadzone is set too small, the system would go oscillating because of the limitation of the actuation valves' switching frequency. Therefore, the width of the pressure deadzone is commonly set according to the application requirement and the platform capability. Typically, for most soft robotic applications, the pressure deadzone is set to be between 1 KPa and 2 KPa. This is mainly due to the valve's limited switching frequency, pneumatic control delay from tube transmission, the sensor's precision capability, and the pressure's sensitivity to small volumes or temperature changes.

In ExtenSA, the pressure deadzone is set to 2 KPa to get a steady pneumatic control without oscillation by comparing many experimental results. This is mainly due to the thin tubes used for each bellow; they cause pneumatic control delay and limited sensor precision of around 1 KPa.

3.3.2 Estimation Error from Pressure Deadzone

The existence of a pressure deadzone would influence the soft arm's behavior when an external payload is exerted, inducing an error in the estimation result.

For example, in a certain working scenario, such as keeping the ExtenSA at a particular bending angle, when external loads are exerted, the pressure inside the actuators tends to change due to the deformation of bellows. However, if the pressure change is within the pressure deadzone, then the pressure controllers will not be triggered. The valves will not open, resulting in closed chambers and causing the actual pressure to either rise or fall, as shown in **Figure 4B**.

Only when the pressure change is out of the pressure deadzone will the pressure controller take action, but it will do so only to regulate the pressure to one boundary of the deadzone.

In either case, a repelling pressure change is observed, which would result in two consequences. First, the actual change of α due to external loads would be smaller than the ideal change when pressures are perfectly regulated. This is because of the repelling pressure behavior, which provides an opposing bending torque. Second, the modeled α from measured pressures is to increase. This procedure is depicted in the third column in **Figures 4A**, **5**. Due to the pressure deadzone, the measured α_{LR}^T is larger than expected, and the modeled α_{0R} is also larger than expected. If the previous state-change payload estimation method is used in 16, there will be an error induced by the deadzone.

3.3.3 Improved Method by Using Change of Error

The state-change model could be regarded as using the change-of-truth $\Delta\alpha^T = \Delta A$ to estimate the payload. But due to the pressure deadzone, in reality, the change of the truth may not be solely from the exerting of payload but also from the deadzone effect. Therefore, the real change $\Delta\alpha_R^T$ is not equal to $\Delta\alpha^T$.

The modified method reduces the influence of deadzone by using the change of the error between the truth and the modeled unloaded value, instead of ΔA, as the input to the state-change model. The error E is defined as the difference between the measured angle with the modeled angle(unloaded). At first, in free state, the error is. After loading, the error becomes $E_{\alpha_{LR}} = \alpha_L^T - \alpha_{LR}$. We then get the improved estimation of $\Delta\alpha^T$:

$$\Delta\alpha^T = E_{\alpha_L} - E_{\alpha_0} \overset{\text{def}}{=} \Delta E.$$

We define the result of $\Delta\alpha^T$ derived using the improved method as ΔE to distinguish it from the previous state-change model where $\Delta\alpha^T = \Delta A$. Therefore, using $\Delta\alpha^T = \Delta E$ as the input to **Eq. 16** can decrease the influence of the deadzone-induced error and improve the accuracy of payload estimation results.

4 EXPERIMENTS

In this section, the experimental results of the model-based control, external load estimation, and improved version are demonstrated. A dedicated embedded pneumatic control platform was built for these experiments. The control board is a STM32F767ZI NUCLEO board from STMicroelectronics with a core frequency of 216 MHz. It could generate 12-channel individual PWM to control 12 solenoid valves SX12F-DG that could operate at a maximum of 350 Hz. Two pumps are used as sources of pressurized air and

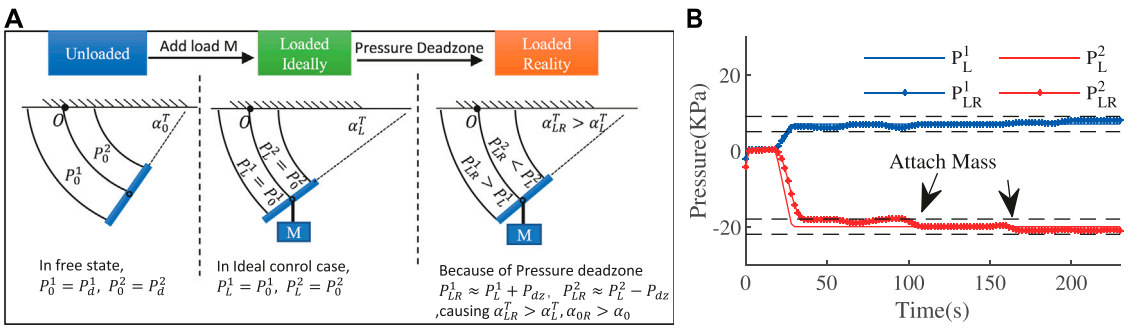


FIGURE 4 | The pressure deadzone results in increased bending angle and modeled value in loaded situation.

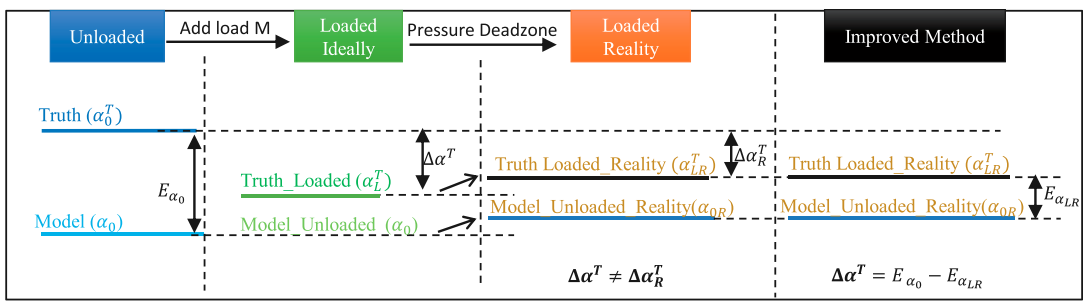


FIGURE 5 | Improved Payload Esitmation Method of ExtenSA. The existence of pressure control deadzone would cause the true value α_L^T and modeled value α_0 to change slightly, inducing an estimation error. The improved method is to use $\Delta\alpha^T = E_{\alpha_0} - E_{\alpha_{LR}}$, i.e., the change of error between the truth and modeled value instead of just using the change of truth, as the input to the state-change model. This would help reduce the influence of the deadzone induced state change.

vacuum. The overall embedded pneumatic control platform could regulate the pressure from range −100 KPa to 200 KPa with a deadzone set to 2 KPa. A remote PC communicates with the embedded platform through the serial interface at an updating frequency of 1 KHz, on which a dedicated GUI written in Python was used to display the soft arm's status and accept input commands. The soft arm has an IMU at the ending plate and a wire sensor through the central axis, which is used to gain the feedback information of L, α, and β. A six-axis force sensor from ATI was installed at the mounting point where the external weight is attached as a reference. The platform is shown in **Figure 6A**.

4.1 Model-Based Control

In this section, we show that the model could be used to op-loop control the soft arm by providing the feedback part.

A joystick was used as the command input device, which gives commands to the soft arm to elongate, bend, and rotate. These commands of the configuration space are used to get pressure commands by **Eq. 8**. The generated pressure commands are shown in **Figure 6B**. The soft arm tracks these desired α, β, and L simultaneously.

The result is shown in **Figure 6C**. The pure open-loop control has achieved a moderate tracking performance. This would help to control soft arms to maintain high active compliance by allowing for smaller feedback gains.

4.2 Estimation Result Using State-Change Model

The first experiment is based on tracking sinusoidal signals of bending angle as shown in **Figure 7A**. Since this is not a static situation, we need to use the moving mean instead. The moving mean of α^T within a time window of one motion period (100 s) was plotted in **Figure 7B**. α_0^T represents the mean value of the first 100 s representing the free state. Therefore, the value $\Delta\alpha^T = \alpha_L^T - \alpha_0^T$ could be obtained.

The result of external loads approximation was shown in **Figure 7C**. The result was capable of being stabilized after one motion period, which is just the time for the stabilization of the mean of α_L^T. With a smaller motion period, this stabilization time would be reduced. The error of load approximation was plotted in **Figure 7D**. The error is large during the transition period, and, after around one period, the error is around 8% in the case of 0.5 Kg and 37% in the case of 0.9 Kg.

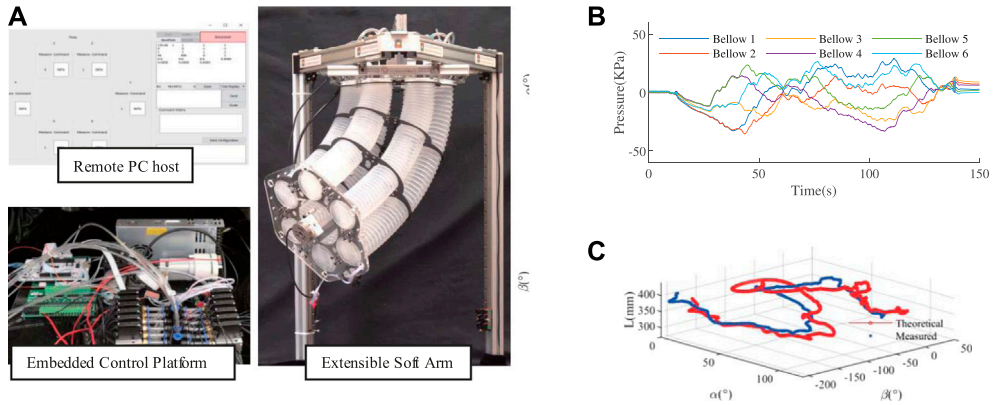


FIGURE 6 | (A) The experimental platform set-up consists of a remote PC host to display and give commands, an embedded pneumatic control platform to regulate 12-channel pressure and the soft arm equipped with an IMU, a length sensor, and a force sensor. **(B)** The optimization generated commands.**(C)** The simultaneous control of α, β, and *L* based on the model.

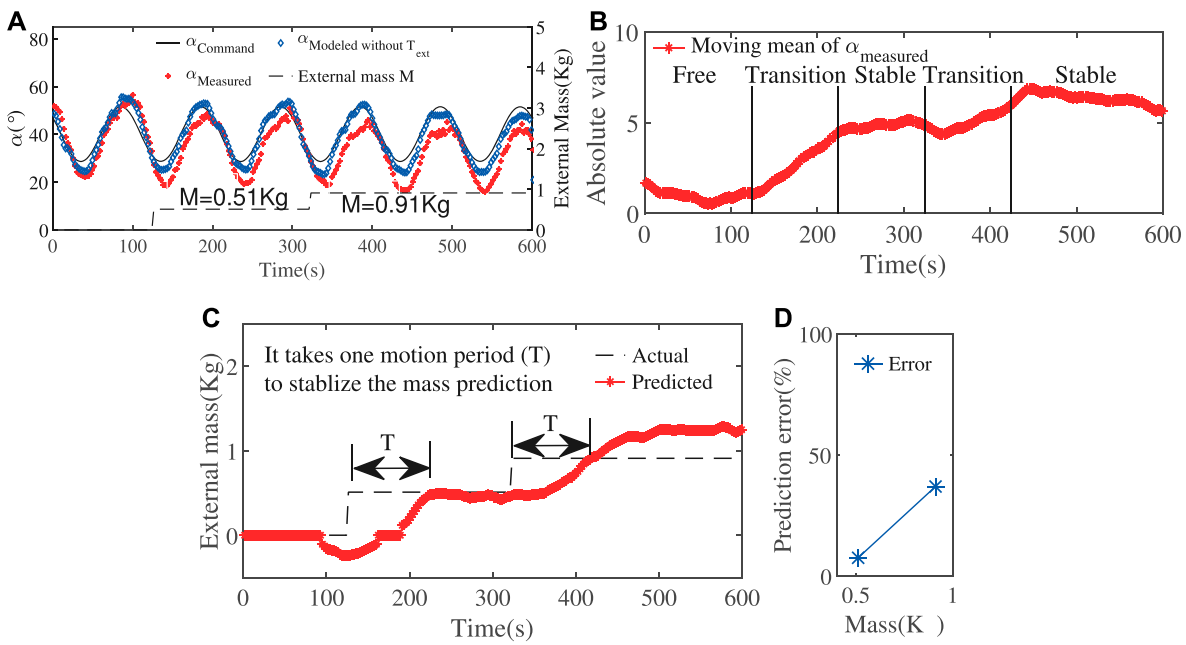


FIGURE 7 | (A) Sinusoidal movement of the bending angle with increasing loads. **(B)**The moving mean of the bending angle α_L^T is calculated within a time window of one motion period. **(C)** The external loads could be successfully approximated after one motion period. **(D)** The approximation error in steady state.

4.3 Estimation Result with Improved Methods

In this section, we use the improved method to estimate the external load and compare it with the result from the state-change model.

In the following experiments, the ExtenSA was set to a constant bending angle of $\alpha = 57°$ at the beginning. The length was set to be a constant $L = 0.35$ m. Then external loads were added to the endplate of the ExtenSA at the time $t = 68$ s and $t = 103$ s. The change of α was plotted in **Figure 8A**.

In the beginning, the measured angle α_{Measured} (α_0^T) (red line) was around 60°, slightly larger than the command, and the model

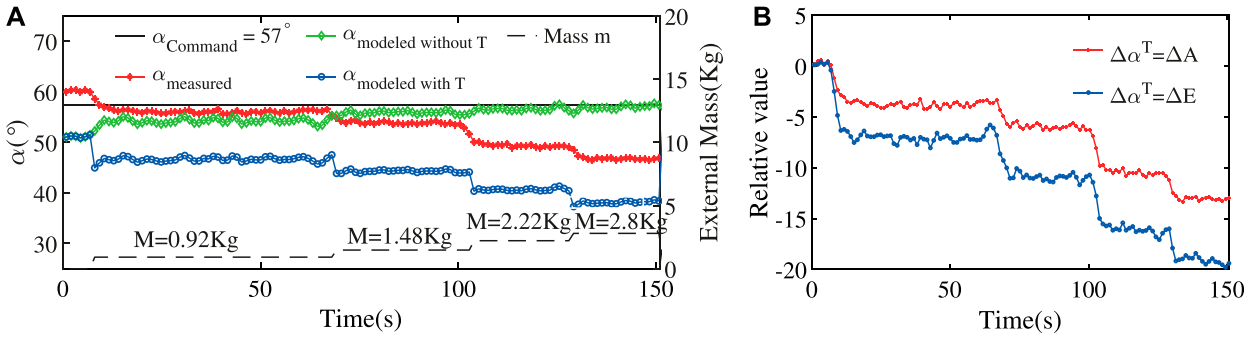


FIGURE 8 | Loading Test at Constant Position of ExtenSA. **(A)** The α change during a loading test where the initial command was given at $\alpha_{command}$ = 57° and the length was at l_m = 0.35 m. Different external loads were added to the end of the ExtenSA, resulting in a change of α. **(B)** The relative value of *A* and *E* with respect to their initial state shows that Δ*E* displays a more significant change.

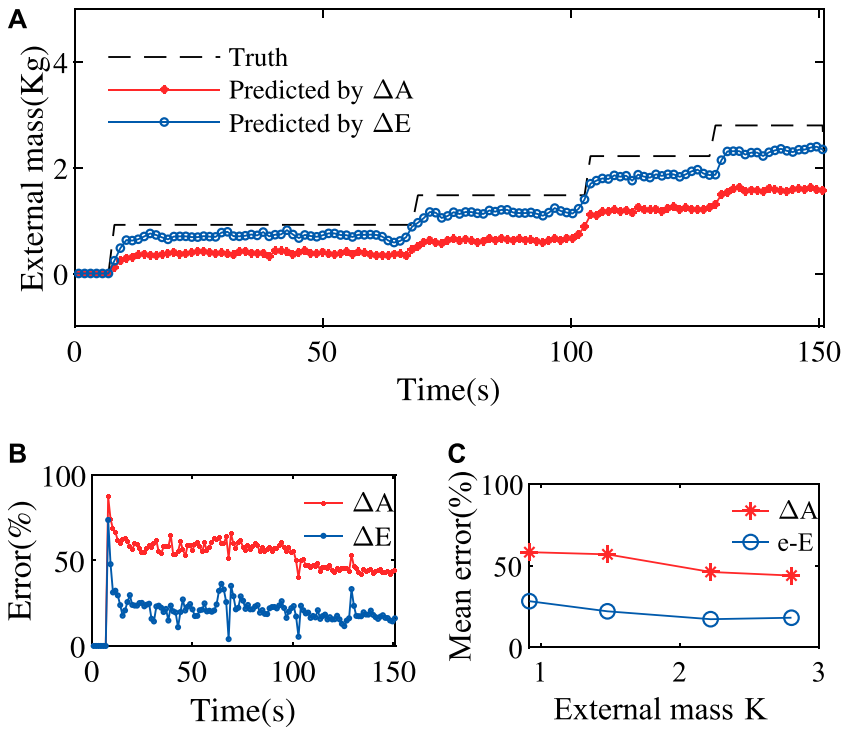


FIGURE 9 | Improved Payload Estimation Results of ExtenSA. **(A)** The payload estimation method using Δ*E* shows a better estimation performance than using Δ*A*. **(B)** The estimation error was decreased from around 60% to around 20% by using Δ*E* instead of Δ*A*. **(C)** The estimation error would be smaller for larger weights because the resulting compliant behavior would have a larger signal-noise ratio.

predicted angle $\alpha_{Modeled\_without\_T_{ext}}$ α_0 (green line) was around 52°. The difference between them is the free state model error E_{α_0}.

After the first loading of a mass of 0.92 Kg at t = 68 s, α was dropped to a new value of around 56° (α_{LR}) under the effect of pressure deadzone. In the meantime, the deadzone causes α_0 to increase to around 55° (α_{0R}). The same happened when a succeeding load was exerted at t = 103 s.

The comparison between Δ*E* and Δ*A* was plotted in **Figure 8B**. It could be seen that, Δ*E* shows a more significant change than Δ*A* under loading change, suggesting that they could be used as a better signal to calculate $\Delta\alpha^T$.

The estimation result using the Δ*E* and Δ*A* is given in **Figure 9A**. Since this is a static posture, the related values are the real-time values without taking means. The result showed that using Δ*E* would produce a more accurate approximation of external load than using Δ*A*. Furthermore, the stabilization time for the estimation procedure is only around several

seconds, much smaller than previous experiments where a whole motion period time is needed.

The estimation errors was plotted in **Figures 9B,C**. As seen, the overall estimation error is undergoing a slowly reducing process as the external mass is increasing. The estimation error was reduced from around 60% by directly using the state-change model to around 20% by using the improved method, testifying to the effectiveness of the improved method.

4.4 Discussion

The state-change model uses the change of bending angle $\Delta\alpha^T = \alpha_L^T - \alpha_0^T \overset{\text{def}}{=} \Delta A$ to estimate the mass. This method avoids the potential negative estimation outcome by directly using the model **Eq. 4**. This method is based on the idea that the modeling error is nearly constant in the free and loaded situation, and thus the difference would reduce the effect of the common modeling error. This method also assumes that the low-level pressure control is ideal, keeping the weighted pressure quantities Φ_c, Φ_s unchanged during the process. Yet, in reality, this assumption could only be loosely met because of the existence of pressure control deadzone. Nevertheless, this method still gives a moderate approximation of the external load. In applications where the pressure deadzone is smaller, the approximation result would be better.

The improved method uses the change of error $\Delta\alpha^T = E_{\alpha_L} - E_{\alpha_0} \overset{\text{def}}{=} \Delta E$ as a better indicator of $\Delta\alpha^T$ in the existence of pressure control deadzone. This method compensated for the change of the modeled value and the inadequate change of the actual angle, both caused by the pressure deadzone. Therefore, this method would give a better approximation than the state-change model when the pressure deadzone is affecting the loading behavior. In cases of small pressure deadzone effects, this method would be reduced to the state-change model naturally. Therefore, it is always a better choice to use this method to estimate the external payload.

5 CONCLUSION

In this paper, the possibility and effectiveness of using a simplified analytical model to retrieve external load information are studied. The main idea is to use the state-change model to eliminate the common errors of the modeling part and improve the estimation accuracy by considering practical pressure control deadzones. The promising aspect of utilizing this kind of method is in situations where only limited sensor information is provided or could be economically got. As the soft robotics lack proper sensors and rely on their intrinsic compliance to deal with uncertainty, our state-model-based method, which tries to extract information from this masked behavior, would provide economic guidance for high-level planning.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

XC brought up all the estimation methods and conducted all the experiments in this paper. DD took part in the design and fabrication of the soft arm. ZW provided the guidance and funding for the project.

FUNDING

This work was jointly supported by NSFC Grant 51975268, Hong Kong ITF Grant ITS/457/17FP, ITS/305/19FP, SUSTECH-AISONO Joint Lab Grant, and SUSTECH Edcucation Endowment.

ACKNOWLEDGMENT

We would like to thank the Innovation and Technology Fund in Hong Kong for the support of conducting this research.

REFERENCES

Bajo, A., Goldman, R. E., and Simaan, N. (2011). "Configuration and joint feedback for enhanced performance of multi-segment continuum robots," in Proceedings-IEEE international conference on robotics and automation, Shanghai, China, May 9–13, 2011 (New York, NY: IEEE), 2905–2912. doi:10.1109/ICRA.2011.5980005

Chen, X., Guo, Y., Duanmu, D., Zhou, J., Zhang, W., and Wang, Z. (2019). Design and modeling of an extensible soft robotic arm. *IEEE Robot. Autom. Lett.* 4, 4208–4215. doi:10.1109/LRA.2019.2929994

Chen, X., Peng, J., Zhou, J, Chen, Y., Wang, M. Y., and Wang, Z. (2017). "A robotic manipulator design with novel soft actuators," in IEEE international conference on robotics and automation (ICRA), Singapore, May 29, 2017 (New York, NY: IEEE). doi:10.1109/ICRA.2017.7989220

Chen, X., Yi, J., Li, J., Zhou, J., and Wang, Z. (2018). Soft-Actuator-based robotic joint for safe and forceful interaction with controllable impact response. *IEEE Robot. Autom. Lett.* 3, 3505–3512. doi:10.1109/LRA.2018.2854409

Della Santina, C., Katzschmann, R. K., Bicchi, A., and Rus, D. (2020). Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment. *Int. J. Robot Res.* 39, 490–513. doi:10.1177/0278364919897292

Falkenhahn, V., Hildebrandt, A., Neumann, R., and Sawodny, O. (2017). Dynamic control of the bionic handling assistant. *IEEE ASME Trans. Mechatron.* 22, 6–17. doi:10.1109/TMECH.2016.2605820

Falkenhahn, V., Mahl, T., Hildebrandt, A., Neumann, R., and Sawodny, O. (2015). Dynamic modeling of bellows-actuated continuum robots using the euler-Lagrange formalism. *IEEE Trans. Robot.* 31, 1483–1496. doi:10.1109/TRO.2015.2496826

Fang, G., Wang, X., Wang, K., Lee, K. H., Ho, J. D., Fu, H. C., et al. (2019). Vision-based online learning kinematic control for soft robots using local Gaussian process regression. *IEEE Robot. Autom. Lett.* 4, 1194–1201. doi:10.1109/LRA.2019.2893691

Jones, B. A., and Walker, I. D. (2006). Kinematics for multisection continuum robots. *IEEE Trans. Robot.* 22, 43–55. doi:10.1109/TRO.2005.861458

Katzschmann, R. K., Santina, C. D., Toshimitsu, Y., Bicchi, A., and Rus, D. (2019). Dynamic motion control of multi-segment soft robots using piecewise constant

curvature matched with an augmented rigid body model. 2019 2nd IEEE international conference on soft robotics (robosoft), Seoul, Korea, April 14–18, 2019 (New York, NY: IEEE), 454–461. doi:10.1109/ROBOSOFT.2019.8722799

Kim, S., Laschi, C., and Trimmer, B. (2013). Soft robotics: a bioinspired evolution in robotics. *Trends Biotechnol.* 31, 287–294. doi:10.1016/j.tibtech.2013.03.002

Laschi, C., and Cianchetti, M. (2014). Soft robotics: new perspectives for robot bodyware and control. *Front. Bioeng. Biotechnol.* 2, 3. doi:10.3389/fbioe.2014.00003

Laschi, C., Mazzolai, B., and Cianchetti, M. (2016). Soft robotics: technologies and systems pushing the boundaries of robot abilities. *Sci. Robot.* 1, eaah3690. doi:10.1126/scirobotics.aah3690

Majidi, C. (2013). Soft robotics: a perspective-current trends and prospects for the future. *Soft Robot.* 1, 5–11. doi:10.1089/soro.2013.0001

Malzahn, J., and Bertram, T. (2014). Collision detection and reaction for a multi-elastic-link robot arm. *IFAC Proceedings Volumes* 47, 320–325. doi:10.3182/20140824-6-za-1003.01545

Marchese, A. D., Onal, C. D., and Rus, D. (2014). Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators. *Soft Robot.* 1, 75–87. doi:10.1089/soro.2013.0009

Mohd Jani, J., Leary, M., Subic, A., and Gibson, M. A. (2014). A review of shape memory alloy research, applications and opportunities. *Mater. Des.* 56, 1078–1113. doi:10.1016/j.matdes.2013.11.084

Mosadegh, B., Polygerinos, P., Keplinger, C., Wennstedt, S., Shepherd, R. F., Gupta, U., et al. (2014). Pneumatic networks for soft robotics that actuate rapidly. *Adv. Funct. Mater.* 24, 2163–2170. doi:10.1002/adfm.201303288

O'Halloran, A., O'Malley, F., and McHugh, P. (2008). A review on dielectric elastomer actuators, technology, applications, and challenges. *J. Appl. Phys.* 104, 071101. doi:10.1063/1.2981642

Polygerinos, P., Lyne, S., Wang, Z., Fernando, L., Mosadegh, B., Whitesides, G. M., et al. (2013). Towards a soft pneumatic glove for hand rehabilitation. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, November 3–7, 2013 (New York, NY: IEEE), 1512–1517

Polygerinos, P., Wang, Z., Galloway, K. C., Wood, R. J., and Walsh, C. J. (2015a). Soft robotic glove for combined assistance and at-home rehabilitation. *Robot. Autonom. Syst.* 73, 135–143. doi:10.1016/j.robot.2014.08.014

Polygerinos, P., Wang, Z., Overvelde, J. T., Galloway, K. C., Wood, R. J., Bertoldi, K., et al. (2015b). Modeling of soft fiber-reinforced bending actuators. *IEEE Trans. Robot.* 31, 778–789. doi:10.1109/TRO.2015.2428504

Pratt, G. A., and Williamson, M. M. (1995). Series elastic actuators. *IEEE Inter. Conf. Intel. Robot. Sys.* 1, 399–406. doi:10.1109/iros.1995.525827

Santina, C. D., Bicchi, A., and Rus, D. (2019). Dynamic control of soft robots with internal constraints in the presence of obstacles. IEEE international conference on intelligent robots and systems, Macau, China, November 3–8, 2019 (New York, NY: IEEE), 6622–6629. doi:10.1109/IROS40897.2019.8967668

Suarez, A., Heredia, G., and Ollero, A. (2018). Physical-virtual impedance control in ultralightweight and compliant dual-arm aerial manipulators. *IEEE Robot. Autom. Lett.* 3, 2553–2560. doi:10.1109/LRA.2018.2809964

Tondu, B., and Lopez, P. (2000). Modeling and control of McKibben artificial muscle robot actuators. *IEEE Contr. Syst. Mag.* 20, 15–38. doi:10.1109/37.833638

Wang, C., Frazelle, C. G., Wagner, J. R., and Walker, I. (2020). Dynamic control of multi-section three-dimensional continuum manipulators based on virtual discrete-jointed robot models. *IEEE ASME Trans. Mechatron.* 4435, 1. doi:10.1109/TMECH.2020.2999847

Wang, L., and Wang, Z. (2020). Mechanoreception for soft robots via intuitive body cues. *Soft Robot.* 7, 198–217. doi:10.1089/soro.2018.0135

Wang, Z., Polygerinos, P., Overvelde, J. T. B., Galloway, K. C., Bertoldi, K., and Walsh, C. J. (2017). Interaction forces of soft fiber reinforced bending actuators. *IEEE ASME Trans. Mechatron.* 22, 717–727. doi:10.1109/TMECH.2016.2638468

Webster, R. J., and Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots: a review. *Int. J. Robot Res.* 29, 1661–1683. doi:10.1177/0278364910368147

Yi, J., Chen, X., Song, C., and Wang, Z. (2018). Fiber-reinforced origamic robotic actuator. *Soft Robot.* 5, 81–92. doi:10.1089/soro.2016.0079

Yi, J., Chen, X., and Wang, Z. (2018). A three-dimensional-printed soft robotic glove with enhanced ergonomics and force capability. *IEEE Robot. Autom. Lett.* 3, 242–248. doi:10.1109/LRA.2017.2737481

Zhou, J., Chen, Y., Chen, X., Wang, Z., Li, Y., and Liu, Y. (2020). A proprioceptive bellows (PB) actuator with position feedback and force estimation. *IEEE Robot. Autom. Lett.* 5, 1867–1874. doi:10.1109/LRA.2020.2969920

Zhou, J., Chen, X., Chang, U., Lu, J.-T., Leung, C. C. Y., Chen, Y., et al. (2019). A soft-robotic approach to anthropomorphic robotic hand dexterity. *IEEE Access* 7, 101483–101495. doi:10.1109/ACCESS.2019.2929690

Zhou, J., Yi, J., Chen, X., Liu, Z., and Wang, Z. (2018). BCL-13: a 13-DOF soft robotic hand for dexterous grasping and in-hand manipulation. *IEEE Robot. Autom. Lett.* 3, 3379–3386. doi:10.1109/LRA.2018.2851360

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.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 03 March 2021
doi: 10.3389/frobt.2021.606938

A Modular Geometrical Framework for Modelling the Force-Contraction Profile of Vacuum-Powered Soft Actuators

Samuel Dutra Gollob[1], *Clara Park*[1], *Bon Ho Brandon Koo*[1] *and Ellen T. Roche*[1,2]*

[1]Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, United States, [2]Institute for Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA, United States

In this paper, we present a generalized modeling tool for predicting the output force profile of vacuum-powered soft actuators using a simplified geometrical approach and the principle of virtual work. Previous work has derived analytical formulas to model the force-contraction profile of specific actuators. To enhance the versatility and the efficiency of the modelling process we propose a generalized numerical algorithm based purely on geometrical inputs, which can be tailored to the desired actuator, to estimate its force-contraction profile quickly and for any combination of varying geometrical parameters. We identify a class of linearly contracting vacuum actuators that consists of a polymeric skin guided by a rigid skeleton and apply our model to two such actuators-vacuum bellows and Fluid-driven Origami-inspired Artificial Muscles-to demonstrate the versatility of our model. We perform experiments to validate that our model can predict the force profile of the actuators using its geometric principles, modularly combined with design-specific external adjustment factors. Our framework can be used as a versatile design tool that allows users to perform parametric studies and rapidly and efficiently tune actuator dimensions to produce a force-contraction profile to meet their needs, and as a pre-screening tool to obviate the need for multiple rounds of time-intensive actuator fabrication and testing.

Keywords: soft robotics, numerical model, artificial muscle, virtual work, vacuum-powered soft actuator

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Wenlong Zhang, Arizona State University Polytechnic campus, United States
Lisbeth Mena, Universidad Carlos III de Madrid, Spain

***Correspondence:**
Ellen T. Roche
etr@mit.edu

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *16 September 2020*
Accepted: *15 January 2021*
Published: *03 March 2021*

Citation:
Gollob SD, Park C, Koo BHB and Roche ET (2021) A Modular Geometrical Framework for Modelling the Force-Contraction Profile of Vacuum-Powered Soft Actuators. Front. Robot. AI 8:606938. doi: 10.3389/frobt.2021.606938

INTRODUCTION

Soft robotics is a growing field, owing somewhat to an increasing demand for machines that can interact more safely with humans and their environment, generate complex multi-degree-of-freedom motions, and resist impact damage (Rus and Tolley, 2015). The sub-field of soft artificial muscles is relevant, as they are commonly used to actuate soft robots as opposed to traditional motors. Although a variety of artificial muscle actuation schemes have been developed, including shape-memory alloys (Kim et al., 2009), tension cables (Calisti et al., 2011), and phase transitions (Miriyev et al., 2017), fluidic actuation is widely used, as it is compatible with soft matrices, with programmed fluidic channels, and provides a means to increase actuator volume and effective stiffness analogous with the contraction and stiffening of biological muscle (Rus and Tolley, 2015). Pneumatic artificial muscles have been used in, and theorized for, a range of applications, from medical implantable devices (Roche et al., 2017; Mac

Murray et al., 2018), to exoskeletons (Polygerinos, et al., 2015a; Porter et al., 2020), and both soft and rigid robotic applications (Andrikopoulos et al., 2011).

Most soft pneumatic actuators described in the literature operate with positive pressure, often involving a section of the actuator which expands with pressure and a strain-limiting component which guides the elastic expansion in a desired direction. This duality has been achieved by creating geometrical asymmetry in elastomeric actuators (Ogura et al., 2009; Katzschmann et al., 2016), introducing an off-axis strain-limiting material for bending motions (Martinez et al., 2012; Mosadegh et al., 2014), and reinforcing the outer skin of the actuator with fibers (Chou and Hannaford, 1996; Connolly et al., 2015; Deimel and Brock, 2016; Wirekoh and Park, 2017). Although positive pressure actuators can produce complex motions and large forces (Rus and Tolley, 2015), they have limited contraction ratios, high actuation pressure requirements, and are subject to delamination or bursting (Chou and Hannaford, 1996; Sanan et al., 2014; Niiyama et al., 2015; Li et al., 2017). Owing to their dependence on volume increase for contraction, they pose a design challenge for applications where space is constrained.

Vacuum-operated soft pneumatic actuators are an alternative to positive pressure actuators that can avoid some of these pitfalls, while still achieving similar bending (Robertson and Paik, 2018; Tawk, 2018), linear (Yang et al., 2016; Yang et al., 2017; Li et al., 2017; Li et al., 2019; Felt et al., 2018; Lee and Rodrigue, 2019), and complex programmed motions (Li et al., 2017; Jiao et al., 2019). This class of actuators rely on a decreasing volume for actuation, in contrast to positive pressure actuators where the volume typically increases upon actuation. Similar to the strain-limiting operating principle for positive pressure actuators, vacuum actuators often involve a thin strain-limited "skin" that is responsible for a decrease in volume upon actuation, and a "skeleton" that limits compression to guide the volume decrease in a desired direction (Li et al., 2017; Felt et al., 2018; Tawk et al., 2018; Lee and Rodrigue, 2019). For the purpose of this paper, this type of vacuum actuator will be referred to as a "skin-skeleton vacuum actuator." Particularly, this work focuses on skin-skeleton actuators that undergo linear contraction upon actuation. This class of actuators has achieved contraction ratios near or above 90%, is often lightweight, fast-moving, and resistant over many cycles, requires low actuation pressures, and produces a high power to weight ratio compared to positive pressure actuators (Li et al., 2017; Felt et al., 2018; Tawk et al., 2018; Lee and Rodrigue, 2019). As a result of these design features, these actuators have potential benefit for a variety of applications, especially those requiring large linear displacement which is challenging to achieve with commonly reported artificial muscles.

Previous work has developed a variety of actuator models, often based on the Finite Element Method (FEM), for describing their actuator designs (Agarwal et al., 2017; du Pasquier et al., 2019; Nguyen and Zhang, 2020; Polygerinos et al., 2015b). These models allow in-depth characterization of corresponding actuators, predicting buckling modes, stress distributions and and actuator motion as a function of pressure, expected force output, and cycle lifetime. FEM approaches have been shown to characterize actuators for their use in a particular application, and maintain the versatility of their design for other applications – for example Nguyen and Zhang (2020) characterize a family of modular cells using FEM that can be combined for curling, linear, and twisting motions as desired by the end user.

Though FEM models are successful in describing actuator performance in detail, their complexity means they are not ideal for higher-level design iterations and selection of broad design spaces for performance constraints. To our knowledge, a generalized, versatile model that can rapidly generate information on an actuator's output is missing in the existing body of work. Such a model could be used before the time-consuming prototyping, material testing, and FEM model creation that comes with a more developed design.

Inspired by previous literature, we developed one such model that makes use of the virtual work principle to extract an actuator's force-contraction output force based on its volume loss rate. This is implemented via a simple and versatile numerical algorithm using solely geometrical features of the actuator. The model can then overlay other components of the actuator–such as a restoring force–to better approximate its force output.

The force-contraction profile (FCP) is a common characterization metric to describe the actuator output force over the course of its contraction assuming a constant pressure, and it is nonlinear for most vacuum actuators, creating demand for tools that can allow one to understand and predict the FCP for a given actuator design. There have been a variety of simplified models that attempt to predict such profiles. While some models make use of force balancing analytical and numerical finite element models (Polygerinos et al., 2015b; Li et al., 2017), others have modelled actuator outputs using analytical solutions to the principle of virtual work (Chou and Hannaford, 1996; Li et al., 2017; Felt et al., 2018; Lee and Rodrigue, 2019), which allows a force profile to be estimated solely from the actuator's geometry:

$$F = P * \frac{\mathrm{d}V}{\mathrm{d}s} \tag{1}$$

Where V is the actuator's internal volume, P is the actuation pressure (usually assumed constant), and s is the contraction or current length of the actuator. Assuming no energy loss and an inextensible skin, the output of the virtual work equation (**Eq. 1**) can be used to estimate the force output of the actuator directly (Chou and Hannaford, 1996; Li et al., 2017). The models based on the principle of virtual work mentioned above apply an analytical solution derived from the design and geometry of the actuator in question, following a typical workflow: a skin geometry is defined, used to derive a formula for volume as a function of contraction, and the volume formula is differentiated. In one instance, this analytical approach was combined with a minimizing function (Felt et al., 2018) to allow for the skin to change in cross-sectional geometry to mimic the physical tendency to minimize volume in a vacuum.

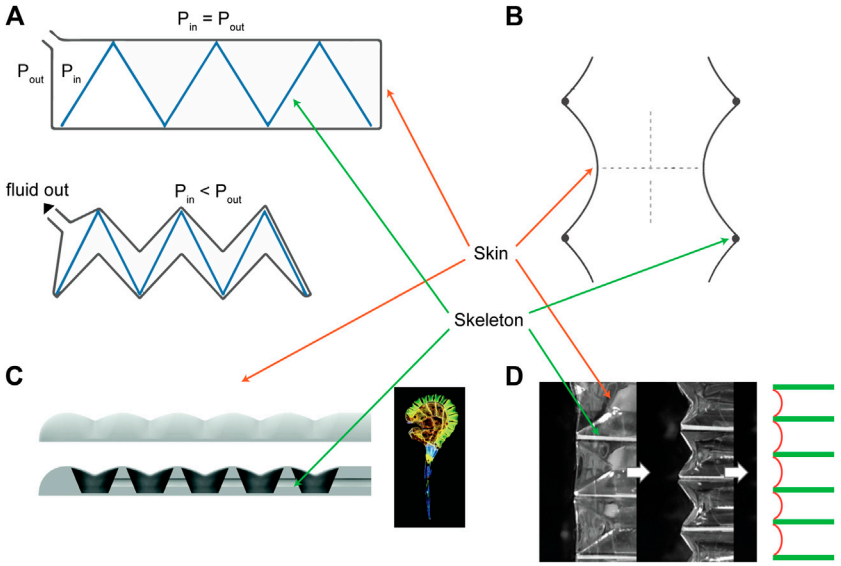


FIGURE 1 | Overview of existing skin-skeleton vacuum actuator designs, highlighting the skin and skeleton components. These include **(A)** FOAMs (Li et al., 2017), **(B)** Bellows actuators (Felt et al., 2018), **(C)** bending soft actuators (Tawk et al., 2018), and **(D)** Origami bellows actuator (Lee and Rodrigue, 2019). All figure reproductions approved by publishers.

In this paper, we expand the use of the virtual work concept and present a generalized platform that enables rapid prediction of the FCP of a linearly contracting skin-skeleton vacuum actuator for any skin or skeleton geometry. By implementing a generalized numerical approach in MATLAB (MathWorks), we create a versatile model that can easily be applied to different actuator designs, without the need for the development of a separate analytical model for each design. To demonstrate the application and capabilities of the framework, we use it to model the FCPs of two representative types of linear skin-skeleton actuators: the bellows actuators (Felt et al., 2018; **Figure 1A**), and the Fluid-driven Origami-inspired Artificial Muscles (FOAMs) (Li et al., 2017; **Figure 1B**). The bellows actuator was chosen for its simple design and pre-existing modelling work, and the FOAM was chosen because its semi-rigid zigzag shaped skeleton adds geometrical complexity and behaves like a spring, adding an additional component to test the framework's modularity. Finally, we validate the framework by experimentally characterizing actuators with varying geometric parameters. This framework has potential utility as a design tool for soft roboticists, or device designers, enhancing the computational efficiency of the virtual work principle with modularity, allowing rapid application to various actuator designs and geometries.

MATERIALS AND METHODS

Conceptual Framework

Guided by the virtual work approach described in previous studies (Chou and Hannaford, 1996; Felt et al., 2018), we derived a simple generalized formula for the FCP of a vacuum actuator based purely on its geometry. As **Figure 1** illustrates, most soft vacuum actuators with linear contraction motion exhibit volume loss in both the axial (ΔV_a) and radial (ΔV_r) directions. It is beneficial to categorize the volume loss in this way because, while the spatial derivative of the axial volume loss ($\mathrm{d}V_a/\mathrm{d}s$) leads to a constant force profile, the derivative of the radial volume loss is responsible for the nonlinearity in the actuator FCP, as will be shown. This also allows for an easy non-dimensional transformation of the output, as can be seen in the brief derivation below, based on the labeled V_r and V_a values from **Figure 2**:

$$\begin{aligned} V_{T,\mathrm{act}} &= A_c s - V_r; \\ V_{T,\mathrm{piston}} &= A_c s, \\ \frac{F_{act}}{P} &= \left(A_c - \frac{\mathrm{d}V_r}{\mathrm{d}s}\right); \\ \frac{F_{\mathrm{piston}}}{P} &= A_c, \\ \frac{F_{\mathrm{act}}}{F_{\mathrm{piston}}} &= \boldsymbol{F}^{*}_{\mathrm{act}} = 1 - \frac{1}{\boldsymbol{D}}\left(\frac{\mathbf{d}\boldsymbol{V_r}}{\mathbf{d}\boldsymbol{s}}\right), \end{aligned} \tag{2}$$

where V_T is the total internal volume of the actuator's contractile cell, F is its output force in Newtons, P is the constant actuation pressure and D is a characteristic length that replaces the cross-sectional area A_c in the case of a 2D simplification of the actuator. The subscripts *act* and *piston* refer respectively to the actuator in question and a piston of equivalent cross-sectional area, where a piston is defined as having no radial volume loss (**Figure 2**). F^{*}_{act} is the piston-scaled force of the actuator, a non-dimensional force or a ratio of the actuator's force output compared to that of its equivalent piston. In this case, the actuator is simplified as a two-dimensional equivalent, so A_c becomes a characteristic radial

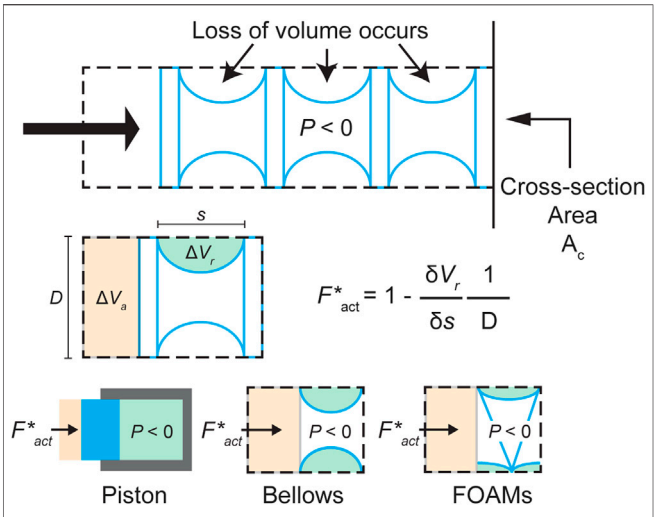


FIGURE 2 | A schematic of the general skin-skeleton vacuum actuator working principle, conceptualizing the components for the model derivation, with relevant volume losses labeled. Bottom row shows how **Eq. 2** can be applied to different actuator designs. ΔVr = volume loss in radial direction. ΔV_a = volume loss in axial direction. Ac = characteristic radial length. F^*_{act} = scaled actuator output force. P = actuation pressure. s = current actuator length. D = cross-sectional area.

length, D, while V_r becomes a two-dimensional slice of the lost radial volume. As **Eq. 2** shows, the scaled output force of the actuator is a function of the derivative of the radial volume loss over its contraction s. This assumes a constant pressure and constant bounding cross-sectional area (D). By setting the characteristic length D to 1, **Eq. 2** describes a generalized scale-independent force profile curve.

As **Figure 2** illustrates, this concept of radial volume can be translated into different actuator types. The goal of our framework is to allow for a generalized application of the radial volume rate concept, such that the force profile for different actuators can be extrapolated.

Implementation into Model

To generalize the concept of radial volume loss, we identify two components that can be used to represent a vacuum actuator in our model: a skin profile and a boundary profile. As **Figure 3A** demonstrates, the model requires that a vacuum actuator be discretized into contractile cells, similar to that used in the derivation. This cell is simplified as a two-dimensional shape with a zero-thickness skin, described by a function f_s, and a set of boundaries, described by the boundary function f_b.

In this paper, we focus on modelling the bellows (Felt et al., 2018) and FOAM (Li et al., 2017) actuators, because both fall into the skin-skeleton category, have linear motion, and have been well-described experimentally and analytically in the literature. The bellows actuator has a simple working principle due to the low number of components in its assembly and its minimal skeleton design, while the FOAM actuator is interesting for the resistance of the folded skeleton resistance to contraction, which imparts an opposing spring force to the actuator. In both cases, the characteristic skin function is a parabola of constant length fixed at both ends of the cell, chosen to emulate the profile of the actuator skin as it conforms to the underlying skeleton during contraction.

$$f_s(s,h)(x) = -h\left(1 - \frac{x^2}{(s/2)^2}\right); \quad -\frac{s}{2} \le x \le \frac{s}{2}, \tag{3}$$

Where s is the length of the contractile cell, h is the skin's "sag depth" into the actuator, and x is the axial coordinate of a point along the skin (**Figure 3A** shows these variables in a schematic). The algorithm can accept alternative skin functions, provided they include the input h for sag depth, as discussed in the model implementation section.

The boundary conditions were defined separately for the bellows and FOAM actuators, as shown in **Figure 3A**. Although there are no defining structural bounds for the bellows actuator skin, a boundary at the midline was defined since the axisymmetry of the actuator causes the skin to contact itself on actuation for cases where the gap distance between rings is greater than one diameter (Felt et al., 2018). For the FOAM, the boundary was defined by the zigzag shaped skeleton, which was assumed to have zero thickness. The boundary equations are as follows:

$$f_{b,\text{bellows}}(x) = -\frac{D}{2}, \tag{4}$$

$$f_{b,\text{FOAM}}(L,s)(x) = \frac{2d}{s}|x| - d; \quad d = \sqrt{l_i^2 - \left(\frac{s}{2}\right)^2}, \tag{5}$$

where L is the constant length of one of the sections of the zigzag of the FOAM's skeleton, defined as $D \div \cos(\theta_i/2)$, and d is the height of the skeleton for a given value of s. **Figure 3A** includes a schematic of the skin and boundary functions.

With the specified skin and boundary functions, the FCP can be solved, as illustrated in **Figure 3B**. The major computational section of the model is the process for calculating the geometrical configuration of the skin for each point in the contraction (between contractile cells, equal to the skin length, b, to 0). This is achieved via a binary search algorithm, which attempts to find the lowest value of h that leads to a skin configuration of length b (equal to the initial gap length of b), as the model assumes an inextensible skin. The algorithm requires a function $l_s(h)$, which outputs the length of the skin configuration for a given sag depth h. For a given increment i in the contraction, the binary search starts at $h_{i,1} = h_{i-1,\text{final}}$ (or 0 for the first increment), and adds a constant step value to h, updating the lower bound $h_{i,\text{lower}}$ until it reaches a value of h that returns a length greater than the desired target, at which point it sets the upper bound $h_{i,\text{upper}}$. The next increment, $h_{i,\text{next}}$ is linearly interpolated between $h_{i,\text{lower}}$ and $h_{i,\text{upper}}$ based on how far from the target section length the from the upper and lower bound configurations are:

$$h_{i,\text{next}} = \frac{\left(h_{i,\text{lower}} dist_{\text{upper}} + h_{i,\text{upper}} dist_{\text{lower}}\right)}{dist_{\text{upper}} + dist_{\text{lower}}}, \tag{6}$$

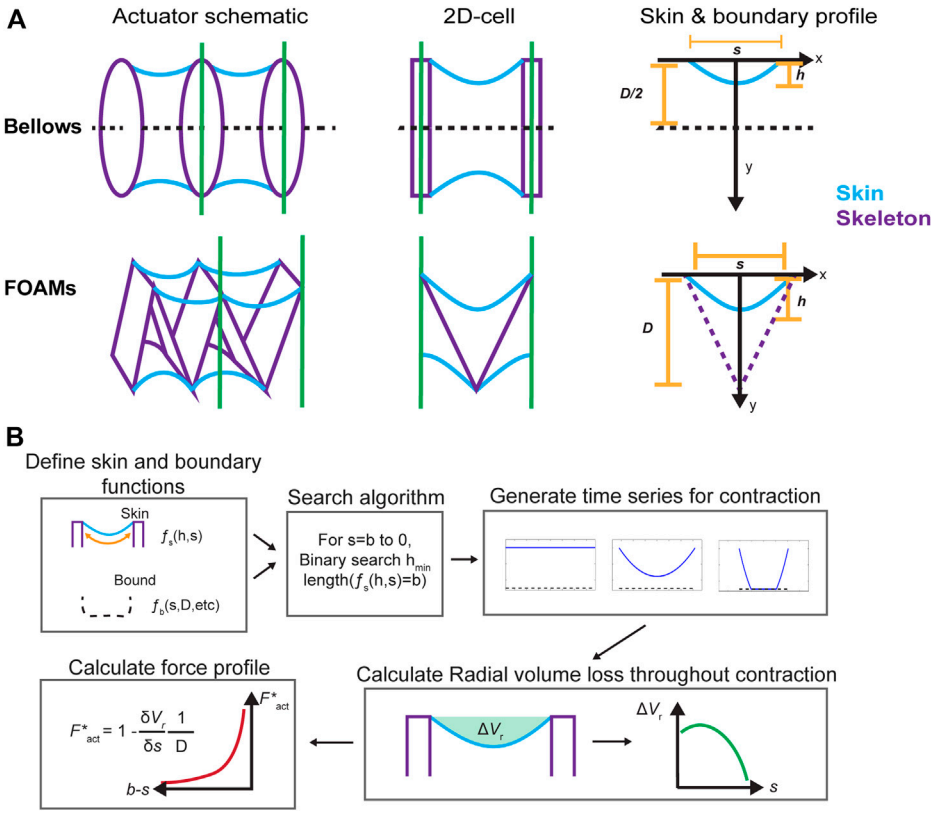


FIGURE 3 | (A) Simplifying actuator geometry for virtual work model. **(B)** A schematic of model implementation. s = length of a contractile cell. h = skin sagging depth. D = actuator cross-sectional area. b = initial spacing between two contractile cells or constant length of skin section. V_r = radial volume loss. F_s = skin function. f_b = boundary function. F^*_{act} = piston-scaled actuator output force.

where $dist_j = \left| l_s(h_{i,j}) - b \right|$. If $h_{i,\text{next}}$ is below the target, it becomes $h_{i,\text{lower}}$, and similarly for $h_{i,\text{upper}}$ in the case that the increment is above the target value. Eventually, this converts to the target value within a tolerance, but to return the smallest valid h value, the function returns only once the upper and lower bounds are close to each other by a certain tolerance. This assumes the section length is monotonically increasing as a function of h, but can be adjusted to account for local maxima/minima using traditional optimization function methods.

The final crucial piece of the algorithm is the skin profile function, which solves the geometrical configuration of the skin for a given h. Given a value for h, it first calculates the skin profile (in this case, always a parabola) and identifies any intercepts with the boundary function. If there is no intercept, it returns the skin profile as a 2D-array of x-y coordinates. If there is an intercept with the boundary, the function draws the section of the parabola up to the intercept, then draws the section of the boundary for the remainder of the way or until the second intercept. The process is repeated recursively to complete a profile.

The result of the binary search function and the skin profile function working in unison is a skin configuration for each contraction increment i, from s at full to zero length, referred to as the "time series" in **Figure 3B**. This represents the shape of the actuator's skin throughout the contraction. If, at some increment, the skin profile can no longer be solved, as is the case for the bellows actuators once the skin is in full contact with the boundary, that increment defines the end of the contraction. The V_r value for each increment is calculated and then numerically differentiated following the function for F^*_{act} to finally generate the FCP.

Finite Element Modeling of FOAM Actuators

In the case that an actuator has a non-negligible spring force that resists its contraction, we hypothesized that one can super-impose the calculated spring resistance force with the virtual work model FCP to reach an accurate estimate of the actuator's true FCP. To estimate the spring resistance force from skeleton, a quasi-static FEM model of a FOAM skeleton was created in Abaqus/Explicit (Dassault Systèmes). The skeleton was modeled with a 30-degree fold angle, 50 × 20 × 10 mm bounding dimensions, and 1 mm thickness and modelled as a linearly elastic polyvinyl chloride plastic (Density = 1.4 g/cc, Young's Modulus = 2.4 GPa, Poisson's Ratio = 0.3 as defined by the manufacturer specifications) and 8-node linear brick, reduced integration, hourglass control (C3D8R) elements. This skeleton was fixed at one end, restricting both rotation and displacement, and a displacement boundary condition of 35 mm was applied to the other end, compressing the skeleton gradually over time. The reaction force of the skeleton in the axial direction was extracted to quantify its

spring resistance. In parallel, a full FOAM model was generated, where the same skeleton was surrounded by a bounding skin. The skin was modelled using a thermoplastic elastomer with high stiffness (Density = 0.8 g/cc, Young's Modulus = 600 MPa, Poisson's ratio = 0.3, average properties from MatWeb) with 4-node, quadrilateral, stress/displacement shell elements (S4R) of thickness 0.02 mm, with a membrane idealization (such that the skin is dominated by tensile forces). A general, frictionless contact interaction was defined for all elements in the simulation. The skin was fixed at one end and a variable displacement condition was applied to another end (no boundary conditions were applied to the skeleton). Then, the model was run with the following actuation steps. First, a preload of vacuum pressure up to −70 kPa was applied inside the skin while the two ends of the actuator were held fixed. Once the pressure reached −70 kPa, the force was held constant and a gradual displacement boundary condition of 25 mm was applied to one end of the skin, while the other was kept fixed, allowing the actuator to contract. The axial reaction force at the fixed end of the skin was then extracted to quantify the FCP of the FOAM actuator. For the analysis, we compared the FCP generated by the FOAM actuator in the FEM and the net force predicted by the virtual work model, which was calculated as the pure FCP from the geometrical virtual work model subtracted by the skeleton spring force obtained in the FEM.

Actuator Fabrication

FOAM Actuators

FOAM actuators consist of a thin skin layer surrounding a rigid zigzag structure that serves as the skeleton (**Figure 4A**). For the skin layer, we sealed two sheets of 0.05 mm-thick thermoplastic elastomer (Fibreglast) using an impulse sealer for 20 s along two edges at a nominal spacing of 53 mm. For the skeleton, 0.254 mm-thick polyester sheet (McMaster-Carr) was laser cut in a series of 10 segments (L = 20 mm, W = 40 mm) with minor features on each segment to allow air flow and perforated lines between the segments to help folding. The skeleton was manually folded along the perforated lines at desired angles of 30, 60, and 90 degrees. For assembly, the skeleton was inserted into the skin membrane and sealed using an impulse sealer at skeleton lengths of 60, 110, and 150 mm, respectively. A piece of PTFE was used to create a gap in the seal for subsequent tube insertion. To ensure that the skeleton material did not slide inside the skin during actuation, we used thermoformable anchors at the ends of the skeleton that were sealed with the skin, and therefore fixed at each end.

Vacuum Bellows

A vacuum bellows actuator consists of a thin tubular membrane surrounding rigid rings that are evenly spaced along the axis of the tube (**Figure 4B**). Two 0.04 mm polyethylene sheets (McMaster-Carr, 7889T28) were sealed along two edges at a nominal width of 40 mm for 4 s using an impulse sealer (Hacona, H-6705) to make a 25 mm-diameter tubular membrane. For the rigid rings, a three-part assembly consisting of one concentric ring surrounded by two thinner annular rings placed at the edges of the inner ring was fabricated. The inner ring was made of 4.76 mm acrylic (McMaster-Carr) and laser cut to form an outer diameter of 40 mm with minor cut features that enable airflow between segments for middle segments and a 3.175 mm center hole for placing the tubing at one end. The outer annular rings were made from 1.59 mm acrylic with an inner diameter of 20 mm and outer diameter of 25 mm, and were bonded to the inner ring using cyanoacrylate (Loctite). The assembled rigid rings were positioned inside the membrane and orthogonally to the wall, and then secured around the groove created in the ring assembly using fishing line (9442T2, McMaster-Carr). The remainder of the ring assemblies were positioned along the membrane at the desired spacing and fastened in similar way. A 3.175 mm OD polyurethane tube was inserted through the first ring assembly for vacuum supply and the ends of the membrane were sealed to the acrylic using cyanoacrylate adhesive and SilPoxy (Smooth-On).

Testing

Actuator Testing

To obtain FCPs for each actuator, we measured the force-displacement curve using a mechanical tensile tester (Instron 5944) for all actuators. The actuators were held at the ends with a 2-kN load cell (**Supplementary Figure S1**) and allowed to contract at a rate of 100 mm/min until the force reached zero. Constant vacuum pressures of −15 kPa for the vacuum bellows and −25 kPa for the FOAM actuators were applied throughout the test using a manual vacuum gauge (IRV10-N07, SMC). Actuation pressure was measured using a TruWave pressure sensor (Edwards Lifesciences) and the average pressure for each actuator was used for normalizing the measured output forces. For both Bellows and FOAM actuators, three replicates (n = 3) were used for each experiment (i.e n = 3 for each value of R for Bellows, each value of θ for FOAMs).

Skin Material Testing

To compare the mechanical properties of different skin materials, a uniaxial tensile test was performed on an Instron 5944 at a rate of 1 mm/min. All rectangular test specimens had widths of 20 mm and lengths of 40 mm. The thickness of the skin materials was 0.04 mm for the polyethylene film and 0.05 mm for the thermoplastic elastomer. The corresponding Young's modulus was obtained by taking the best fit slope between strains of 0% to 5%, and the average value (n = 5) was used for scaling the forces in the model.

Spring Compression Testing

To characterize the spring force generated by the skeletons in FOAM actuators, we measured the force exerted by a 30-degree zigzag skeleton during compressive loading at a rate of 100 mm/min on a mechanical tensile tester (Instron 5944). The spring constant was derived by taking the slope of force-displacement graph and the mean value (n = 3) was used to model the spring force for the FOAM actuators.

RESULTS

Bellows

Force Profile Explanation

Figure 5 illustrates the shape of a force profile for a bellows actuator, overlaid with a simplified time series of the skin profiles

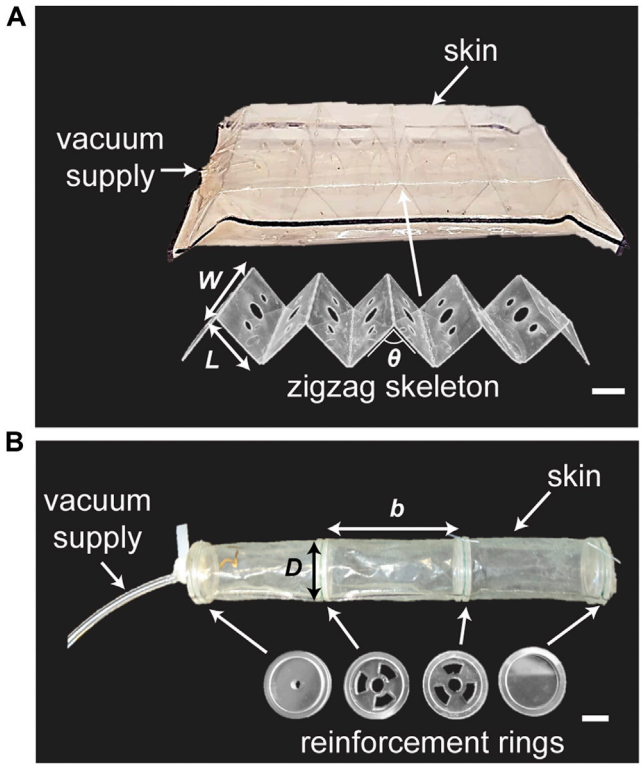


FIGURE 4 | Actuator prototypes. **(A)** A FOAM actuator contains a linear zigzag skeleton with multiple segments (W x L) separated by θ inside a bag of thin skin. W = width of skeleton, L = length of a skeleton section, θ = skeleton initial angle. **(B)** A vacuum bellows actuator contains multiple reinforcement rings equally spaced by R inside a thin tubular skin. R = ring spacing. D = ring diameter. Scale bar = 1°cm.

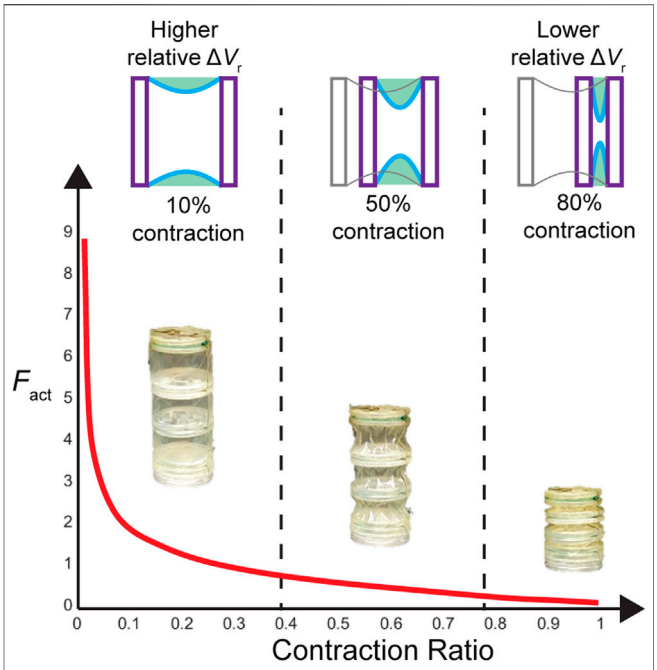


FIGURE 5 | Illustration of the force-contraction profile of bellows actuators, depicting the geometrical reasoning around the shape of the profile based on the concept of the virtual work function. Initially, the actuator skin is completely straight, and as it contracts, the parabolic profile leads to a rapid loss in radial volume, which justifies the large initial force value (large derivative). As the contraction nears its middle and end, the parabolic skin profile sags less and becomes narrower, leading to a smaller rate of volume loss and thus a smaller piston-scaled force.

generated by the model. As shown, the large force output at the start of the contraction is directly tied to the large relative change in radial volume at the beginning, as the skin experiences the greatest drop. In later stages, the output force curve flattens as the skin gets closer to its final configuration, and the loss in radial volume decreases.

Parameter Sweep

We performed a parametric sweep of different ring gap-diameter ratios using our virtual work model to evaluate the effect of the ring spacing on the force profile of bellows actuators. **Figure 6A** shows the calculated force profiles for varying ring gap-diameter ratios, $R = b/D$. The peak force decreases with decreasing gap distance, the force profile becomes more linear with decreasing gap distance, and gap distances greater than one diameter in length lead to a maximum scaled contraction equal to R^{-1}. **Figure 6B** overlays the full parameter sweep of values of R from 0 to 4, demonstrating how much stroke and force can be generated by an ideal vacuum bellows actuator for the given ring and gap dimensions.

Experimental Validation

To validate the model experimentally, we measured the force-contraction behavior of bellows actuators with varying gap ratios, R, of 0.5, 1, and 2. **Figure 7** shows the piston-scaled force produced by the actuator over the scaled contraction (defined as displacement divided by the actuator length). As a comparison, the virtual work model FCP with the equivalent R is overlaid with the results, after a magnitude scaling factor is applied. This factor accounts for the material properties of the actuator skin, which has some extensibility compared to the inextensible skin assumption in the model. We derived a relationship for the scaling factor through a set of FEM experiments extracting the FCP of a bellows actuator with varying skin thicknesses and stiffnesses, from which we could derive an empirical relationship between output force and Young's modulus (E) and skin thickness (t) (See **Supplementary Figure S3**). We found the scaling factor is a function of $(Et)^{1/3}$, which agrees with Roark's formula for the tension in a cable with a distributed load (Young and Budynas, 2002)–see **Supplementary Section 1.1** for further details. Li et al., (2017) observed the relevance of skin Young's modulus and thickness for the output of vacuum actuators, which informed our decision to develop this scaling factor. For our bellows actuators, which use a 0.04 mm thick polyethylene film as the skin material (E = 127 MPa), we scaled the output force

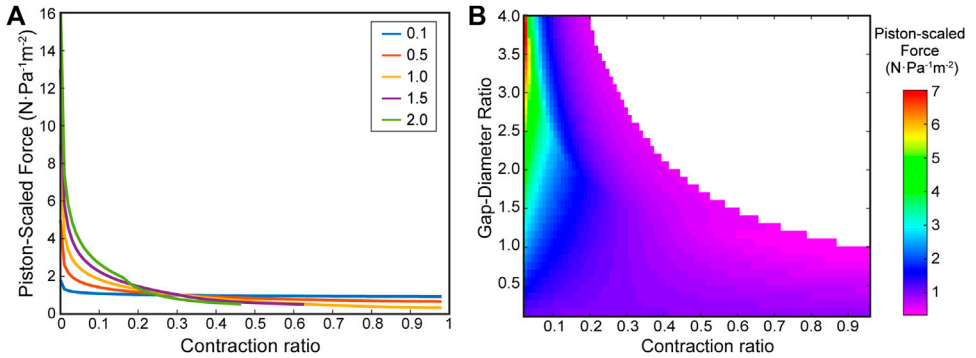


FIGURE 6 | Virtual work model for vacuum bellows actuator with varying ring distances. **(A)** Force-contraction profile for varying gap-diameter ratios (R) = 0.1, 0.5, 1, 1.5, 2, showing the trends caused by varying ring gap. **(B)** A heatmap from a high-resolution parameter sweep of the gap-diameter ratio. The color bar indicates piston-scaled force predicted by the model, and the contraction ratio is cut off at 0.04 for the purposes of visualizing the contrast throughout the heatmap (large magnitudes past 0.04 lead to colors focused on high values).

predicted by our model by 0.35, calculated through our derived scaling factor equation (see **Supplementary Eq. S1**).

FOAMs

Next, we expanded our virtual work model framework to the FOAMs actuators developed by Li et al., (2017). These actuators not only require a different boundary geometry, but the addition of a spring force from the folded rigid skeleton.

Parameter Sweep

First, we predict the FCP of the FOAMs geometry without an additional spring factor, only considering the effect of its triangular bounding function shape. The skeleton angle θ in FOAM actuators is analogous to the ring gap distance R in bellows actuators, as the fold-to-fold distance in the zigzag skeleton is a function of θ: $R = 2\tan(\theta/2)$. **Figure 8** demonstrates the FCPs predicted from the FOAMs boundary setup for varying θ in absence of spring force.

Finite Element Modeling of FOAM Actuators

Figure 9A shows deformation of skeleton during spring compression test performed in FEM, and **Figure 9B** shows contraction of a FOAM actuator using the same skeleton under constant pressure. **Figure 9C** compares the FCPs generated with the FEM (Fnet,FE) and virtual work models (Fnet,mod), demonstrating that the virtual work model can accurately describe a force profile when the skeleton spring reaction force is subtracted from it. The peak force from the virtual work model was within a 7% error and the full contraction length was within a 1% error.

Actuator Selection Example

To demonstrate the use of this model as a design guidance tool with its large parameter sweep capabilities, we developed an example for selecting the ideal actuator gap ratio (R) for a simplified exoskeleton or humanoid robot arm application using a bellows actuator. As shown in **Figure 11A**, the actuator is anchored across an elbow and must be able to counter a constant torque equivalent to 1 N applied at the end of the arm (F_{res}), throughout the bending of the arm $\left(\alpha : \left[\frac{\pi}{16}, \frac{5\pi}{6}\right]\right)$. The forearm anchor d_1 is close to the elbow and the shoulder anchor d_2 is further from the elbow to mimic the layout of a bicep. The biomechanical analysis with a d_1 of 5 cm, d_2 of 20 cm, and d_3 of 30 cm shows the required contraction is 36% and the peak force coincides with the zero-contraction point, which agrees with the intrinsic FCP of a bellows actuator. The optimized actuator for this case is one with maximum peak initial force and a contraction of at least 36%. Given that R is proportional to peak force and inversely proportional to contraction (**Figure 6B**), the optimized problem is solved by identifying the highest R value with a final contraction of at least 36%. A simple maximizing algorithm was applied with these conditions to the parameter sweep data, and an optimized R of 2.4 was found.

Finally, by choosing a scaling factor of 0.5 (to match the TPU above, a common polymer skin material) and actuator cross section of 5 × 5 cm (as a realistic dimension for this application), the actuator output curve was compared to the required output, as overlaid in **Figure 11B**, and a linear pressure-force relationship is used to derive the necessary actuation pressure throughout a quasi-static contraction. The same was repeated for an R of one to illustrate how the pressure control conditions change with actuator selection.

DISCUSSION

Bellows Actuator Model and Experimental Comparison

In our work, the bellows actuator was chosen as the first application as its simple design allows it to be realistically idealized by the assumptions made in the model, without the need for additional components. In **Figure 6**, we demonstrate that the model can predict FCPs with varying gaps as well as performing large parameter sweeps with high granularity. Our

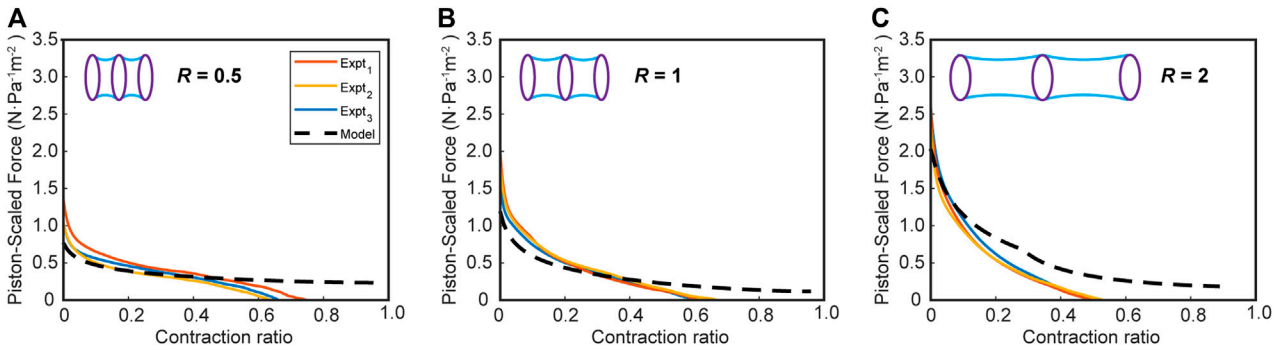


FIGURE 7 | The force-contraction profile for vacuum bellows actuators in the experimental and scaled virtual work model for varying ring gap-diameter ratios **(A)** R = 0.5, **(B)** R = 1, **(C)** R = 2 for a ring diameter of 25 mm. Number of experiments, n = 3.

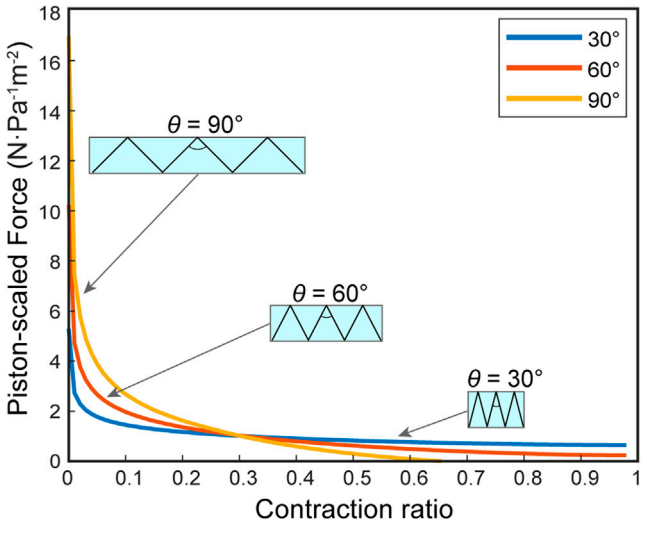


FIGURE 8 | The virtual work model prediction of force-contraction profile for FOAM actuators with skeleton angles (θ) = 30, 60, and 90 degrees. The profiles are based only on the geometrical features, without the inclusion of spring force.

findings are consistent with the major trends found in the analytical model from Felt et al., (2018). Namely by increasing *R*, the force magnitude on the initial part of the contraction is increased while the total stroke is decreased for an *R* above 1. Additionally, the smaller the value of R the more constant the FCP becomes. The heat map in **Figure 6B** also agrees closely with that in Felt et al., (2018) and can give users a holistic view of the design space and help inform the actuator selection process (see **Supplementary Figure S8** and Section *Actuator Selection Example*). As can be seen in **Supplementary Figure S8**, the models are in close agreement–the estimated contraction lengths, the trend of force distribution are similar, though the Felt model implemented a different more complex skin profile function which is a strong possible reason for the discrepancy in scaled force magnitudes.

Comparing the model and experimental results in **Figure 7**, we observe a close agreement in terms of FCP. During the early phase of the contraction, particularly for the R = 0.5 and one sets, there is very close agreement but we see some deviation of the modeling results from the experimental results toward the end of contraction in all cases for bellows actuators (**Figure 7**), most likely due to the zero-thickness skin and zero-energy loss assumption in the virtual work model. With these assumptions, the skin collapses in an orderly fashion until the cell contraction reaches 100%, when in reality the thickness and chaotic crumpling of the skin causes a nonlinear restoring force that increases toward the end of the contraction. This phenomenon of a nonlinear decreasing force is corroborated in the experimental results from Felt et al., (2018).

To demonstrate that this skin restoring force is a significant contributor to the discrepancy between the model and experimental results, we performed a second experiment where the actuators were compressed with an internal gauge pressure of zero and their restoring force was measured (see the **Supplementary Material** for experiment details). By overlaying this restoring force with the model for individual actuators, the output curve is estimated more closely (**Supplementary Figure S5**). The FCP curves downward at the end of the contraction, matching the experiment, and the final contraction is more closely approximated (with an average error of 14%). Given the complex mechanics of this crumpling skin restoring force, future work would be needed to create a predictable model based on this phenomenon. The effect of a restoring spring force is further investigated in our FOAM model, where the skeleton's restoring force is characterized by a linear trend.

FOAMs Model and Experimental Comparison

Comparing the FOAMs parameter sweep figure (**Figure 8**) with the results see in **Figure 6**, we can see the same general trend is preserved, where there are higher forces and lower maximum contraction with increasing θ. One key difference is demonstrated by the $\theta = 90^{\circ}$ case, where the maximum contraction is restricted

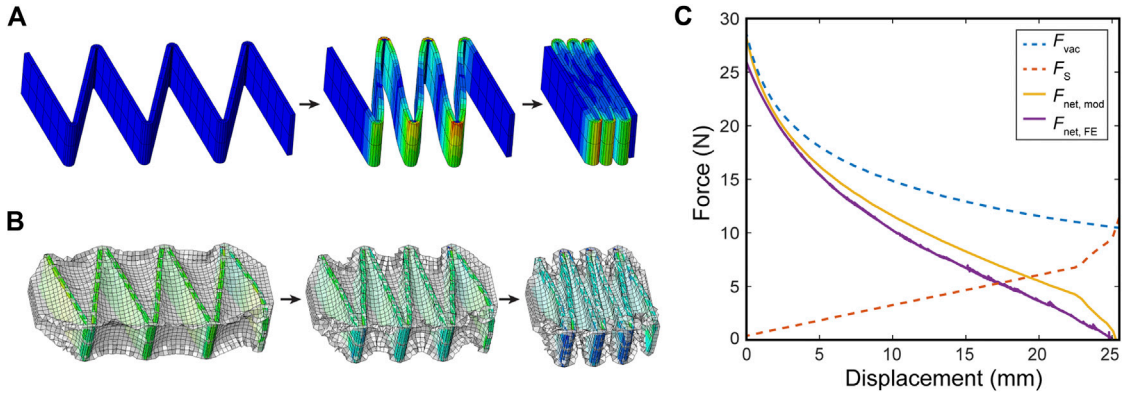


FIGURE 9 | (A) Deformation of a 30-deg skeleton during compression test in a Finite Element (FE) model. **(B)** Free contraction of FOAM actuator containing the same skeleton under constant negative pressure in a FE setup. **(C)** A comparison of FE Model and virtual work model for FOAM actuators for $\theta = 30$, $P = -70$ kPa. Fs = Skeleton spring force predicted by FE. F_{vac} = Force predicted by virtual work model without spring force. $F_{net,mod} = F_{vac} - F_s$ = Net force predicted by the MATLAB model. $F_{net,FE}$ = FOAM force predicted by FE.

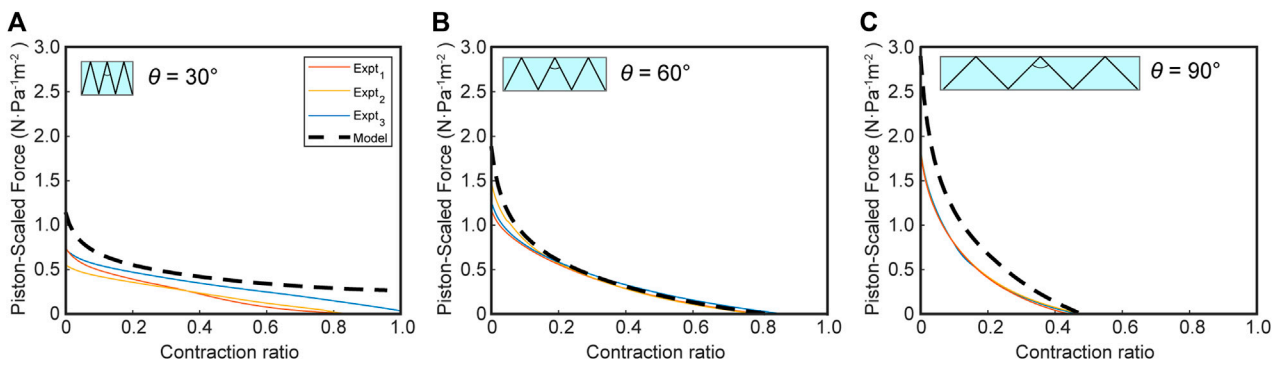


FIGURE 10 | Force-contraction profile for FOAM actuators in experimental and virtual work model for varying skeleton angles [θ]; **(A)** 30°, **(B)** 60°, **(C)** 90°. Model = predicted net force = (pressure-only force without skeleton)-(skeleton spring force). Number of experiments, n = 3.

when the scaled force magnitude reaches zero, meaning the radial volume loss derivative reaches a maximum, unlike for the bellows actuator where maximum contraction is bounded by a geometrical constraint when the skin comes into maximum contact with the boundary. This demonstrates the importance of defining specific boundary functions in the virtual work model, based on actuator design.

As hypothesized, **Figure 9** illustrates the importance of including the skeleton spring force in the model for predicting the FOAM force profile. Though the initial FCP from the pure virtual work model matches that of the FEM, once the spring force becomes non-negligible, the virtual work model deviates and predicts much higher force and contraction at the end of the FCP. The results also show that the spring force can be subtracted linearly as a post-processing step, rather than having to be integrated into the virtual work model framework itself.

In keeping with this demonstration, the experimental results in **Figure 10** show that the virtual work model with the subtracted skeleton spring force can closely predict the FOAM actuator force profile. The results of excluding a spring force in the model are two-fold; an over-estimated contraction distance and a larger force profile toward the end section, as is visualized in **Figure 9**. In the case of $\theta = 30^{\circ}$, the contraction is smaller compared to our prediction. This indicates that the spring resistance force from the skeleton may contribute less when the absolute displacement of the skeleton is smaller (because the 30-degree skeleton is comparatively shorter than the others). In this case, similar to the bellows actuator, the loss may be dominated by resistance from the skin, which is not captured in this model.

Actuator Selection Example

The example in **Figure 11** demonstrates how the proposed model allows users to rapidly identify ideal actuator parameters (R) given requirements of their problem, and how they might use the calculated FCP to extract more questions about implementation.

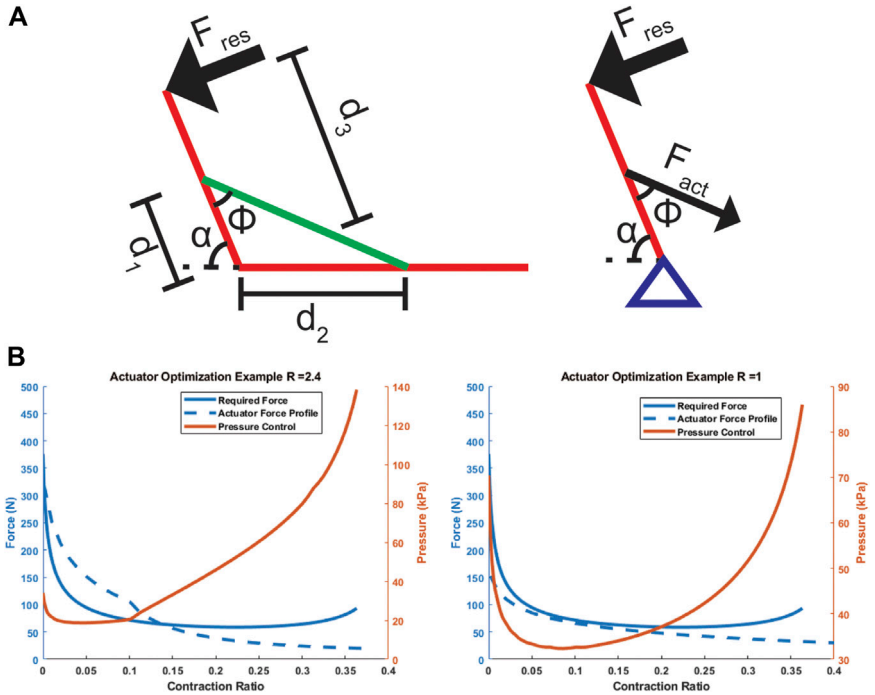


FIGURE 11 | Schematic **(A)** and results **(B)** of actuator design selection example. **(A)** depicts a simplified arm setup with the actuator in green acting as a contractile muscle. **(B)** Depicts the desired force curve from a quasi-static analysis of the system, a force output curve for the chosen actuator, and a pressure control curve to match the actuator output to the desired force.

The second analysis with an actuator of R = 1 in **Figure 11B** illustrates this point, as that actuator requires a higher initial pressure and lower final pressure compared to the large pressure ranges required for the R = 2.4 case. However, the R = 1 actuator is not restricted to 36% contraction, and so would require a more complex sensing and control scheme to remain within the parameters of the problem. This demonstrates that the model's calculated FCP can help in the consideration of other design parameters such as available pressure ranges, pressure control resolution, and required control scheme.

Generalizing Model to Other Actuator Designs

As initially explained, this model assumes a thin inextensible film being pulled toward a skeleton or boundary with a vacuum. It is ideally applied to actuators such as the bellows and FOAMs as shown above, with additional components to account for the material of the skin and the system's restoring force. From these two examples, we have demonstrated the model's ability to be applied to different actuator geometries and its modularity in combining with external models. With these conclusions in mind, the model can be applied to other more mechanically complex actuator designs, and we list some examples in this section.

The vacuum powered curling actuator in Tawk et al., 2018 has the necessary thin skin and skeleton components, though it does not undergo linear contraction. **Supplementary Figure S7** shows how the skin and boundary functions can be applied, with the boundary function changing during contraction to accommodate for the contractile cell's bending. The output linear force can be converted to a torque by assuming a point load at the half-way height or other more complex force distributions along the height. A cantilever-based torsional spring model would also be overlaid to account for the restoring force of the bending skeleton.

Another possible application would be the buckling actuators in Yang et al., 2016; **Supplementary Figure S7**). The key issue in this case is the skin is a proportionally thick elastomeric layer, which may still allow for the inextensible assumption, but not that of a thin-film. A virtual thickness must be included in the model to account for the limitation on the stroke length, and a more complex spring model based on buckling would be included to model the skin's restoring force. In that case, parameters such as skin thickness and material would be interesting to modulate, in addition to the usual contractile cell length and aspect ratio.

A process such as this–defining the skin and skeleton profile functions, reassessing the base model's assumptions, and defining additional restoring force or loss components to overlay into model–can be repeated for other actuators that fall under the vacuum powered skin-skeleton category of actuators.

As a final note on further generalizability-the power of the virtual work principle is the computational simplicity in

reaching an accurate estimate of a vacuum actuator's force-contraction profile. This implies a possible approach to actuator design for more complex FCPs by controlling the volume loss rate. Actuators with origami skin patterns (Martinez et al., 2012; Li et al., 2017; Lee and Rodrigue, 2019), for example, have high potential in this application, as a new model could be developed and use the geometrical predictability of origami to generate new actuator designs to fit desired force-profiles.

Limitations

In addition to illustrating the concept of virtual work, the curve in **Figure 5** also illustrates one of the limitations of the model: the large initial force. This large force is due to the assumption that the skin at zero contraction is in full tension. In any practical implementation of an actuator, this does not hold for multiple reasons. First, there is often a small tolerance between the skin and the actuator skeleton, or in the case of the bellows actuator where there is no gap, asymmetries in the construction can lead to sections of the skin that are not in perfect tension. Perhaps more importantly, this assumption conflicts with the quasi-static nature of the system, as the pressure applies a radially-oriented (perpendicular) force on the skin that can only be compensated for by the tension on the skin (Li et al., 2017), requiring some initial curvature to allow for a radial component of the tension force. The effect of this physical inaccuracy is that, when comparing with experimental data, it is important to shift the model horizontally toward the negative x direction by some small percentage to eliminate the high peak force. After implementing a scaling factor that considers the above logic about a required initial skin curvature (**Supplementary Section 1.1**), the initial peak force was still higher than the experimental case, most probably due to the other flaws of fabrication that lead to an initial curvature in the skin. Given these unquantifiable factors, a shifting factor of 2% was implemented. This means in all comparisons of the model FCP with experimental data, the model FCP is shifted horizontally so its zero-point begins at a contraction of 2%.

In **Figure 7**, for the bellows with an R of 2, the model overpredicts the output force most probably due to this tension phenomenon. The large unsupported length of skin may have experienced higher cumulative pressure force and larger tension force compared to R = 0.5, 1, (as supported by the higher peak force for R = 2 in **Figure 6A**). This means the initial sagging of the skin for R = 2 may be greater than for 0.5 and 1, calling for a larger shift. Shifting the model by 2% for R = 2 eliminates the larger error in the estimated initial force and better approximates the early phase of the force profile.

Finally, in **Figure 8**, there is a consistent over-estimation of the initial force. We hypothesize that this is a result of the tension phenomenon combined with the fact that the skin in FOAMs, unlike for bellows actuators, is not directly anchored to the skeleton, implying that the sections of skin between the skeleton are slightly longer than the ideal perfect-tension model predicts. Due to the overall similarity in the profile shape, there is a better agreement with experimental results if the model is further shifted horizontally in this case as well.

CONCLUSION

In this work, we hypothesized that the nonlinear force-contraction profile of a skin-skeleton vacuum actuator can be derived purely from a geometrical calculation of its volume loss rate. We tested this with the model and experiments presented here and proved that our hypothesis was valid. Though the magnitudes of the resultant forces are dependent on external properties, such as skin material, skeleton and skin restoring forces, and actuation pressure, we show that the shape of the FCP is dominated by the work done through volume loss rate. By applying the piston-scaled force and scaling factor and overlaying external models to capture the system's restoring force, we can closely estimate the output of different actuators with much less computational and set-up time than developing actuator-specific FEM or analytical models. Importantly, these external models can be separately super-imposed on the FCP calculated by the virtual work model, preserving its modularity. Inaccuracies in this model are compensated for by its generalizability and utility as a design guiding tool to allow rapid parameter space sweeps. We demonstrate the latter point through a simple optimization example for a hypothetical application.

Future work would include developing a more robust system of modules that can model the restoring forces for various skin and skeleton geometries to interface with the virtual work model, as this is currently one of the largest sources of discrepancy between the model and experimental data. More work can also be done on understanding the effect of the skin material on the output force magnitude.

Notwithstanding these challenges, the generalized and modular nature of this framework enables its implementation as a design tool for a wide variety of vacuum actuators, provided they can be represented by one or more simplified skin-skeleton contraction cells. This ability to rapidly model a variety of actuators and actuator geometrical parameters has broad implications in improving design efficiency and speed in the fields of medical devices, robotics, and soft machines. For example, it can enable actuator design for application-specific force profiles, such as in patient-specific devices or robotic design, and the rapid computation time can be useful for dynamic closed-loop control applications in soft robotics.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

SG, CP, BK and ETR designed the experiments. SG developed the model. SG, CP and BK conducted the

experiments and analyzed the data. SG, CP, BK and ETR wrote the manuscript.

FUNDING

ETR acknowledges funding from MIT Department of Mechanical Engineering, the Institute for Medical Engineering and Science at the Massachusetts Institute of Technology, the National Science Foundation (CAREER Award 1847541) and the Muscular Dystrophy Association Research Grant 577961, and an MIT MISTI award. SG is funded by a Graduate Research Fellowship Program from the National Science Foundation, (Award number 1745302) and CP is a recipient of MathWorks Engineering Fellowship Fund from MathWorks, Inc. BK is funded by a Teaching Assistant Fund from the MIT Department of Mechanical Engineering.

ACKNOWLEDGMENTS

We acknowledge our funding sources.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2021.606938/full#supplementary-material.

REFERENCES

Agarwal, G., Robertson, M. A., Sonar, H., and Paik, J. (2017). Design and computational modeling of a modular, compliant robotic assembly for human lumbar unit and spinal cord assistance. *Sci. Rep.* 7 (1), 14391. doi:10.1038/s41598-017-14220-3

Andrikopoulos, G., Nikolakopoulos, G., and Manesis, S. (2011). "A survey on applications of pneumatic artificial muscles," in 2011 19th mediterranean conference on control and automation, MED, 2011, Corfu, Greece, June 20–23, 2011 (Piscataway, NJ: IEEE), 1439–1446. doi:10.1109/MED.2011.5982983

Calisti, M., Giorelli, M., Levy, G., Mazzolai, B., Hochner, B., Laschi, C., et al. (2011). An octopus-bioinspired solution to movement and manipulation for soft robots. *Bioinspir. Biomim.* 6 (3), 036002. doi:10.1088/1748-3182/6/3/036002

Chou, C. P., and Hannaford, B. (1996). Measurement and modeling of McKibben pneumatic artificial muscles. *IEEE Trans. Robotics Automation* 12 (1), 90–102. doi:10.1109/70.481753

Connolly, F., Polygerinos, P., Walsh, C. J., and Bertoldi, K. (2015). Mechanical programming of soft actuators by varying fiber angle. *Soft Robotics* 2 (1), 26–32. doi:10.1089/soro.2015.0001

Deimel, R., and Brock, O. (2016). A novel type of compliant and underactuated robotic hand for dexterous grasping. *Int. J. Robotics Res.* 35 (1–3), 161–185. doi:10.1177/0278364915592961

du Pasquier, C., Chen, T., Tibbits, S., and Shea, K. (2019). Design and computational modeling of a 3D printed pneumatic toolkit for soft robotics. *Soft Robot* 6 (5), 657–663. doi:10.1089/soro.2018.0095

Felt, W., Robertson, M. A., and Paik, J. (2018). "Modeling vacuum bellows soft pneumatic actuators with optimal mechanical performance," in 2018 IEEE international conference on soft robotics, RoboSoft 2018, Livorno, Italy, April 24–28, 2018 (Piscataway, NJ: Institute of Electrical and Electronics Engineers Inc), 534–540. doi:10.1109/ROBOSOFT.2018.8405381

Jiao, Z., Ji, C., Zou, J., Yang, H., and Pan, M. (2019). Vacuum-powered soft pneumatic twisting actuators to empower new capabilities for soft robots. *Adv. Mater. Tech.* 4 (1), 1800429. doi:10.1002/admt.201800429

Katzschmann, R. K., Marchese, A. D., and Rus, D. (2016). Hydraulic autonomous soft robotic fish for 3D swimming. *Springer Tract. Adv. Robot.* 109, 405–420. doi:10.1007/978-3-319-23778-7_27

Kim, S., Hawkes, E., Cho, K., Jolda, M., Foley, J., and Wood, R. (2009). "Micro artificial muscle fiber using NiTi spring for soft robotics," in In 2009 IEEE/RSJ international conference on intelligent robots and systems, St. Louis, MO, October 10–15, 2009 (Piscataway, NJ: IEEE), 2228–2234. doi:10.1109/IROS.2009.5354178

Lee, J. G., and Rodrigue, H. (2019). Origami-based vacuum pneumatic artificial muscles with large contraction ratios. *Soft Robot* 6 (1), 109–117. doi:10.1089/soro.2018.0063

Li, S., Vogt, D. M., Rus, D., and Wood, R. J. (2017). Fluid-driven origami-inspired artificial muscles. *PNAS* 114 (50), 13132–13137. doi:10.1073/pnas.1713450114

Li, S., Batra, R., Brown, D., Chang, H. D., Ranganathan, N., Hoberman, C., et al. (2019). Particle robotics based on statistical mechanics of loosely coupled components. *Nature* 567, 361–365. doi:10.1038/s41586-019-1022-9

Mac Murray, B. C., Futran, C. C., Lee, J., O'Brien, K. W., Amiri Moghadam, A. A., Mosadegh, B., et al. (2018). Compliant buckled foam actuators and application in patient-specific direct cardiac compression. *Soft Robot* 5 (1), 99–108. doi:10.1089/soro.2017.0018

Martinez, R. V., Fish, C. R., Chen, X., and Whitesides, G. M. (2012). Elastomeric origami: programmable paper-elastomer composites as pneumatic actuators. *Adv. Funct. Mater.* 22 (7), 1376–1384. doi:10.1002/adfm.201102978

Miriyev, A., Stack, K., and Lipson, H. (2017). Soft material for soft actuators. *Nat. Commun.* 8 (1), 596–598. doi:10.1038/s41467-017-00685-3

Mosadegh, B., Polygerinos, P., Keplinger, C., Wennstedt, S., Shepherd, R. F., Gupta, U., et al. (2014). Pneumatic networks for soft robotics that actuate rapidly. *Adv. Funct. Mater.* 24 (15), 2163–2170. doi:10.1002/adfm.201303288

Nguyen, P. H., and Zhang, W. (2020). Design and computational modeling of fabric soft pneumatic actuators for wearable assistive devices. *Scientific Rep.* 10 (1), 66. doi:10.1038/s41598-020-65003-2

Niiyama, R., Sun, X., Sung, C., An, B., Rus, D., and Kim, S. (2015). Pouch motors: printable soft actuators integrated with computational design. *Soft Robotics* 2 (2), 59–70. doi:10.1089/soro.2014.0023

Ogura, K., Wakimoto, S., Suzumori, K., and Yasutaka, N. (2009). "Micro pneumatic curling actuator-nematode actuator-," in 2008 IEEE International Conference on Robotics and Biomimetics, Bangkok, Thailand, Feburary 22–25, 2009 (Piscataway, NJ: Institute of Electrical and Electronics Engineers (IEEE)), 462–467. doi:10.1109/robio.2009.4913047

Polygerinos, P., Galloway, K. C., Savage, E., Herman, M., O'Donnell, K., and Walsh, C. J. (2015a). "Soft robotic glove for hand rehabilitation and task specific training," in Proceedings-IEEE international conference on robotics and automation, Seattle, WA, May 26–30, 2015 (Piscataway, NJ: Institute of Electrical and Electronics Engineers Inc), 2913–2919. doi:10.1109/ICRA.2015.7139597

Polygerinos, P., Wang, Z., Overvelde, J. T. B., Galloway, K. C., Wood, R. J., Bertoldi, K., et al. (2015b). Modeling of soft fiber-reinforced bending actuators. *IEEE Trans. Robotics* 31 (3), 778–789. doi:10.1109/TRO.2015.2428504

Porter, A. P., Marchesini, B., Potryasilova, I., Rossetto, E., and Newman, D. J. (2020). "Soft exoskeleton knee prototype for advanced space suits and planetary exploration," in 2020 IEEE Aerospace Conference, Big Sky, MT, March 7–14, 2020 (Piscataway, NJ: Institute of Electrical and Electronics Engineers (IEEE), 1–13. doi:10.1109/aero47225.2020.9172373

Robertson, M. A., and Paik, J. (2018). "Low-inertia vacuum-powered soft pneumatic actuator coil characterization and design methodology," in In 2018 IEEE international conference on soft robotics, RoboSoft 2018, Livorno, Italy, April 24–28, 2018 (Piscataway, NJ: Institute of Electrical and Electronics Engineers Inc), 431–436. doi:10.1109/ROBOSOFT.2018.8405364

Roche, E. T., Horvath, M. A., Wamala, I., Alazmani, A., Song, S. E., Whyte, W., et al. (2017). Soft robotic sleeve supports heart function. *Sci. Transl. Med.* 9 (373), eaaf3925. doi:10.1126/scitranslmed.aaf3925

Rus, D., and Tolley, M. T. (2015). Design, fabrication and control of soft robots. *Nature* 521, 467. doi:10.1038/nature14543

Sanan, S., Lynn, P. S., and Griffith, S. T. (2014). Pneumatic torsional actuators for inflatable robots. *J. Mech. Robotics* 6 (3), 031003. doi:10.1115/1.4026629

Tawk, C., In Het Panhuis, M., Spinks, G. M., and Alici, G. (2018). Bioinspired 3d printable soft vacuum actuators for locomotion robots, grippers and artificial muscles. *Soft Robot* 5 (6), 685–694. doi:10.1089/soro.2018.0021

Wirekoh, J., and Park, Y. L. (2017). Design of flat pneumatic artificial muscles. *Smart Mater. Structures* 26 (3), 035009. doi:10.1088/1361-665X/aa5496

Yang, D., Verma, M. S., So, J.-H., Mosadegh, B., Keplinger, C., Lee, B., et al. (2016). Buckling pneumatic linear actuators inspired by muscle. *Adv. Mater. Tech.* 1 (3), 1600055. doi:10.1002/admt.201600055

Yang, D., Verma, M. S., Lossner, E., Stothers, D., and Whitesides, G. M. (2017). Negative-pressure soft linear actuator with a mechanical advantage. *Adv. Mater. Tech.* 2 (1), 1600164. doi:10.1002/admt.201600164

Young, W. C., and Budynas, R. G. (2002). *Roark's formulas for stress and strain*. 7th Edn.New York, NY:McGraw-Hill Companies.

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.

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 12 March 2021
doi: 10.3389/frobt.2021.557830

Modeling and Reconstruction of State Variables for Low-Level Control of Soft Pneumatic Actuators

Serhat Ibrahim, Jan Christoph Krause, Alexander Olbrich and Annika Raatz*

Institute of Assembly Technology, Leibniz Universität Hannover, Hannover, Germany

To further advance closed-loop control for soft robotics, suitable sensor and modeling strategies have to be investigated. Although there are many flexible and soft sensors available, the integration into the actuator and the use in a control loop is still challenging. Therefore, a state-space model for closed-loop low-level control of a fiber-reinforced actuator using pressure and orientation measurement is investigated. To do so, the integration of an inertial measurement unit and geometric modeling of actuator is presented. The piecewise constant curvature approach is used to describe the actuator's shape and deformation variables. For low-level control, the chamber's lengths are reconstructed from bending angles with a geometrical model and the identified material characteristics. For parameter identification and model validation, data from a camera tracking system is analyzed. Then, a closed-loop control of pressure and chambers' length of the actuator is investigated. It will be shown, that the reconstruction model is suitable for estimating the state variables of the actuator. In addition, the use of the inertial measurement unit will demonstrate a cost-effective and compact sensor for soft pneumatic actuators.

Keywords: soft roboitcs, sensorization of soft robots, modeling of soft robots, control of soft robots, nonlinear control, test bench design

OPEN ACCESS

Edited by:
Concepción A. Monje, Universidad Carlos III de Madrid, Spain

Reviewed by:
Kean C Aw, The University of Auckland, New Zealand
Jorge Muñoz, Universidad Carlos III de Madrid, Spain

****Correspondence:***
Serhat Ibrahim
ibrahim@match.uni-hannover.de

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: *30 April 2020*
Accepted: *28 January 2021*
Published: *12 March 2021*

Citation:
Ibrahim S, Krause JC, Olbrich A and Raatz A (2021) Modeling and Reconstruction of State Variables for Low-Level Control of Soft Pneumatic Actuators. Front. Robot. AI 8:557830. doi: 10.3389/frobt.2021.557830

1 INTRODUCTION

Soft robots, with flexible shape and infinite configuration possibilities, offer completely new capabilities compared to conventional industrial robots [Trivedi et al. (2008) and Marchese et al. (2014)]. Due to their compliance, soft robots adapt to their environment. This makes them suitable for grippers handling objects with undefined shapes. Since there is no risk of damage in the event of a collision, they are also suitable for human-robot collaboration. A decisive factor determining the movement of soft pneumatic actuators is their design. New actuator designs and mechanisms have been developed for this field of research [Runge and Raatz (2017), Galloway et al. (2013) and Garcia et al. (2020)]. The soft and flexible structures with mostly nonlinear material properties and hyperelasticity present a challenge for modeling, sensing and control. Especially the use of suitable sensors for state detection of the actuator needs to be researched. Due to the low force and high deformability of the actuator, conventional strain gauges cannot be used for this purpose. One option is the use of contact-free camera tracking systems Runge and Raatz (2017). The disadvantage, however, besides the high costs, is the use in confined spaces (high space requirement of the cameras) to avoid covering in cluttered scenes. For use in confined spaces, sensors, which are integrated into the actuator, are more suitable [Szelitzky et al. (2014)].

TABLE 1 | Overview of different measurement methods for the determination of actuator deformation compiled from the current literature.

References	Measurement	Uncertainty	Frequency	DoF
Al Jaber and Althoefer (2018)	Optical	—	—	2
Best et al. (2015)	IMU	—	—	3
Donno et al. (2008)	Optical	0.01°	1 kHz	1
Felt et al. (2016)	Inductive	2°	—	1
Gerboni et al. (2017)	Conductive	1.08°	40 Hz	1
Gibbs and Asada (2005)	Resistor	2.4°	2 Hz	1
Roduit et al. (1998)	Resistor	2°	—	2
Seel et al. (2014)	IMU	3.3°	60 Hz	3
Visentin and Fiorini (2018)	Impedance	—	—	2
Yuen et al. (2018)	Capacive	—	10 Hz	1

Table 1 shows different methods for measuring the bending of soft actuators with integrated sensors. Roduit et al. (1998) and Gibbs and Asada (2005) use resistance measurements to determine a bending angle. Roduit et al. (1998) use the difference in position of two parallel cables and Gibbs and Asada (2005) use conductive fibers. Felt et al. (2016) present an inductive measuring method. Wire ties are cast around the fins of a soft pneumatic actuator and its inductance is measured. Here, an inductance change of up to 19% is observed for bends up to 190°. Besides mapping quantified elongation into a geometric shape, another method is the estimation by covering change in pose with an inertial measuring unit. In this case, the pose of an object is observed based on acceleration and rotation rates as well as magnetometer data. Best et al. (2015) and Seel et al. (2014) use this method to measure the bending between rigid links. Seel et al. (2014) achieve an accuracy of 3° at a frequency of 60 Hz.

Gerboni et al. (2017) use a commercial flex bend sensor based on conductivity measurements for a soft pneumatic actuator with one degree of freedom (DoF). In the experiment with a closed-loop control, an accuracy of 1.08° is achieved at a clock rate of 40 Hz. Yuen et al. (2018) describe the manufacture of strain sensors, which are directly integrated into several film layers in a soft pneumatic actuator. The capacitive based sensor consists of multiple layers with silicone-based conductive electrodes and silicone elastomers as the dielectric. For the measurement using electrical impedance tomography, shredded carbon fibers are arranged as electrodes in the actuator, as described in Visentin and Fiorini (2018). The change in electrical conductivity is measured to reconstruct the bending. The optical bending sensor presented by Donno et al. (2008) is a very accurate measuring method. Non-polarized laser light is polarized by a filter and sent through an optical fiber. If this optical fiber is bent, its polarization changes. Then the change in angle can be recorded *via* a photo electrode with a second polarizing filter. The accuracy for measurements with up to 1 kHz is specified as 0.01°. The use of alloys that are liquid at room temperature should also be mentioned here. EGaIn sensors can also be used for bending measurement [Mengüç et al. (2013)]. Their support fixtures are based on similar or same material as the actuator to avoid inflecting the behaviour of the actuator. However, the production of such sensors is proving to be difficult, and for this purpose separate system components must be developed.

In addition to the sensors, models or neural networks are also used to estimate the state parameters for closed-loop control [Runge and Raatz (2017), Tan et al. (2019) and Katzschmann et al. (2019)] from, for example, pressure measurements. Katzschmann et al. (2019) have published an approach for closed-loop control, where a reduced order finite element model is used for the feedback.

The research presented here aims to enable a low-level control for a three DoF fiber-reinforced actuator (FRA) using orientation measurement of the actuator's tip. For this purpose, an inertial measurement unit (IMU) is studied. The low-level system description is done at chamber level, where the chamber's pressure and length are considered. A reconstruction model is developed to observe the state variables, which are relevant for the control. In particular, the observed states include the lengths of the individual actuator chambers, which cannot be measured directly. To build the measurement model, an actuator segment is assumed to have a shape with a piece-wise constant curvature. The parameters are identified using particle swarm optimization and the validation of the measurement model is performed using a camera tracking system. The developed models are used for chamber length control and pressure control. Kinematic relationships between actuators chambers are not modeled. They are included in this concept as unknown disturbances. Compared to Katzschmann et al. (2019), we focus on the closed-loop control of individual segments at chamber level. For this purpose, we consider all components from the effector to the actuator chambers. The lumped second order dynamic model from Skorina et al. (2015) is on low level as well. In contrast to our work the effector system with a pneumatic valve is neglected for modeling.

For the test bench, a PC with Simulink Real-Time as operating system is used. It communicates with the Beckhoff IO-devices over EtherCAT bus. Three Enfield LS-V05 5/3 proportional directional valves are connected to regulate the airflow to the three FRA chambers. To reduce measurement noise, a peripheral EK1100 EtherCAT bus coupler with analog inputs connects five pressure sensors by First Sensors to measure pressure in all chambers, as well as supply and atmospheric pressure. To detect the orientation, the IMU is connected *via* a microcontroller with an EtherCAT shield. All components are commercially available.

2 MODELING OF THE SYSTEM

In the following, the system components are modeled for use in a closed-loop control (**Figure 1**). First, the behavior of the valves that regulates the airflow is described. The valve model is needed for the development of the sliding mode control (**Section 5.1**). Then the connection tubes between valves and actuator chambers are considered. Afterwards the actuator is modeled. For this purpose, the individual dynamic modeling of the chambers are

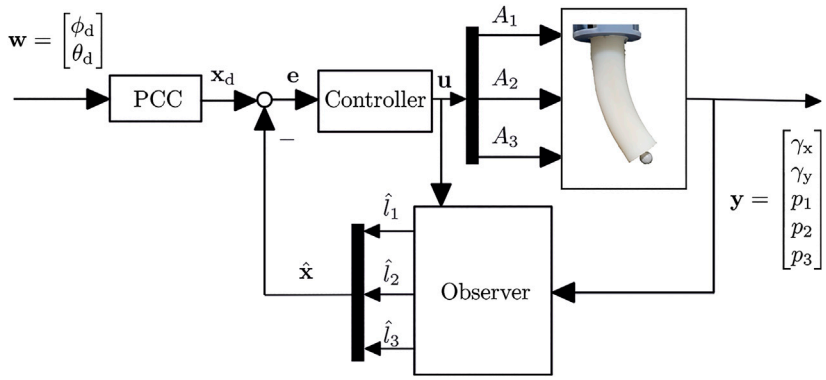


FIGURE 1 | The closed-loop control uses the difference between observed chamber length and the ones from PCC configuration as feedback.

combined to form a complete description of the entire actuator's geometry. Finally, a state-space representation of the soft robot system is set up.

2.1 Model of Valve

The valve model is based on the work of Ben-Dov and Salcudean (1995) and Richer and Hurmuzlu (2000a), Richer and Hurmuzlu (2000b). For a detailed description of the valve modeling, we refer to our preliminary work in Ibrahim et al. (2019). The air mass flow $\dot{m}_v$ through an orifice A of the valve is described with

$$\dot{m}_v = c_f A \frac{p_u}{\sqrt{T}} \Psi(p_d, p_u). \tag{1}$$

This mass flow depends on the upstream pressure p_u and downstream pressure p_d as well as the temperature T. Here, the temperature is assumed to be homogeneous throughout the system. The flow coefficient describes the ratio of real and ideal volume flow with

$$c_f = \frac{\dot{V}_{real}}{\dot{V}_{ideal}}. \tag{2}$$

The flow function can be calculated with

$$\Psi(p_u, p_d) = \begin{cases} \sqrt{\frac{\kappa}{R}\left(\frac{2}{\kappa+1}\right)^{\kappa+1/\kappa-1}} & \frac{p_d}{p_u} \le p_{crit} \\ \sqrt{\frac{2\kappa}{R(\kappa-1)}\left(\frac{p_d}{p_u}\right)^{1/\kappa}\sqrt{1-\left(\frac{p_d}{p_u}\right)^{\kappa-1/\kappa}}} & \frac{p_d}{p_u} > p_{crit} \end{cases}. \tag{3}$$

The flow Ψ depends on the critical pressure p_{crit}, which is calculated with

$$p_{crit} = \left(\frac{2}{\kappa+1}\right)^{\kappa/\kappa+1}, \tag{4}$$

with κ as the heat capacity ratio. Here, Ψ is constant for a pressure ratio p_d/p_u, which is smaller than the critical value. Differently, it is a nonlinear function, which depends on the upper- and downstream pressure.

The orifice A of the valve depends on the spool position x_s. Assuming that a rectangular slider with an edge length b covers a circular opening with a radius r, the effective area is calculated as a circle segment. With the coordinate

$$x_e = x_s - \frac{b}{2} + r, \tag{5}$$

the area A can be calculated with

$$A(x_e) = \begin{cases} 0 & x_e < 0 \\ (x_e - r)\sqrt{2rx_e - x_e^2} + r^2 \arccos\left(\frac{r - x_e}{r}\right) & 0 \le x_e \le 2r \\ \pi r^2 & x_e > 2r \end{cases}. \tag{6}$$

The dynamic of the spool displacement is described with the second order differential equation

$$m\ddot{x}_s = -F_f + F_S - 2kx_s - d\dot{x}_s. \tag{7}$$

F_f stands for the frictional force that occurs during the movement. The spool of the valve has a damping d and a stiffness $2k$. Its force is calculated with

$$F_S = K_S i_S = \frac{K}{\tau} u, \tag{8}$$

with the spool current i_S and the motor constant K_S, as well as input voltage u, gain K and time constant τ.

2.2 Model of Connecting Tubes

The tubes, which connect the valves with the actuator chambers, affect the air mass flow. The friction in the tube leads to a loss of flow, which causes a time delay, which is based on the sonic speed c_{sonic}. The incoming mass flow $\dot{m}_{in}$ can be calculated using the tube diameter d and the tube length l with

$$\dot{m}_{in}(t) = \phi \dot{m}_v\left(t - \frac{l}{c_{sonic}}\right). \tag{9}$$

Here, ϕ is the attenuation coefficient and it is calculated with

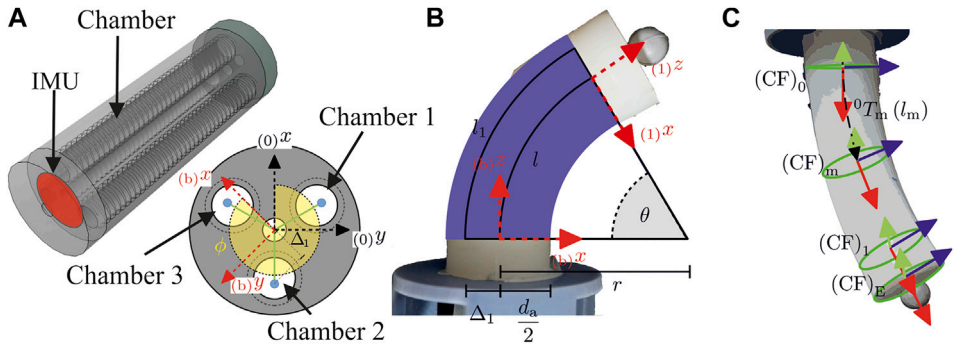


FIGURE 2 | Geometry of soft pneumatic actuator with PCC parameters ϕ, θ, l, top view **(A)**, side view **(B)** and coordinate frames used for the reconstruction model **(C)**.

$$\phi = e^{-\frac{R_t R T}{2 p_{end}} \frac{l}{c_{sonic}}}. \tag{10}$$

The pressure p_{end} is measured at the end of the tube. The friction resistance is described in Ibrahim et al. (2019) as

$$R_t = \frac{0.158}{d^2}\left(\frac{BT^{3/2}}{T+S}\right)^{1/4}\left(4\frac{\dot{m}_v}{d\pi}\right)^{3/4}. \tag{11}$$

with the model of Sutherland and its constant $B = 1.4747 \times 10^{-6} Pas/\sqrt{k}$ and a substance-specific temperature S. By using short and wide tubes, the friction and time delay are minimal and can be neglected. With airtight connectors and tanks, the leakage is also minimal and therefore neglected as well.

2.3 Model of Soft Pneumatic Actuator

In the following actuator modeling is presented using the example of a FRA made of Dragonskin 10 silicon [Polygerinos et al. (2015)]. **Figure 2A** shows the actuator segment. The Deformation of the actuator is due to expansion of the chambers, which are located along the actuator length. First the entire actuator is considered and a geometric model is created. Then the dynamics of an individual chambers of the actuator are considered and modeled.

2.3.1 Geometric Modeling

Based on work from Webster and Jones (2010) the actuator's shape is approximated with a piece-wise constant curvature (PCC). The configuration is described with the arc length l_a, the bending direction ϕ and the bending angle $\theta = (1/r)l_a$ with bending radius r. As seen in **Figure 2**, a segment with length l_a consists of three symmetrical arranged chambers with a length of l_i. An elongation of at least one of these chambers leads to a bending and extension of the segment. For a mapping between the PCC parameters (ϕ, θ, l_a) and the task space coordinates the homogeneous transformation matrix

$${}^0\boldsymbol{T}_{1PCC} = \begin{bmatrix} {}^0\boldsymbol{R}_{1PCC} & {}_{(1)}\boldsymbol{t} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} c_\phi^2 c_\theta + s_\phi^2 & s_\phi c_\phi (c_\theta - 1) & c_\phi s_\theta & c_\phi (1 - c_\theta)\frac{l_a}{\theta} \\ s_\phi c_\phi (c_\theta - 1) & s_\phi^2 c_\theta + c_\phi^2 & s_\phi s_\theta & s_\phi (1 - c_\theta)\frac{l_a}{\theta} \\ -c_\phi s_\theta & -s_\phi s_\theta & c_\theta & s_\theta \frac{l_a}{\theta} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{12}$$

from Webster and Jones (2010) is used. This describes the transformation between the segment's base $(CF)_0$ and the end of the PCC part $(CF)_1$ (**Figure 2**). For static parts of the segment a linear displacement d_E, that results in a transformation ${}^1_T\mathrm{E}$ with ${}^1\boldsymbol{R}_E = \boldsymbol{I}$ and thus only a translation vector ${}_{(E)}\boldsymbol{t} = (0, 0, d_E, 1)^T$, is added.

The mapping between the PCC parameters and the chamber length is also given in Webster and Jones (2010). For each chamber i the arc length is

$$l_i = l_a - d_i \cos(\sigma_i - \phi)\theta. \tag{13}$$

The chamber position is specified by the distance d_i to the central axis and the angle σ_i to the ${}_{(0)}\boldsymbol{x}$-axis.

2.3.2 Chamber Modeling

The basis for chamber dynamics is the low-level model from Ibrahim et al. (2019). This describes the pressure curve depending on incoming and outgoing mass flow $\dot{m}_{in}$ and $\dot{m}_{out}$ respectively, as well as changes in volume $\dot{V}$. Considering a chamber with a volume V, the pressure change is described with

$$\dot{p} = \frac{RT}{V}\left(\alpha_{in}\dot{m}_{in} - \alpha_{out}\dot{m}_{out}\right) - \alpha\frac{p_i}{V}\dot{V}. \tag{14}$$

The chamber's volume is affected by the difference between internal pressure and atmospheric pressure. Since the FRA only expands the chambers in one axial direction, the spherical approach from Ibrahim et al. (2019) is not suitable. For this reason, a cylinder model is constructed in the following. The volume of a cylinder is

$$V = \pi r_c^2 l, \tag{15}$$

and it is described by the circular base with the chamber radius r_c and the chamber length l. At idle state, the pressure in the actuator chamber is p_{atm} and the volume is $V = \pi r_c^2 l_0$ with the initial length l_0.

Analogous to the sphere model, the force due to the pressure difference $p_{diff} = p - p_{atm}$ is

$$F_p = p_{diff} \pi r_c^2 \tag{16}$$

and the force of the material tension is

$$F_\sigma(l) = \sigma_c(\epsilon) 2\pi r_c w_c, \tag{17}$$

with w_c as the chamber wall thickness. The elongation depends on the length change $\epsilon = l - l_0/l_0$ of the actuator chamber. Based on Ibrahim et al. (2019) and the cylindrical shape the chamber's dynamic is modeled with a second order nonlinear differential equation

$$M\ddot{l} + D\dot{l} = F_p - F_\sigma(l). \tag{18}$$

The coefficient M is the chamber's mass and the coefficient D describes the chamber's damping.

2.4 State Space Representation

Using the equations above, a state-space representation of the soft robot system is set up. For this purpose, a segment with three chambers is considered. In **Figure 1** the control loop is shown. For each chamber a valve is used to regulate the in- and outgoing mass flow. This flow causes a pressure change in the actuators chambers and as a result the chambers in- or deflate. The pressure in each chamber is measured, as well as the actuator's orientation at a certain point along the arc.

2.4.1 System Dynamics

The system dynamics

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \left(\dot{p}_1, \dot{p}_2, \dot{p}_3, \ddot{l}_1, \ddot{l}_2, \ddot{l}_3, \dot{l}_1, \dot{l}_2, \dot{l}_3\right)^{\mathrm{T}}, \tag{19}$$

describes the change of the state variables $\boldsymbol{x} = (\boldsymbol{p}, \dot{\boldsymbol{l}}, \boldsymbol{l})^{\mathrm{T}}$. These are the chamber pressures $\boldsymbol{p} = p(_1, p_2, p_3)^{\mathrm{T}}$, the chamber lengths $\boldsymbol{l} = (l_1, l_2, l_3)^{\mathrm{T}}$ and its derivative $\dot{\boldsymbol{l}} = (\dot{l}_1, \dot{l}_2, \dot{l}_3)^{\mathrm{T}}$. The product of the valve's opening cross-section A_i and the flow coefficient $c_{f,i}$ is selected as system input $\boldsymbol{u} = (u_i, u_2, u_3)^{\mathrm{T}} = (c_{f,1}A_1, c_{f,2}A_2, c_{f,3}A_3)^{\mathrm{T}}$. Introduced in Ibrahim et al. (2019), the use of fast switching valves allows to neglect the spool dynamic **Eq. 7**.

The states are described, based on **Eq. 14**, as

$$\dot{p}_i = \frac{RT_a}{\pi r_c^2 l_i}\left(\alpha_{in,i}\dot{m}_{in,i}(u_i, p_i) - \alpha_{out,i}\dot{m}_{out,i}(u_i, p_i)\right) - \alpha_i \frac{p_i}{l_i}\dot{l}_i, \tag{20}$$

and based on **Eq. 18** as

$$\ddot{l}_i = \frac{r_{c,i}^2}{M_i}\left((p_i - p_{atm})\pi - \sigma_{c,i}(\epsilon) 2\pi - \frac{D_i}{r_{c,i}^2}\dot{l}_i\right), \tag{21}$$

with $i = 1, 2, 3$. For mass flow equation please refer to (Section Model of Valve).

2.4.2 Measurement Model

The measurement model is used for mapping between the state space $\boldsymbol{x}$ and the measurement output $\boldsymbol{y} = (p_1, p_2, p_3, \gamma_x, \gamma_y)^{\mathrm{T}}$ with bending angles γ_x and γ_y. The pressures $\boldsymbol{p} = (p_1, p_2, p_3)^{\mathrm{T}}$ are both state and measurement quantities. The orientation at any point of the arc is represented with Euler angles in RPY notation. The corresponding rotation is described with the rotation matrix

$$\boldsymbol{R}_{RPY} = \begin{bmatrix} c_{\gamma_z} c_{\gamma_y} & c_{\gamma_z} s_{\gamma_y} s_{\gamma_x} - s_{\gamma_z} c_{\gamma_x} & c_{\gamma_z} s_{\gamma_y} c_{\gamma_x} + s_{\gamma_z} s_{\gamma_x} \\ s_{\gamma_z} c_{\gamma_y} & s_{\gamma_z} s_{\gamma_y} s_{\gamma_x} + c_{\gamma_z} c_{\gamma_x} & s_{\gamma_z} s_{\gamma_y} c_{\gamma_x} - c_{\gamma_z} s_{\gamma_x} \\ -s_{\gamma_y} & c_{\gamma_y} s_{\gamma_x} & c_{\gamma_y} c_{\gamma_x} \end{bmatrix}. \tag{22}$$

From the states $\boldsymbol{l} = (l_1, l_2, l_3)^{\mathrm{T}}$, the PCC parameters (ϕ, θ, l_a) are calculated first and then the bending angles γ_x and γ_y at arc position l_m are determined by comparison of the entries of the rotation matrices **Eqs. 12**, **22**.

If the quotient

$$\frac{l_2 - l_1}{l_3 - l_2} = \frac{d_1 \cos(\sigma_1 - \phi) - d_2 \cos(\sigma_2 - \phi)}{d_2 \cos(\sigma_2 - \phi) - d_3 \cos(\sigma_3 - \phi)}, \tag{23}$$

is formed from **Eq. 13** and the addition theorem $\cos(\alpha - \beta) = \cos\alpha\cos\beta + \sin\alpha\sin\beta$ is applied, the equation

$$\tan\phi = \frac{\frac{l_2 - l_1}{l_3 - l_2}(d_2 \cos\sigma_2 - d_3 \cos\sigma_3) - (d_1 \cos\sigma_1 - d_2 \cos\sigma_2)}{(d_1 \sin\sigma_1 - d_2 \sin\sigma_2) - \frac{l_2 - l_1}{l_3 - l_2}(d_2 \sin\sigma_2 - d_3 \sin\sigma_3)}, \tag{24}$$

results. With

$$l_2 - l_1 = \theta\left(d_2 \cos(\sigma_2 - \phi) - d_1 \cos(\sigma_1 - \phi)\right). \tag{25}$$

the angle

$$\theta = \frac{l_2 - l_1}{d_2 \cos(\sigma_2 - \phi) - d_1 \cos(\sigma_1 - \phi)}, \tag{26}$$

is determined. Transposing **Eq. 13**, the arc length is

$$l_a = l_1 + d_1 \cos(\sigma_1 - \phi)\theta. \tag{27}$$

Getting from PCC parameters to RPY angles, we first determine the rotation matrix at the measuring position. Based on the PCC parameters and the measurement position l_m we determine the rotation matrix ${}^0\boldsymbol{R}_m(l_m)$. For a measurement position at any point $l = l_m$ on the central arc, the transformation matrix ${}^0\boldsymbol{T}_m(l_m)$ is based on ${}^0\boldsymbol{T}_{1PCC}$ from **Eq. 12** with bending angle

$$\theta_m = \theta \frac{l_m}{l_a}, \tag{28}$$

and arc length l_m.

From the comparison of the rotation matrices **Eqs. 12**, **22** follows

$$\gamma_y = \arctan2\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2} \right). \tag{29}$$

If $\cos(\gamma_y) = 0$, the angle $\gamma_x = 0$. For other cases

$$\gamma_x = \arctan2\left(\frac{r_{32}}{\cos\gamma_y}, \frac{r_{33}}{\cos\gamma_y} \right), \tag{30}$$

applies. From these equations the measurement model can be set up with

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{p} \\ \gamma_x \\ \gamma_y \end{bmatrix}. \tag{31}$$

The state space representation consists of $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ and $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$ is used to analyze and simulate the system. In practical applications not only the description of the system behavior is relevant. Furthermore, a consideration of the states $\boldsymbol{x}$ during operation is essential.

In summary, the measurement model is based on the correspondence of the rotation matrices, which was established on the one hand by the sensor values in RPY coordinates and on the other hand by the approximation of the actuator shape by the PCC parameters. The offset between the actuator's tip and the measurement position is also included by shifting the position with the PCC parameters.

3 RECONSTRUCTION

In the control loop shown in **Figure 1**, the controlled variable is the length of the chambers. Since the lengths are not directly measurable, a reconstruction is necessary. In the following a model for reconstruction is described, which determines the state of the actuator from sensor measurements.

3.1 Reconstruction With a Static Inverse Measurement Model

The pressures $\boldsymbol{p}$ and the orientation at a certain point on the arc are available as measured system outputs. The relationship between the measured and state variables is determined by the measurement model from **Eq. 31**. The pressure is directly mapped from state to system output. The inverse function of the measurement model is not sufficient to determine the actuators shape, because the mapping is not bijective. With the measured orientation, only a relative chamber length is captured. Therefore the overall actuator length is unknown and a reconstruction of the length is performed. Mapping the system output to the state variables, the inverse measurement model

$$\begin{bmatrix} \boldsymbol{p} \\ l_1 \\ l_2 \\ l_3 \end{bmatrix} = \boldsymbol{g}^{-1} \begin{pmatrix} \boldsymbol{p} \\ \gamma_x \\ \gamma_y \end{pmatrix}, \tag{32}$$

is formed with the measurement equations that have already been established as well as the system dynamics. In a first step the orientations measurement is used to calculate the PCC parameters ϕ_m and θ_m at a measurement position l_m. The arc length l_a cannot be determined, because it does not affect the orientation as seen in **Eq. 12**.

Unlike the two measurement variables γ_x and γ_y the rotation γ_z around the z-axis is unknown. Checking matrix **Eq. 12**, it becomes apparent that the entries r_{12} and r_{21} are identical. To match the rotation matrices, this must also apply to ${}^1\boldsymbol{R}_{0\mathrm{RPY}}$ on **Eq. 22**. Thus follows

$$r_{12,\mathrm{RPY}} = r_{21,\mathrm{RPY}}, \tag{33}$$

$$\sin\gamma_z\cos\gamma_y = \cos\gamma_z\sin\gamma_y\sin\gamma_x - \sin\gamma_z\cos\gamma_x, \tag{34}$$

and therefore for the angle

$$\gamma_z = \arctan_2\left(\sin\gamma_x\sin\gamma_y, \cos\gamma_y + \cos\gamma_x\right). \tag{35}$$

If the rotation matrices **Eqs. 12**, **22** are compared with each other, the bending direction can be found in

$$\tan\phi_m = \frac{\sin\phi_m}{\cos\phi_m} = \frac{r_{3,2}}{r_{3,1}}. \tag{36}$$

This Results in the Following Angles

$$\phi_m = \arctan_2\left(c_{\gamma_y} s_{\gamma_x}, -s_{\gamma_y}\right) + \pi, \tag{37}$$

To get the bending angle θ we need

$$\tan\theta_m = \frac{\sin\theta_m}{\cos\theta_m} = \frac{r_{1,3}\cos\phi + r_{2,3}\sin\phi}{r_{3,3}}, \tag{38}$$

and so it is

$$\theta_m = \left|\arctan2\left[\left(c_{\gamma_z} s_{\gamma_y} c_{\gamma_x} + s_{\gamma_z} s_{\gamma_x}\right) c_\phi + \left(s_{\gamma_z} s_{\gamma_y} c_{\gamma_x} - c_{\gamma_z} s_{\gamma_x}\right) s_\phi, c_{\gamma_y} c_{\gamma_x}\right]\right|. \tag{39}$$

at the measurement position l_m. Since the arctan definition range is $[-\pi, \pi]$, **Eq. 37** is shifted by π, so ϕ_m is in the $[0, 2\pi]$ PCC definition range. The angle θ_m is positively defined, hence no full consideration of all quadrants of the inverse angle function arctan is necessary for the PCC parameter.

In contrast to ϕ_m and θ_m, the arc length l_a cannot be reconstructed from the orientation measurement. A reconstruction based on the actuator's model is necessary. Considering a static case the force equilibrium is

$$F_p = F_\sigma. \tag{40}$$

With **Eqs. 16**, **17**, strain, based on pressure, is

$$\epsilon_i = \sigma_c^{-1}\left(\frac{p_{\mathrm{diff},i} r_c}{2 w_c}\right). \tag{41}$$

Through the defined strain the chamber lengths

$$l_i = (1 + \epsilon_i) l_{0,i}, \tag{42}$$

can be determined. With **Eqs. 24**, **26** the arc length l_a is known from **Eq. 27**.

From the reconstructed arc length l_a the bending angle at segment end

$$\theta = \theta_m \frac{l_a}{l_m} \tag{43}$$

can be derived with **Eq. 28**. In a last step the chamber lengths l_i are determined with **Eq. 13**.

Reconstruction of the state variables was also performed by comparing the rotation matrices. With the orientation measurement, the PCC parameters can be determined at the measurement position. Since the actuator length cannot be found using the orientation measurement, it was necessary to look at the actuator forces. For static case the state variables can be reconstructed now.

3.2 Measurement Devices for Shape Sensing

In this research, a camera tracking system and an IMU are used to capture actuator's shape. The camera tracking system is used for identification and validation experiments and the IMU is used for orientation measurement of the actuator's tip (**Figure 2A**).

3.2.1 Camera Tracking System

An OptiTrack Flex three camera system is installed to track the FRA's segment tip. The system is infrared based, therefore reflecting markers are attached to the end of the FRA. With a resolution of 100 frames per second, 2D images of six cameras are reconstructed into a 3D representation, thus calculating the tip's position. As a result, it is possible to record the position of the actuator with the cameras in a cycle of 100 Hz.

3.2.2 Inertial Measurement Unit

The IMU Waveshare12476 has an ICM20948 chip, which includes a compass, a gyroscope and an accelerometer. The rotations γ_x^{IMU}, γ_y^{IMU} and γ_z^{IMU} in the frame of the IMU $(CF)_{IMU}$ can be estimated. The IMU uses the earth's gravitational force (direction of the z-axis) and the earth's magnetic field (direction of the y-axis) for the orientation of the basic coordinate system. Furthermore, the IMU includes a processor for motion processing algorithms, which forwards the data *via* the I2C bus to the host processor. In this setup, a clock rate of 40 Hz is achieved.

4 IDENTIFICATION

In the previous sections model equations, which depend on various parameters, have been derived. Therefore the parameters have to be determined. Some parameters are based on literature, others can be found in CAD models or can be measured directly. However, a few parameters cannot be determined directly and thus they must be identified. In the following, parameters to be determined are highlighted and their identification procedures are described.

4.1 Parameter of Valve Model

The function of the valves is described with the mass flow **Eq. 1**. The following parameters have to be defined:

- The ideal gas constant R and the isentropic exponent κ,
- discharge coefficient c_f,
- as well as the mapping between the valve's orifice $A(u)$ and the input voltage u.

A detailed description of parameter choice and identification can be found in Krause et al. (2019).

4.2 Parameter of Actuator Model

Regarding the actuator, a distinction is made between chamber modeling and geometric modeling of the entire actuator. First, the chamber dynamics is considered. For **Eqs. 14**, **18**, the parameters needed are

- the coefficients α_{in} and α_{out} based on the occurring heat transfer,
- the stress-strain curve $\sigma(\epsilon)$,
- the chamber radius r_c and wall thickness w_c and the
- chamber's mass M and damping D.

The identification process of these parameters is also mentioned in Krause et al. (2019). In addition to the procedure mentioned above, an identification of the actuator geometry is carried out. Also, a more precise volume description for identifying the stress strain curve is possible. The actuator's geometric model is parameterized with

- the chamber positions, that consist of the angle σ_i and the offset d_i to the central axis,
- the offset from the end of the PCC segment to the end effector d_E,
- as well as the length of the unstressed PCC segment l_0.

The chamber position is based on the design of the actuator's mold. If the three chambers are arranged as in **Figure 2**, their position is specified with the angle

$$\sigma_i = \frac{2i+1}{3}\pi. \tag{44}$$

Assuming a symmetric design, the offsets are equal with $d_i = d_c$. The chamber displacement d_c and the linear distance to the end effector d_E are based on the actuator's CAD data. The initial actuator length l_0 of the PCC segment needs to be identified. For this purpose, the pressure control from Ibrahim et al. (2019) is used to deflect the actuator in different bending directions and angles. The true position is recorded with the camera tracking system described in Section Camera tracking system. A marker is attached at the end effector with a displacement d_{mk}. The camera tracking system records the marker position ${}_{(cam)}\boldsymbol{r}_{mk}$ in the camera frame $(CF)_{cam}$. This is calibrated with a ground plane to match its frame orientation ${}^{cam}\boldsymbol{R}_0$ and origin to the actuators base $(CF)_0$. Only a displacement

d_{top} at the top of the actuator mount is left. Thus, the transformation is

$$ {}_{\text{cam}}\boldsymbol{T}_0 = \begin{bmatrix} {}^{\text{cam}}\boldsymbol{R}_0 & {}_{(\text{cam})}\boldsymbol{r}_{\text{top}} \\ 0 & 1 \end{bmatrix}, \tag{45} $$

with ${}^{\text{cam}}\boldsymbol{R}_0 = \boldsymbol{I}$ and ${}_{(\text{cam})}\boldsymbol{t}_{\text{top}} = [0, 0, d_{\text{top}}, 1]^{\text{T}}$. With the homogeneous transformation matrix ${}^0\boldsymbol{T}_1$ from Webster and Jones (2010), the marker position ${}_{(1)}\boldsymbol{r}_{\text{mk}} = [0, 0, d_{\text{mk}}, 1]^{\text{T}}$ is mapped to the base coordinate system $(\text{CF})_0$. Hence, the estimated marker's position is

$$ {}_{(\text{cam})}\widehat{\boldsymbol{r}}_{\text{mk}} = {}^{\text{cam}}\boldsymbol{T}_0\,{}^0\boldsymbol{T}_{1\text{PCC}\,(1)}\boldsymbol{r}_{\text{mk}}. \tag{46} $$

To estimate the marker position, the transformation matrix ${}^0\boldsymbol{T}_{1\text{PCC}}$ and therefore the PCC parameters are needed.

For measurement, an IMU is used. As described in Section Reconstruction with a static inverse measurement model, an estimation of the actuator elongation is necessary. A special actuator design leads to constraints for the arc length l_{a}. If there is a construction with high stiffness in the longitudinal side, one can set $l_{\text{a}} = l_0 =$ constant. This is the case for the 3D printed PneuNet actuator from Garcia et al. (2020). While the chambers lengthen and shorten during the bending of the actuators, the central axis does not change in length.

If only positive elongation of the chambers is possible, the arc length l_{a} is not fixed and approximated as a function of the bending angle θ. Assuming a linear relationship, we estimate the change of arc length to be

$$ l_{\text{a}} = l_0 + d_{\text{n}}\theta. \tag{47} $$

This is equivalent to a displacement of a neutral axis in bending direction, where there is no strain. This behavior is typical for the fiber-reinforced actuator with at least one relaxed chamber. For identification with length approximation and camera tracking system the.

- displacement d_{top} of the camera base frame,
- the displacement of the neutral axis d_{n}
- and the marker offset d_{mk}

are also needed.

4.2.1 Orientation of the Inertial Measurement Unit

The orientation of the IMU is recorded at $l_{\text{m}} = l_{\text{a}}$, the tip of the segment. For IMU measurement the transformation ${}^{\text{IMU}}\boldsymbol{R}_1$ is unknown. The rotation matrix

$$ {}^{\text{IMU}}\boldsymbol{R}_1 = \boldsymbol{R}\left(\phi_{\text{z}}, \phi_{\text{y}}, \phi_{\text{x}}\right), \tag{48} $$

is built with RPY angles $(\phi_{\text{z}}, \phi_{\text{y}}, \phi_{\text{x}})$, which must also be identified. To determine the PCC parameter, the rotation matrices

$$ {}^0\boldsymbol{R}_{1\text{PCC}} = {}^0\boldsymbol{R}_{\text{IMU}}\,{}^{\text{IMU}}\boldsymbol{R}_1 \tag{49} $$

must be equal. For the identification routine, we first determine the yaw angle θ_z. This is accomplished similar to **Eq. 35**. By inspecting the PCC rotation matrix **Eq. 12**, notably the entries $r_{1,2}$ and r_{21} are identical. From this relation and from **Eq. 49** the yaw angle is determined with

$$ \tan\theta_z = \frac{\left(s_{\phi_y} c_{\theta_x} - c_{\phi_y} s_{\phi_z} s_{\theta_x}\right) s_{\theta_y} - c_{\phi_y} c_{\phi_z} c_{\theta_y} - s_{\phi_x} c_{\phi_y} s_{\theta_x} + \left(c_{\phi_x} c_{\phi_z} + s_{\phi_x} s_{\phi_y} s_{\phi_z}\right) c_{\theta_x}}{\left(s_{\phi_x} c_{\phi_y} c_{\theta_x} + \left(s_{\phi_x} s_{\phi_y} s_{\phi_z} + c_{\phi_x} c_{\phi_z}\right) s_{\theta_x}\right) s_{\theta_y} + \left(s_{\phi_x} s_{\phi_y} c_{\phi_z} - c_{\phi_x} s_{\phi_z}\right) c_{\theta_y} + s_{\phi_y} s_{\theta_x} + c_{\phi_y} s_{\phi_z} c_{\theta_x}}. \tag{50} $$

With **Eqs. 36**, **38** the bending direction ϕ and the bending angle θ can be derived from the rotation matrix of the IMU. The arc length l_{a} is approximated with **Eq. 47**.

With these PCC parameters the homogeneous transformation matrix ${}^0\boldsymbol{T}_{1\text{PCC}}$ is built and the marker position ${}_{(\text{cam})}\widehat{\boldsymbol{r}}_{\text{mk}}$ can be estimated from IMU measurements with **Eq. 46**.

4.2.2 Optimization

With particle swarm optimization, the parameter values are optimized to fit the calculated positions ${}_{(\text{cam})}\widehat{\boldsymbol{r}}_{\text{mk}}$ from the sensor to the true data from the camera tracking system. The cost function is built with the Euclidean distance of the marker positions. In order to consider the measurements in the deflected state more intensely, the costs are increased with the Euclidean distance in the (xy)-plane. Consequently, the cost function for N measurements is

$$ c = \sum_{i=1}^{N} \frac{\left\| {}_{(\text{cam})}\widehat{\boldsymbol{r}}_{\text{mk},i} - {}_{(\text{cam})}\boldsymbol{r}_{\text{mk},i} \right\| \sqrt{\left({}_{(\text{cam})}\widehat{\boldsymbol{x}}_{\text{mk},i} - {}_{(\text{cam})}\boldsymbol{x}_{\text{mk},i}\right)^2 + \left({}_{(\text{cam})}\widehat{\boldsymbol{y}}_{\text{mk},i} - {}_{(\text{cam})}\boldsymbol{y}_{\text{mk},i}\right)^2}}{N}. \tag{51} $$

The identification measurement is recorded with pressure steps in each chamber separately and in pairs of two. This movement covers many operation points. After the oscillation has subsided, the measurement data of each stage i is recorded and averaged for noise reduction. This provides the identification data set. The identification of the IMU sensor results in a mean error $\| {}_{(\text{cam})}\widehat{\boldsymbol{r}}_{\text{mk}} - {}_{(\text{cam})}\boldsymbol{r}_{\text{mk}} \|$ of 1.6 mm and a standard deviation of 0.9 mm. For validation, sine pressure curves with different phase shifts are recorded. Here, again a path with different operation points is selected. The path in Cartesian x, y and z-direction is shown in **Figure 3**. Furthermore, the deviations of the individual coordinates between IMU and the camera tracking system are shown in **Figure 3**. The validation results in a mean error of 4.1 mm with a standard deviation of 0.9 mm. The largest deviations occur at changes of the moving direction.

For length reconstruction, based on strain from **Eq. 41**, the relation $\sigma(\epsilon)$ is needed. At steady state, the pressure p_i and the chambers' lengths l_i are recorded. If only one chamber is actuated, there is a bending dependent extension of the arc length. First, the PCC parameters from **Eqs. 36**, **38** and the length $l_{\text{a}} = l_0 + d_{\text{n}}\theta$ as well as the real bending angle **Eq. 43** are determined. With this configuration the chambers' lengths can be calculated by **Eq. 13**. To prevent falsification due to wrong identification of chamber radius r_{c} and wall thickness w_{c}, the augmented stiffness

$$ S(\epsilon) = \sigma(\epsilon)\frac{w_{\text{c}}}{r_{\text{c}}} = \frac{p_{\text{diff}}}{2}, \tag{52} $$

is identified. With different steady states, a look-up table for the stress is filled. The results for all three chambers are shown in

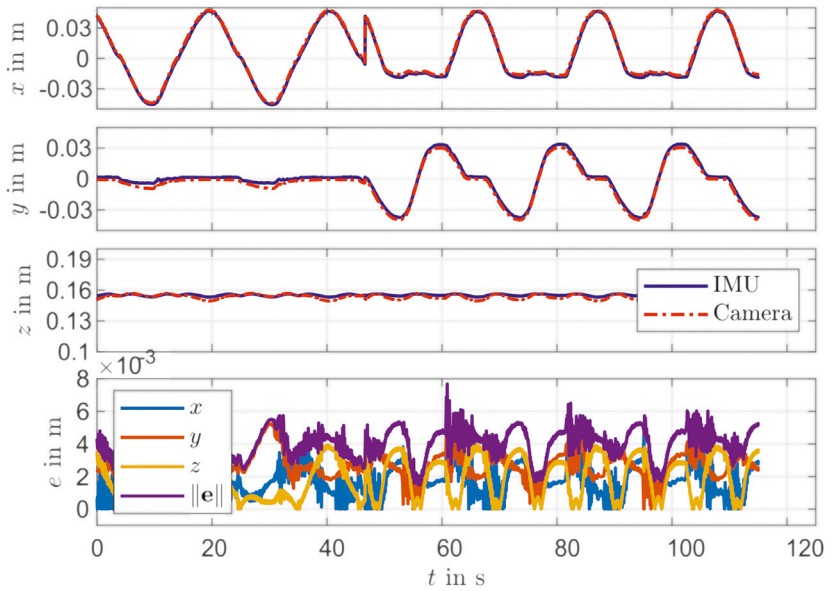


FIGURE 3 | Validation for IMU identification.

Figure 4. The values of the individual chambers differ due to manufacturing tolerances. It should be noted here that the elongation of an individual chambers refers to the length l_0 of the central axis of the entire PCC segment. Therefore an elongation $\epsilon \neq 0$ is possible although the material is not under tension.

5 CLOSED-LOOP CONTROL USING THE RECONSTRUCTION MODEL

In this section the previously described reconstruction model is used for low-level closed-loop control of pressure and chambers' length of the FRA. For the pressure control, a sliding mode control (SMC) is used. In our previous research [Ibrahim et al. (2019)], it was shown, that a SMC was worse than a PI controller, due to the lack of information about the volume of the chamber of the actuator. With the information of the chamber length from the reconstruction model in this research and the known radius of the chambers, their volume can be calculated. This is used to design the SMC and the results are compared with a PI controller.

In addition, a closed-loop control for the chambers' length using a PID controller is implemented and evaluated. Here, a path is also traced and the PCC and Cartesian coordinates are considered. In **Figure 1**, the layout of the control system is shown with **w** as reference input and **y** as feedback.

5.1 Closed-Loop Control of the Pressure With a Sliding Mode Control

The control law for sliding mode control is

$$u = u_{\mathrm{eq}} - \xi \mathrm{sat}\left(\frac{s}{\zeta}\right), \tag{53}$$

with its parameters ξ, the maximum gain, and ζ, which depends on a feasible error $\tilde{p}_{\max}$ and control frequency f_c by

$$\zeta = \frac{2\pi f_c}{5}\tilde{p}_{\max}. \tag{54}$$

The tracking error is $\tilde{p} = p_a(t) - p_d(t)$ and leads to

$$s(\tilde{p}) = \left(\frac{\mathrm{d}}{\mathrm{d}t} + \lambda\right)^{n-1}\tilde{p}(t). \tag{55}$$

Its first order $n = 1$ becomes

$$s(\tilde{p}) = \tilde{p} = p_a - p_d \tag{56}$$

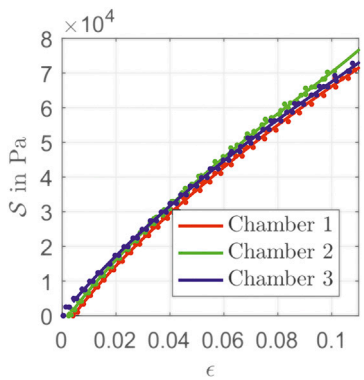


FIGURE 4 | Identified stiffness parameters of the actuator chambers.

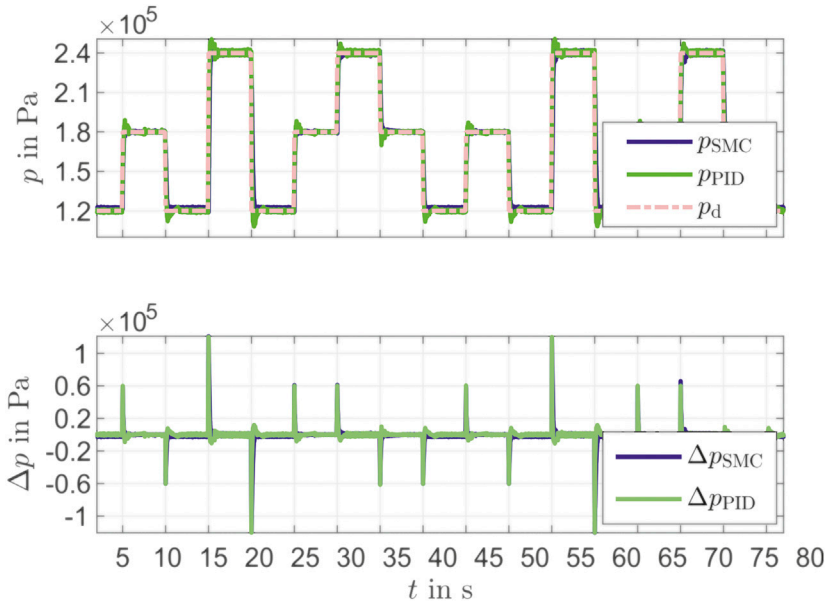


FIGURE 5 | Comparison between SMC and PID Controller for pressure control.

and its time derivative with **Eq. 14** is

$$\dot{s}(\tilde{p}) = \frac{RT}{V}\left(\alpha_{\text{in}}\dot{m}_{\text{in}} - \alpha_{\text{out}}\dot{m}_{\text{out}}\right) - \alpha p_{\text{a}}\frac{\dot{V}}{V} - \dot{p}_{\text{d}}. \tag{57}$$

The condition for equivalent control

$$\dot{s}(\tilde{p}) = 0, \tag{58}$$

is converted to u. If $p_{\text{d}} \geq p_{\text{a}}$ the air has to flow into the actuator. It results in

$$\dot{m}_{\text{out}} = 0, \tag{59}$$

and

$$\dot{m}_{\text{in}} = \frac{V}{\alpha_{\text{in}}RT_{\text{a}}}\left(\dot{p}_{\text{d}} + \alpha p_{\text{a}}\frac{\dot{V}}{V}\right). \tag{60}$$

Inserting **Eq. 1** in **Eq. 60** gives the equivalent control

$$u_{\text{eq}} = c_{\text{f}}A = \frac{\frac{V}{RT_{\text{a}}}\left(\dot{p}_{\text{d}} + \alpha p_{\text{a}}\frac{\dot{V}}{V}\right)}{p_{\text{s}}\Psi\left(p_{\text{s}}, p_{\text{a}}\right)\phi\left(u(t-T), p_{\text{s}}, p_{\text{a}}\right)\alpha_{\text{in}}}. \tag{61}$$

For calculating the attenuation coefficient from **Eq. 10** with tube resistance **Eq. 11**, the previous mass flow and therefore the previous input $u(t-\tau)$ is used. The calculation of the equivalent control for $p_{\text{d}} < p_{\text{a}}$ is determined analogously.

The sliding mode controller was compared with a PI controller (**Figure 5**). Here, two different operation (1.8×10^{-5}Pa and 2.5×10^{-5}Pa) points were approached in one jump and one stair function. The evaluation of the control quality for the steps is shown in **Table 2**. Here, the overshoot, the rising time, the settling time (5%) and the control deviation are considered. The SMC has a lower overshoot at all steps compared to the PID controller. The greater the height of the step, the greater the difference between SMC and PID overshoot (comparison 5 s and 15 s). Since the PID controller is set dynamically, the rising time is shorter than the time of the SMC. The SMC performs better than the PID controller in terms of settling time. The control deviation shows a weakness of the SMC. While with rising steps (5 s, 15 s, 25 s and 30 s) the control deviation between SMC and PID is comparable, SMC shows a clear deviation for falling steps (10 s, 20 s and 35 s). This problem can be solved by optimizing the controller parameters of the SMC [Ibrahim et al. (2019)]. It can be seen that the SMC in combination with the reconstruction model and the IMU provides a better performance than a PID controller for pressure control. Especially with different operation points, the advantages of the SMC become clear.

5.2 Closed-Loop Control of the Chambers' Lengths

Beside the pressure control, a closed-loop control with the previously described state variables l_i (chambers' length) as feedback is considered. The reference variables are the bending direction $\phi(t)$, the bending angle $\theta(t)$ and the segment length l_{a}. With **Eq. 13** each chamber length is calculated and is used as control variable. As a controller, a PID controller designed with Ziegler-Nichols' method is used [Ziegler and Nichols (1942)]. During controller design, it was found, that the chambers behave differently, which can be attributed to manufacturing tolerances. Thus, separate controllers are designed for each chamber of the actuator. The step response of the three chambers for two different operation points (0.144 m and 0.151 m) is shown in **Figure 6**. Furthermore, the deviation of the individual chamber lengths can be taken from **Figure 6**. It is shown, that apart from the steps, the measured chambers lengths follow the desired

TABLE 2 | Performance of SMC and PID controller for pressure control.

Time [s]	Controller	Overshoot [Pa]	Rising time [ms]	Settling time [ms]	Control deviation [Pa]
5	SMC	5,218	92	275	117
	PID	8,943	78	698	124
10	SMC	263	191	4,778	1939
	PID	7,838	92	672	241
15	SMC	3,315	182	275	427
	PID	10,511	78	723	476
20	SMC	368	250	349	1884
	PID	11,920	80	545	219
25	SMC	2,714	90	164	117
	PID	8,185	77	713	131
30	SMC	2,832	136	497	398
	PID	7,187	102	598	475
35	SMC	5,449	114	316	123
	PID	10,078	68	546	129

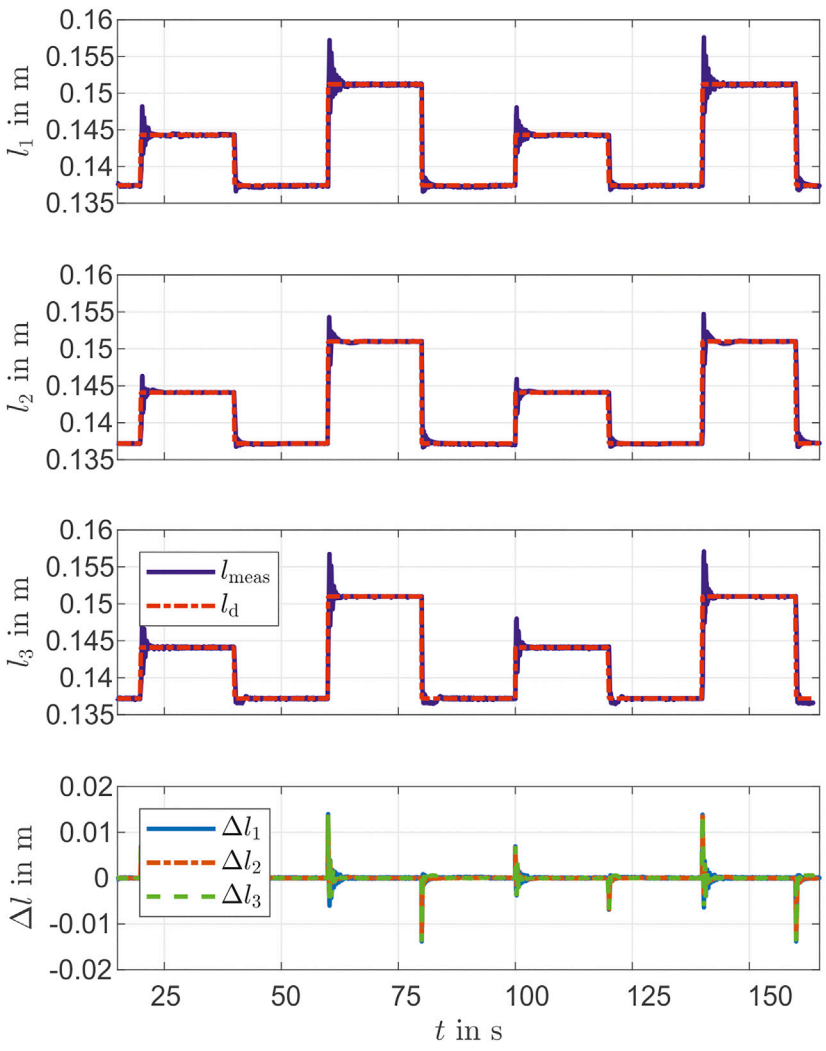


FIGURE 6 | Step response of PID controllers of chambers' lengths.

TABLE 3 | Performance of the PID controller for closed-loop control of the chambers' lengths.

Time [s]	Chamber	Overshoot [m]	Rising time [ms]	Settling time [ms]	Control deviation [m]
20	1	3.88×10^{-3}	136	1870	0.02×10^{-3}
	2	2.22×10^{-3}	132	2,613	0.01×10^{-3}
	3	3.71×10^{-3}	124	1,431	0.04×10^{-3}
40	1	-0.81×10^{-3}	193	761	0.03×10^{-3}
	2	-0.43×10^{-3}	173	940	0.02×10^{-3}
	3	-0.63×10^{-3}	149	1854	0.03×10^{-3}
60	1	5.99×10^{-3}	131	2,605	0.04×10^{-3}
	2	3.28×10^{-3}	160	1,442	0.02×10^{-3}
	3	5.71×10^{-3}	142	1,444	0.01×10^{-3}
80	1	-0.84×10^{-3}	164	930	0.02×10^{-3}
	2	-0.51×10^{-3}	218	1,098	0.03×10^{-3}
	3	-0.71×10^{-3}	170	2,436	0.05×10^{-3}

chambers lengths. The evaluation of controller performance for the first two steps is shown in **Table 3**. For the control deviation, a good value is reached with $< 0.05 \times 10^{-3}$ m for all chambers. Since the controller is set dynamically, the overshoot is large ($< < 6 \times 10^{-3}$ m) but the rising time is small. Within the chambers, chambers one and three show stronger overshoots than chamber 2 with similar rising time. These differences can be explained by the manufacturing tolerances. The overshoots from

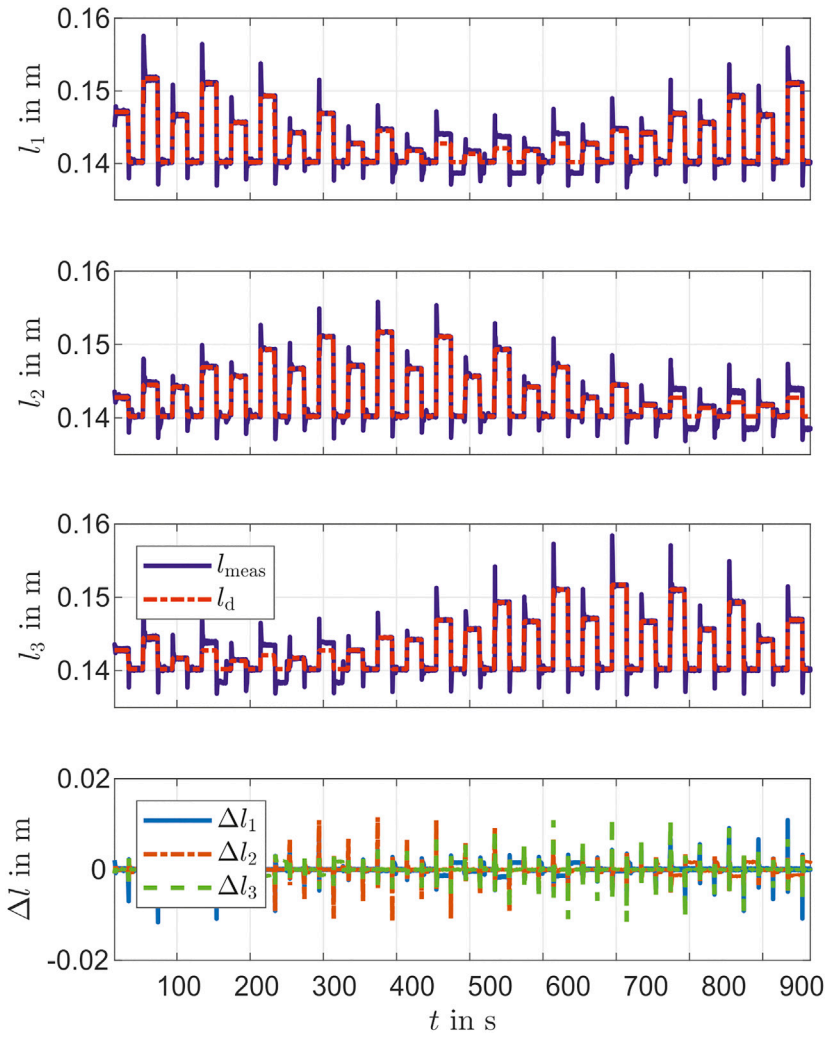


FIGURE 7 | Desired and measured values of the chambers' lengths during validation of PID Controller.

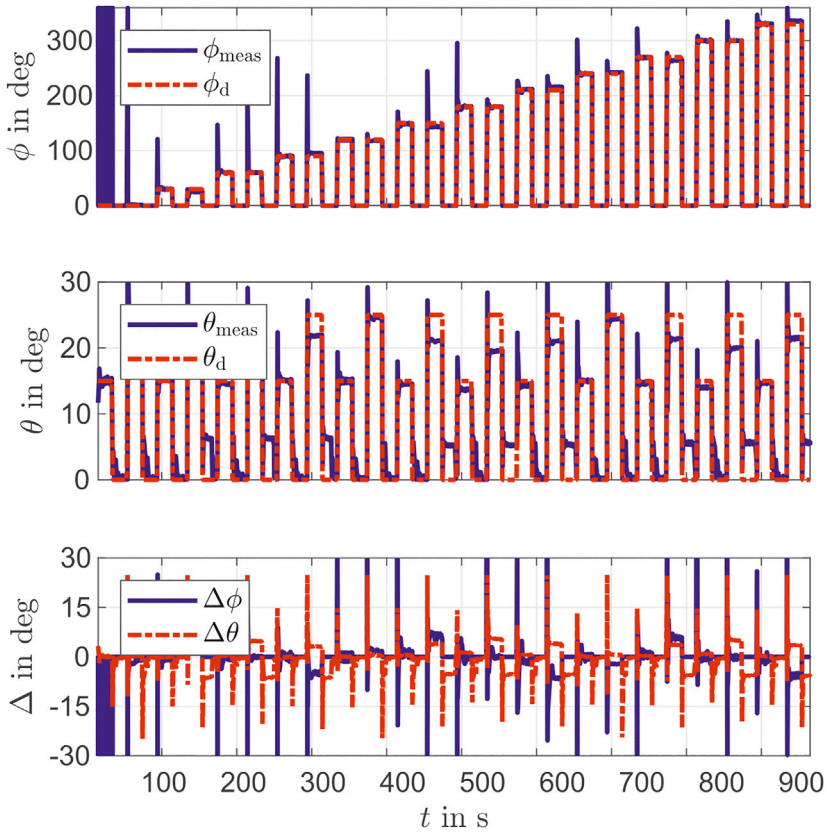


FIGURE 8 | Desired and measured PCC parameters ϕ and θ during validation of PID Controller.

Table 3 and **Figure 6F** do not match, because the Δ of the chambers lengths is shown in the Figure. At the moment of the step there is a dead time, so the delta is greater than the overshoot.

For testing the controller performance, different PCC parameters are specified. With bending angles $\theta = 15^\circ$ and $\theta = 25^\circ$ and actuator length of $l_a = 0.15\,\text{m}$, multiple bending directions $\phi = k30^\circ$ with $k = 0, 1, \ldots, 11$ are used to compute the reference in **Eq. 13**. **Figure 7** shows the lengths of the chambers during the movement. The overshoots are clearly visible in the steps, whereby these increase with increasing step height. This is clearly shown in the Δ of the chamber lengths in **Figure 7**. First l_1 has the largest overshoots, then l_2 and finally l_3. This is due to the dynamic setting of the PID controller. It also becomes, clear that the desired value is not achieved with small chamber lengths. One reason for this could be the stretching of the chamber during the previous actuation. Since no negative pressure is generated, the desired length cannot be achieved. **Figure 8** shows the movement in the PCC parameters ϕ and θ of the actuator. The initial position of the actuator is not defined for the PCC parameters (singularity). For a better view the measurement is set to $\phi = 0$. Also the reference of $\phi = 0^\circ$ lead to results in a neighborhood of $\phi = 360^\circ$. Thus ϕ oscillates at the beginning of the experiment in **Figure 8**. The steps of the desired angle ϕ are well achieved. At the angle θ the larger steps are not quite reached.

The controller performance is validated with the camera tracking system. For this purpose, the desired marker position is determined based on work from Section Reconstruction with a static inverse measurement model and Parameter of Actuator Model. As shown in **Figure 9** there is a mean deviation of $4.3 \times 10^{-3}\,\text{m}$ between the desired path and the reconstructed position. The reconstruction differs from the validation data from the camera tracking system with a mean of $1.9 \times 10^{-3}\,\text{m}$ and a standard deviation of $2.2 \times 10^{-3}\,\text{m}$. The overshoots in x and y are similar in size and the overshoots in z are smaller by a factor of 3. The reason for this is that the influence of the chamber length on x and y is greater than on z.

In addition to a set point stabilization that is done with the steps in the validation above, a control for path tracking is considered, too. For that a circle with a radius of about $31 \times 10^{-3}\,\text{m}$ is constructed with the PCC parameters $\theta(t) = 25^\circ$, $\phi(t) = 360^\circ t/T$ and $l_a = 0.1402\,\text{m}$ is given as reference path. The results for times of circulation $T = 200\,\text{s}$, $T = 100\,\text{s}$ and $T = 5\,\text{s}$ are shown in **Figures 10–12**. It can be seen that there is still a maximum deviation between $4 \times 10^{-3}\,\text{m}$ and $14 \times 10^{-3}\,\text{m}$. The error increases with decreasing of path time for the circle path. The reason for this is the feedback frequency of the IMU. This shows, that a reconstruction model an IMU can be used for suitable low level closed-loop control of a soft pneumatic actuator.

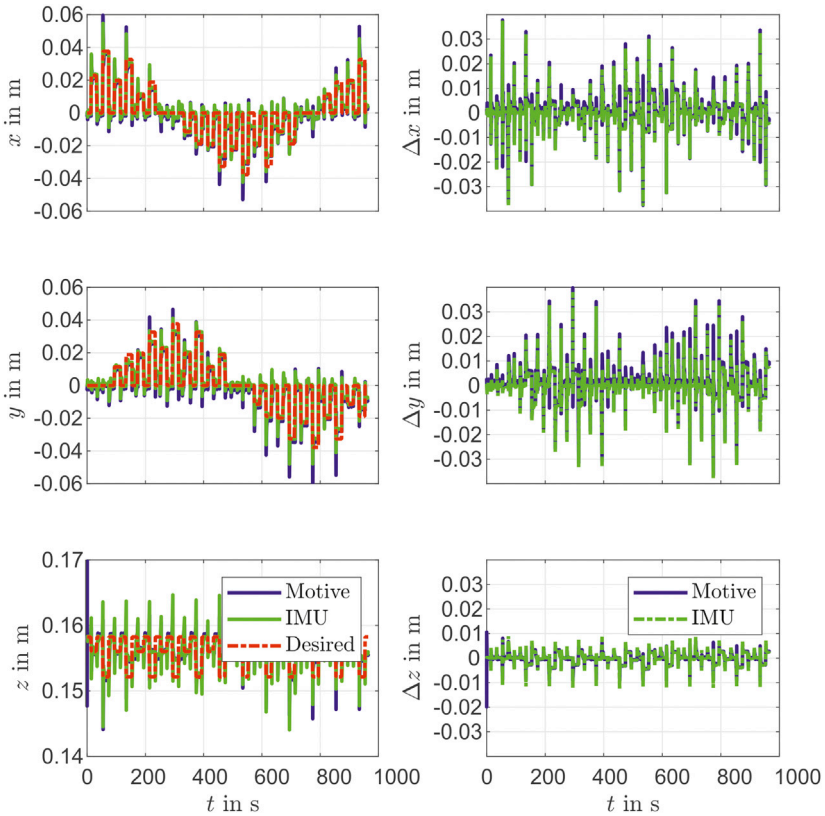


FIGURE 9 | Desired and measured Cartesian parameters during validation of PID Controller.

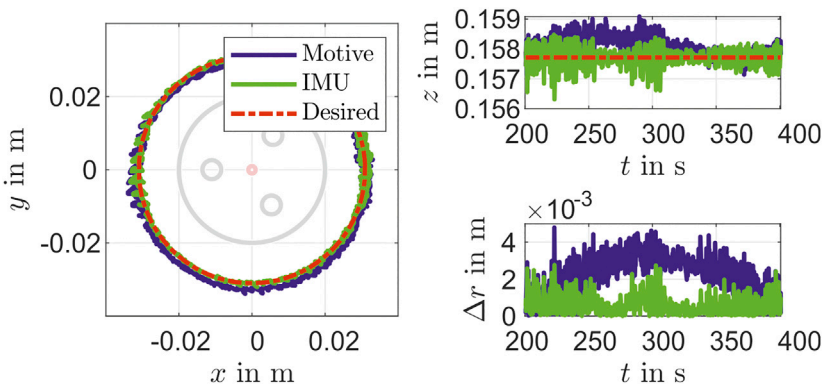


FIGURE 10 | Movement in a circular path in $T = 200$ s.

6 CONCLUSION

In this research, a model for reconstruction of state variables of a soft pneumatic actuator with an inertial measurement unit was demonstrated. A fiber-reinforced soft pneumatic actuator was chosen for the investigation. With the PCC approach, the shape and the deformation variables of the actuator were described and a geometrical model was developed. Then the dynamics of the actuator chambers were modeled using a nonlinear second order differential equation. A state space representation of the soft robotic system was set up with the air pressure, the chambers' length and the first and second time derivation of this as state variables. A measurement model was set up to map between the state variables and the measurement data of the IMU. With the geometric model and data of the pressure and orientation measurement, a reconstruction model for the deformation angles was set up, concerning the specific material properties of the actuator. The reconstruction model was used to determine the volume for a sliding mode controller of

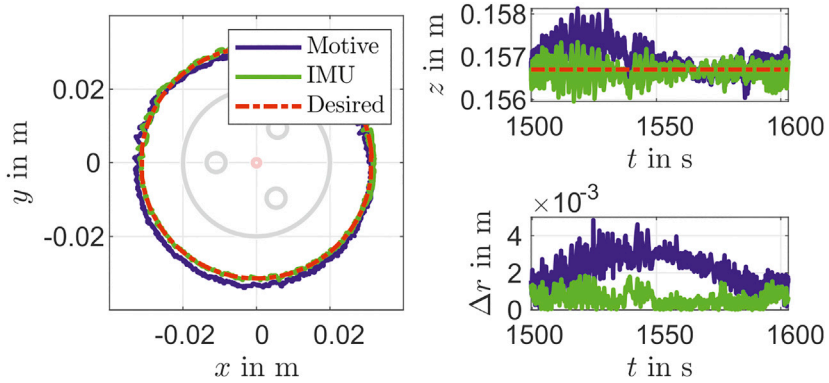


FIGURE 11 | Movement in a circular path in $T = 100\,s$.

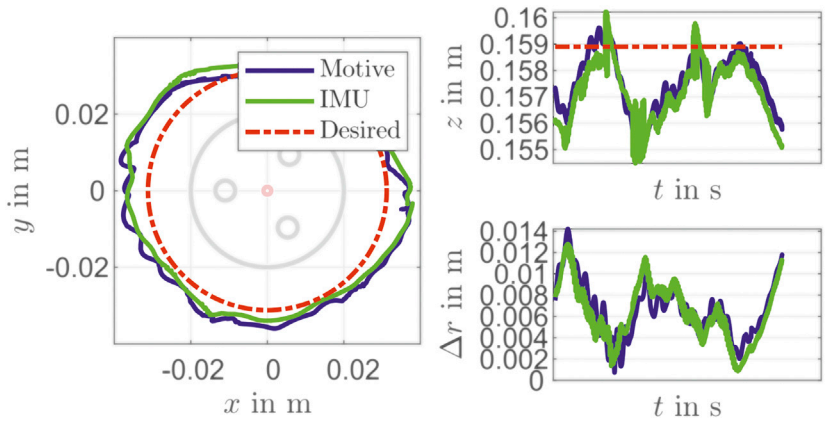


FIGURE 12 | Movement in a circular path in $T = 5\,s$.

pressure. Furthermore, the control of the chambers' lengths of the actuator was investigated.

In the validation of the reconstruction model, a mean error of 4.1 mm with a standard deviation of 0.9 mm results from the camera data for a sinusoidal signal. Also, no large deviations between the reconstruction model and the camera data were detected, during the test of the controller. These have a mean error of 2.2 mm with a standard deviation of 1.9 mm. When designing the PID controller using the reconstruction model for the closed-loop control of the chambers' length, a good control quality were evaluated with settling time $< 2605\,ms$ and an control deviation $< 0.05 \times 10^{-3}\,m$. The controller was set dynamically so that overshoots were present in the step response. For the pressure control, a SMC using the information of the chambers' lengths was designed. The evaluation shows a better performance of the SMC compared to the PI controller, especially with different operation points.

To increase the performance of the controller, it is necessary to increase the feedback frequencies of the IMU. A filtering of the measurement signals can also be considered. Due to the fact that the pressure dynamic differs from the actuator dynamic, the reconstruction of the chambers' length with pressure measurement is insufficient. Therefore, an observer with known model dynamic is necessary. In further work, Kalman-filtering approach for state estimation is recommended. In this approach, different measurement rates and noises from sensors are concerned.

FUNDING

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)-405032969.

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 author.

AUTHOR CONTRIBUTIONS

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

REFERENCES

Al Jaber, F., and Althoefer, K. (2018). "Towards creating a flexible shape senor for soft robots," in IEEE International Conference on Soft Robotics (Robosoft), Livorno, Italy, 114–119. April, 2018. doi:10.1109/ROBOSOFT.2018.8404906

Ben-Dov, D., and Salcudean, S. E. (1995). A force-controlled pneumatic actuator. *IEEE Trans. Robot. Automat.* 11, 906–911. doi:10.1109/70.478438

Best, C. M., Wilson, J. P., and Killpack, M. D. (2015). "Control of a pneumatically actuated, fully inflatable, fabric-based, humanoid robot," in IEEE-RAS 15th international conference on humanoid robots (humanoids). New Jersey, NJ: IEEE, 1133–1140. doi:10.1109/HUMANOIDS.2015.7363495

Donno, M., Palange, E., Di Nicola, F., Bucci, G., and Ciancetta, F. (2008). A new flexible optical fiber goniometer for dynamic angular measurements: application to human joint movement monitoring. *IEEE Trans. Instrum. Meas.* 57, 1614–1620. doi:10.1109/TIM.2008.925336

Felt, W., Suen, M., and Remy, C. D. (2016). "Sensing the motion of bellows through changes in mutual inductance," in International Conference on Intelligent Robots and Systems IROS 2016. New Jersey, NJ: IEEE, 5252–5257. doi:10.1109/IROS.2016.7759772

Galloway, K. C., Polygerinos, P., Walsh, C. J., and Wood, R. J. (2013). "Mechanically programmable bend radius for fiber-reinforced soft actuators," in 16th international conference on advanced robotics (ICAR), November 2013, Montevideo, Uruguay. doi:10.1109/ICAR.2013.6766586

Garcia, M., D. S., Ibrahim, S., Cao, B.-H., and Raatz, A. (2020). Cluj-Napoca, Romania EuCoMeS 2020: New Trends in Mechanism and Machine Science. 488–495. doi:10.1007/978-3-030-55061-5_55

Gerboni, G., Diodato, A., Ciuti, G., Cianchetti, M., and Menciassi, A. (2017). Feedback control of soft robot actuators *via* commercial flex bend sensors. *IEEE/ASME Trans. Mechatron.* 22, 1881–1888. doi:10.1109/TMECH.2017.2699677

Gibbs, P. T., and Asada, H. H. (2005). Wearable conductive fiber sensors for multi-Axis human joint angle measurements. *J. Neuroeng Rehabil.* 2, 7. doi:10.1186/1743-0003-2-7

Ibrahim, S., Krause, J. C., and Raatz, A. (2019). Linear and nonlinear low level control of a soft pneumatic actuator. In IEEE International Conference on Soft Robotics (RoboSoft), February 2019, 434–440. doi:10.1109/ROBOSOFT.2019.8722737

Katzschmann, R. K., Santina, C. D., Toshimitsu, Y., Bicchi, A., and Rus, D. (2019). "Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model." in IEEE International Conference on Soft Robotics (RoboSoft), February 2019, Seoul, South Korea, 454–461. doi:10.1109/ROBOSOFT.2019.8722799

Krause, J. C., Ibrahim, S., and Raatz, A. (2019). "Evaluation environment for control design of soft pneumatic actuators," in *Tagungsband des 4. Kongresses Montage Handhabung Industrieroboter*. Editors T. Schüppstuhl, K. Tracht, and J. Roßmann (Berlin, Heidelberg: Springer, 74–83.

Marchese, A. D., Komorowski, K., Onal, C. D., and Rus, D. (2014). "Design and control of a soft and continuously deformable 2D robotic manipulation system," in IEEE international Conference on Robotics and automation (ICRA). 2189–2196, September, 2014, Hong Kong, China. doi:10.1109/ICRA.2014.6907161

Mengüç, Y., Park, Y.-L., Martinez-Villalpando, E., Aubin, P., Zisook, M., Stirling, L., et al. (2013). "Soft wearable motion sensing suit for lower limb biomechanics measurements." in IEEE Int.ernational Conf.erence on Robotics and Automation, May 2013. 5309–5316. doi:10.1109/ICRA.2013.6631337

Polygerinos, P., Wang, Z., Overvelde, J. T. B., Galloway, K. C., Wood, R. J., Bertoldi, K., et al. (2015). Modeling of soft fiber-reinforced bending actuators. *IEEE Trans. Robot.* 31, 778–789. doi:10.1109/TRO.2015.2428504

Richer, E., and Hurmuzlu, Y. (2000b). A high performance pneumatic force actuator system: Part II-nonlinear controller design. *J. dynamic Syst. Meas. Control*, 122, 426–434. doi:10.1115/1.1286366

Richer, E., and Hurmuzlu, Y. (2000a). A high performance pneumatic force actuator system: Part i—nonlinear mathematical model. *J. of Dyn. Syst., Meas. Control.* 122 3, 416–425. doi:10.1115/1.1286336

Roduit, R., Besse, P.-A., and Micallef, J.-P. (1998). Flexible angular sensor and biomechanical application. *IEEE Trans. Instrum. Meas.* 47, 1020–1022. doi:10.1109/19.744661

Runge, G., and Raatz, A. (2017). A framework for the automated design and modelling of soft robotic systems. *CIRP Ann.* 66, 9–12. doi:10.1016/j.cirp.2017.04.104

Seel, T., Raisch, J., and Schauer, T. (2014). IMU-based joint angle measurement for gait analysis. *Sensors (Basel).* 14, 6891–6909. doi:10.3390/s140406891

Skorina, E. H., Luo, M., Ozel, S., Chen, F., Tao, W., and Onal, C. D. (2015). "Feedforward augmented sliding mode motion control of antagonistic soft pneumatic actuators," in IEEE international conference on robotics and automation (ICRA), May 2015, Seattle, United States, 2544. doi:10.1109/ICRA.2015.7139540

Szelitzky, E., Kuklyte, J., Mandru, D., and O'Connor, N. (2014). Low cost angular displacement sensors for biomechanical applications - a review. *J. Biomed. Eng. Techn.* 2, 21–28. doi:10.12691/jbet-2-2-3

Tan, C. P. E., Nurzaman, S., and Loo, J. (2019). "Non-linear system identification and state estimation in a pneumatic based soft continuum robot," in 2019 3rd IEEE Conference on control technology (CCTA), August 2019. Hong Kong, China, doi:10.1109/CCTA.2019.8920693

Trivedi, D., Rahn, C. D., Kier, W. M., and Walker, I. D. (2008). Soft robotics: biological inspiration, state of the art, and future research. *Appl. Bionics Biomech.* 5, 99–117. doi:10.1080/11762320802557865

Visentin, F., and Fiorini, P. (2018). "A flexible sensor for soft-bodied robots based on electrical impedance tomography,"in IEEE international conference on soft robotics (RoboSoft), Livorno, Italy. 158–163. doi:10.1109/ROBOSOFT.2018.8404913

Webster, R. J., and Jones, B. A. (2010). Design and kinematic modeling of constant curvature continuum robots: a review. *Int. J. Robotics Res.* 29, 1661–1683. doi:10.1177/0278364910368147

Yuen, M. C., Kramer-Bottiglio, R., and Paik, J. (2018). "Strain sensor-embedded soft pneumatic actuators for extension and bending feedback," in IEEE international conference on soft robotics (RoboSoft), April 2018. doi:10.1109/ROBOSOFT.2018.8404920

Ziegler, J. G., and Nichols, N. B. (1942). Optimum settings for automatic controllers. *Trans. ASME.* 64. 759–768. doi:10.1115/1.2899060

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.

GLOSSARY

c_ϕ $\cos(\phi)$

s_ϕ $\sin(\phi)$

$\boldsymbol{A}$ Valve orifice

$c_\mathbf{f}$ Flow coefficient

$\boldsymbol{D}$ Chamber dampening

$d_\mathbf{E}$ Linear displacement for unactuated parts

d_i Chamber displacement to central axis

$d_\mathbf{n}$ Displacement to neutral axis

$F_\mathbf{p}$ Pressure induced force

F_σ Material stress induced force

$\boldsymbol{l}$ Chamber length

l_0 Actuator length in unactuated state

$l_\mathbf{a}$ Arc length

l_i Length of chamber i

$l_\mathbf{m}$ Measurement position

$\boldsymbol{M}$ Chamber mass

$\dot{m}$ Mass flow

$\boldsymbol{p}$ Pressure

$p_\mathbf{a}$ Actuator pressure

R Ideal gas constant

$\boldsymbol{R}_\mathbf{t}$ Friction resistance

$\boldsymbol{R}$ Rotation matrix

$r_\mathbf{c}$ Chamber radius

$\boldsymbol{S}$ Substance-specific temperature

$\mathbf{s}$ sine

T Temperature

$\boldsymbol{T}$ Transformation matrix

$\boldsymbol{u}$ Input voltage

$\dot{V}$ Volume flow

$\gamma_\mathbf{x}$ Sensor measurement around x axis

$\gamma_\mathbf{y}$ Sensor measurement around y axis

$\gamma_\mathbf{z}$ Sensor measurement around z axis

$\boldsymbol{\theta}$ Actuator bending angle

$\theta_\mathbf{m}$ Measured actuator bending angle

$\boldsymbol{\tau}$ Time constant

$\boldsymbol{\phi}$ Actuator bending direction

$\phi_\mathbf{m}$ Measured actuator bending direction

$(\mathbf{CF})_\mathbf{B}$ Base frame of testbed

$(\mathbf{CF})_0$ Base frame of segment

$(\mathbf{CF})_1$ Frame at end of PCC segment

$(\mathbf{CF})_\mathbf{E}$ End effector frame

$(\mathbf{CF})_\mathbf{C}$ Reference frame for IMU

$(\mathbf{CF})_\mathbf{IMU}$ Frame of IMU sensor

$(\mathbf{CF})_\mathbf{cam}$ Camera frame

FRA Fiber-reinforced actuator

IMU Inertial measurement unit

PCC Piecewise constant curvature

SMC Sliding mode control

SPA Soft pneumatic actuator

frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 01 April 2021
doi: 10.3389/frobt.2021.637301

Improved Continuum Joint Configuration Estimation Using a Linear Combination of Length Measurements and Optimization of Sensor Placement

Levi Rupert[1*†], *Timothy Duggan*[2†] *and Marc D. Killpack*[1]

[1] Robotics and Dynamics Lab, Department of Mechanical Engineering, Brigham Young University, Provo, UT, United States, [2] Otherlab Inc., San Francisco, CA, United States

OPEN ACCESS

Edited by:
Concepción A. Monje,
Universidad Carlos III de Madrid,
Spain

Reviewed by:
Kristin M. De Payrebrune,
University of Kaiserslautern, Germany
Deepak Trivedi,
General Electric, United States

***Correspondence:**
Levi Rupert
levi.rupert@gmail.com

[†] These authors have contributed equally to this work

Specialty section:
This article was submitted to Soft Robotics, a section of the journal Frontiers in Robotics and AI

Received: 03 December 2020
Accepted: 25 February 2021
Published: 01 April 2021

Citation:
Rupert L, Duggan T and Killpack MD (2021) Improved Continuum Joint Configuration Estimation Using a Linear Combination of Length Measurements and Optimization of Sensor Placement. Front. Robot. AI 8:637301. doi: 10.3389/frobt.2021.637301

This paper presents methods for placing length sensors on a soft continuum robot joint as well as a novel configuration estimation method that drastically minimizes configuration estimation error. The methods utilized for placing sensors along the length of the joint include a single joint length sensor, sensors lined end-to-end, sensors that overlap according to a heuristic, and sensors that are placed by an optimization that we describe in this paper. The methods of configuration estimation include directly relating sensor length to a segment of the joint's angle, using an equal weighting of overlapping sensors that cover a joint segment, and using a weighted linear combination of all sensors on the continuum joint. The weights for the linear combination method are determined using robust linear regression. Using a kinematic simulation we show that placing three or more overlapping sensors and estimating the configuration with a linear combination of sensors resulted in a median error of 0.026% of the max range of motion or less. This is over a 500 times improvement as compared to using a single sensor to estimate the joint configuration. This error was computed across 80 simulated robots of different lengths and ranges of motion. We also found that the fully optimized sensor placement performed only marginally better than the placement of sensors according to the heuristic. This suggests that the use of a linear combination of sensors, with weights found using linear regression is more important than the placement of the overlapping sensors. Further, using the heuristic significantly simplifies the application of these techniques when designing for hardware.

Keywords: estimation, optimization, continuum joints, soft robotics, proprioception

1. INTRODUCTION

Continuum joints are becoming a common style of robotic joint, especially in the world of soft robotics. These joints bend continuously along their length and offer the ability to form complicated shapes, operate in cluttered environments, and can be compliant which increases the inherent safety of the robot.

While being able to form complicated shapes and easily deform is one of continuum joints biggest strengths, it is also one of the attributes that make them the hardest to use in practice. Current approaches for sensing the configuration of continuum robots include many different methods such as motion capture, optical sensors, and length sensors (see section 1.1 for a review of many of the methods used for sensing continuum joint's state). Many of these state-of-the-art methods operate under assumptions that limit their ability to estimate the full kinematic position of a continuum joint in non-laboratory settings, (e.g., settings where the joint undergoes actual loads during a useful task that cause unanticipated bending). In this work we focus on methods that use measurements of the length of a continuum joint to estimate the configuration of the joint. Many of the previous methods in the literature assume that the bending of the joint is constant curvature. This assumption readily breaks down as soon as any actual loads are applied to the joint. Of particular note, is when the robot is in the s-shape bending as shown in **Figure 1** (see proximal joint). For any method using a single length measurement and a constant curvature assumption, the measurement in this scenario will result in an estimate of zero deflection.

The most accurate way to sense the full configuration of a continuum joint using length sensors, would be to divide the joint into infinitesimal segments and have each of those segments be monitored by length sensors. For every sensor that is added to the joint, the constant curvature assumption can be applied to that smaller segment. By having every segment covered by its own sensor the full configuration could be reconstructed by treating those segments as pieces of a kinematic chain.

This method is not feasible for a real system due to mechanical, electrical, and computational limits. However, it does suggest that it may be possible to increase the number of sensors to get more accurate pose estimation of a continuum joint while still remaining within the mechanical, electrical, and computational constraints of a real system.

However, if sensors are simply placed end to end along the joint, estimation will still be limited to the maximum number of sensors that will be allowed by the physical constraints of the system. Our hypothesis was that by using measurements from sensors that have overlapping coverage of the same discrete joint segments, whether literal discrete segments like the robot shown in **Figure 1**, or representative segments of a soft robot, a more accurate estimate can be accomplished. In this paper, we show that by overlapping sensors on the joint, more information can be gained without the cost of adding a sensor for every representative segment.

We propose and demonstrate two new methods of estimating the configuration of a continuum joint using measurements from overlapping length sensors. The first method averages the per segment length of the sensors for each sensor monitoring a segment, we call this the Equally-Weighted Averaging Method (E-WAM). The second method estimates the segment configuration by using a linear combination of the per segment lengths of all the sensors on the continuum joint, we call this the Weighted Averaging Method (WAM). The weights for the WAM method are found by performing a linear regression as discussed in section 2.5.

To determine the placement of the overlapping sensors, we developed a heuristic placement method as well as an Evolutionary Algorithm (EA) that determines optimal sensor placement for a joint. We compare the results of these two placement methods in this paper.

The primary contributions of this paper are:

1. The novel concept of overlapping length sensors to improve the estimation of a continuum joint's state.
2. WAM: A method for using overlapping sensors to significantly improve continuum joint estimation resulting in an estimate that reduces error by a factor of eleven when using two sensors on a joint rather than using a single sensor.
3. Two methods for determining the placement of overlapping length sensors on a continuum joint, and an objective comparison of their performance.

All of the methods and theory that we develop in this paper are based purely on kinematics and static loading conditions. We confirm and demonstrate our contributions using a Piece-Wise Constant Curvature continuum joint kinematic simulation. Future work would include implementing this on actual hardware and in dynamic environments.

The remainder of this paper is organized as follows, section 1.1 discusses related literature on sensor design and estimation for continuum actuators, as well as methods for determining optimal sensor placement. Section 2 discusses the assumptions we used and develops the models, theory, and algorithms for our estimation methods. Section 3 presents the results of the estimations methods and section 4 discusses the results and possible applications for future work.

1.1. Related Work

The focus of this paper is estimation for a discrete-segment continuum joint. Although there are many types of soft robot joints [including discrete segments (Hannan and Walker, 2003), compliant continuum joints with discrete rigid components (Rone and Ben-tzvi, 2013), and fully soft-bodied robots (Marchese et al., 2014)], we have chosen to develop our methods for discrete segments because (1) it matches our actual development hardware, and (2) most soft robot joints could be represented to varying degrees of fidelity by a discrete-segment model where the kinematics are approximated with a series of representative constant curvature segments, regardless of actual construction.

Related literature can be divided into two main areas: (1) soft robot configuration estimation; (2) soft robot sensor placement.

1.1.1. Soft Robot Configuration Estimation

Of the two areas covered in this paper, by far the most literature exists relative to novel sensors for soft robot configuration estimation. We therefore describe prior work that uses different methods of construction or physical phenomenon to estimate soft robot configuration. We also describe methods used to estimate the actual bend angle or pose.

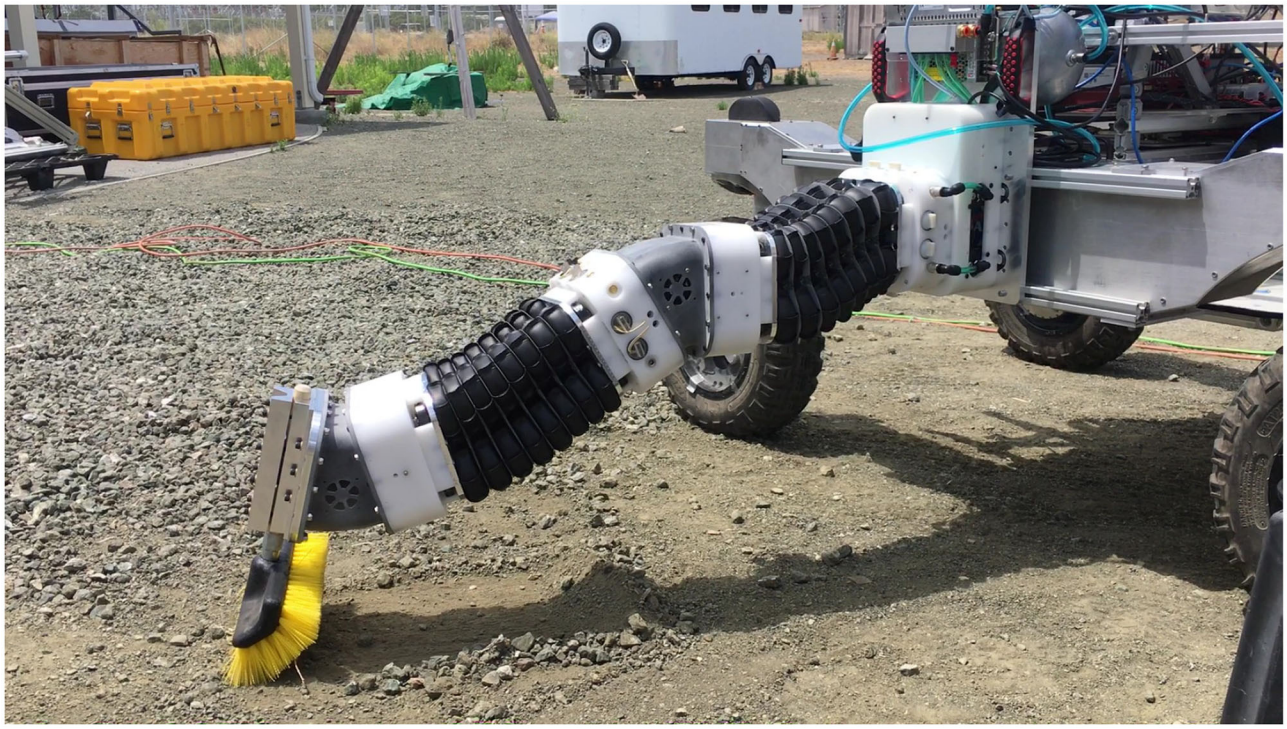


FIGURE 1 | A compliant continuum robot exhibiting non-constant curvature bending in its joints (especially in the first joint) when under load while performing a real-world task.

A significant amount of the research in soft robot configuration estimation has required using motion capture systems with infrared cameras and reflective tracking dots (Marchese et al., 2014; Katzschmann et al., 2019), electromagnetic field detectors (Song et al., 2015; Anderson et al., 2017; Gerboni et al., 2017), or virtual reality tracking hardware (Hyatt et al., 2019). However, using this type of sensor constrains the mobility of the soft robot to operate solely within the range of the motion capture system.

Resistance-based sensing is often used with conductive material or fabrics that are assembled in a way such that the resistance of a circuit varies as the bend angle of the robot changes. Examples use methods ranging from commercial flex sensors (Ozel et al., 2016), to conductive thread (Cianchetti et al., 2012; Zhao and Abbas, 2016; Abbas and Zhao, 2017), or yarn (Wurdemann et al., 2015), to conductive silicone that is cut using principles from kirigami (Truby et al., 2020). There are multiple examples of this approach (see Gibbs and Asada, 2005; She et al., 2015; Elgeneidy et al., 2016, 2018; Yuen et al., 2017; Zhou et al., 2020).

Many papers have focused on using optical methods that tend to revolve around novel combinations or topologies for Fiber Bragg Grating (FBG) sensors (see Wang et al., 2016; Zhuang et al., 2018; He et al., 2019; Sheng et al., 2019). However, other related methods focus on the basic idea of using optical fibers in general (see Yuan et al., 2011; Chen et al., 2019; Godaba et al., 2020). Using optical frequency domain reflectometry combined with added optical gratings the authors in Monet et al. (2020) were able to show that they could improve configuration estimation when in contact or with non-constant curvature for medical applications.

Some methods have relied on photo diodes (Dobrzynski et al., 2011), or combined the strength of traditional camera or ultrasound images in conjunction with optical fibers (see Denasi et al., 2018; Wang et al., 2020). Other researchers used camera-based methods directly to detect contact, or estimate deformation for a deformable link, but with rigid joints (Oliveira et al., 2020).

Other physical phenomenon used include capacitance (Yuen et al., 2017, 2018; Bilodeau et al., 2018; Case et al., 2018), inductance (Felt et al., 2016, 2018, 2019), magnetism (Ozel et al., 2016), impedance (Avery et al., 2019), or a combination of gyroscope, accelerometer, and magnetometer in an inertial measurement unit (IMU) package (Hyatt et al., 2019).

Similar to our efforts to include multiple sensors to improve configuration estimation, there are some researchers who have used overlapping sensors to improve performance. Specifically, Li et al. (2020) used a dual array FBG scheme to improve estimation accuracy. While Felt et al. (2019) used two circuits and measured change in inductance to improve estimation of lateral motion for a continuum joint.

As near as we can tell, all of the previous sensors and estimation methods (minus those that give a global pose such as motion capture) seem to focus on estimating curvature *or* linear motion only, which does not account for deformation that

we would expect when these platforms are heavily loaded. Some methods enable detection of contact, but this is used as a way to relate discrepancies in curvature to a contact event, rather than using the loading condition to more accurately estimate the joint configuration with a non-constant curvature assumption. However, there is some literature where the configuration of flexible members experiencing a point load is estimated using accurate Kirchoff or Cosserat rod models and additional sensor information (such as cameras or force-torque data). In Rucker and Webster (2011) they use an Extended Kalman Filter in conjunction with a Cosserat rod model which requires a measurement of the tip pose and applied forces. While in Borum et al. (2014) the authors use external cameras to help solve for the configuration of a flexible member that can have multiple equilibrium positions (due to bifurcation) by formulating the problem as a geometric optimal control solution. This solution includes estimates for the forces and torques applied at the tip to cause the deformation. In both cases, the deformation was restricted to being planar and was caused by an external force at the tip, rather than being included as part of a potentially self-contained soft robot control scheme.

In Trivedi and Rahn (2009) and Trivedi and Rahn (2014), the authors solve for the configuration of the OctArm robot platform with unknown payloads using Cosserat rod models and three different sensing methods (e.g., force-torque sensors and an inclinometer at the base, multiple cable encoders, and multiple inclinometers along the manipulator) to constrain and solve initial value or ODE problems with given boundary conditions. The method was effective, but required varying levels of accurate knowledge about soft robot parameters depending on the sensing method used and was again restricted to planar applications (although not due to limits in formulation). In addition, this formulation would require additional sensors across the arm if a distributed load were applied (not at the tip or end effector). Similar work uses Cosserat rod models (Sadati et al., 2020) or Kirchoff elastic rod models (Takano et al., 2017; Nakagawa and Mochiyama, 2018) combined with force sensing at the base of the flexible member in order to estimate soft robot configuration or interaction forces and stiffnesses.

These model-based methods hold great promise and could likely be incorporated with our model-free method. However, additional benefits of our method are that even without a complex soft-body model, it performs quite well and is able to handle loading conditions that are not limited to the tip of the flexible member. Any additional information derived from an accurate model within an estimation scheme such as a Kalman filter would likely improve the results shown in this paper.

Finally, using different modalities, many researchers have used neural networks to map sensor output to joint configuration for optical sensors (Sklar et al., 2016), FBG sensors plus ultrasound images (Denasi et al., 2018), pressure readings (You et al., 2017), tactile arrays (Scimeca et al., 2019), or linear potentiometers (Melingui et al., 2014; Merzouki et al., 2014; Day, 2018). In Lun et al. (2019) they develop a flexible sensor using fiber Bragg gratings that when combined with a learned model can be used to accurately reconstruct the surface of a soft robot, but this is not applied specifically to a soft robot. Many of these methods learned a mapping to estimate full pose for the tip of one, or sometimes multiple joints. However, one of the main limitations is that there is no relation or intuition between the data and the black box model that is produced. Also, if the manipulator were to carry a larger load, additional data with the load in place would likely need to be collected, especially if the joint deformed in a way that violated constant curvature assumptions. Information about the load (e.g., overall mass and distribution of mass) may also have to be included in the training data to make the approach general. Because our approach is based on fitting parameters to shapes that are caused by many different loading conditions, we expect this approach to potentially generalize more easily.

1.1.2. Sensor Placement

The general problem of sensor placement (number of sensors and relative positioning) is often approached using a metric of observability in order to improve estimator design (see Krener and Ide, 2009; DeVries and Paley, 2013; Qi et al., 2015). However, observability may not always be the best metric and sensor placement based on simple models and heuristics is an open research problem (Clark et al., 2020).

For our specific contributions, we focus on sensor placement in the context of soft robot configuration estimation. Some researchers have followed the previously mentioned approach of relating soft robot sensor placement to observability (Mahoney et al., 2016). In this case they use a differential representation of the continuum robot's kinematic equations. However, the robot is a concentric tube robot which appears to be unloaded, in contrast to the work we present. In Tapia et al. (2020), they require hyperelastic material models and finite element discretizations to simulate nonlinear behavior of a given soft robot with expected loading. This is similar to our method with two main differences. Our loading and deformation models are much simpler and the proposed optimization in that paper requires the sensors to be integrated with the actual fabrication of the soft robot, unlike ours which can be added after the fact and only needs to measure length. Other relevant work includes Deutschmann et al. (2019), where the authors optimize the attachment points for length sensors to estimate the pose of a 6-DoF continuum-joint robot head. This required a beam finite element model with a fixed load (the robot head) and the data was fused with IMU. Finally, in Kim et al. (2014) they use FBG sensors and an optimization with a similar notion to our weighted reconstruction, using their own set of basis functions. However, the type of optimization presented does not necessarily translate to overlapping sensors (which we have found to be very beneficial in the results presented in this paper).

2. MATERIALS AND METHODS

In this section we describe the methods used to develop the simulated continuum robot configuration, estimate the continuum robot configuration from the attached sensors, and develop the evolutionary algorithm used to find the optimal sensor placement along the continuum robot joint.

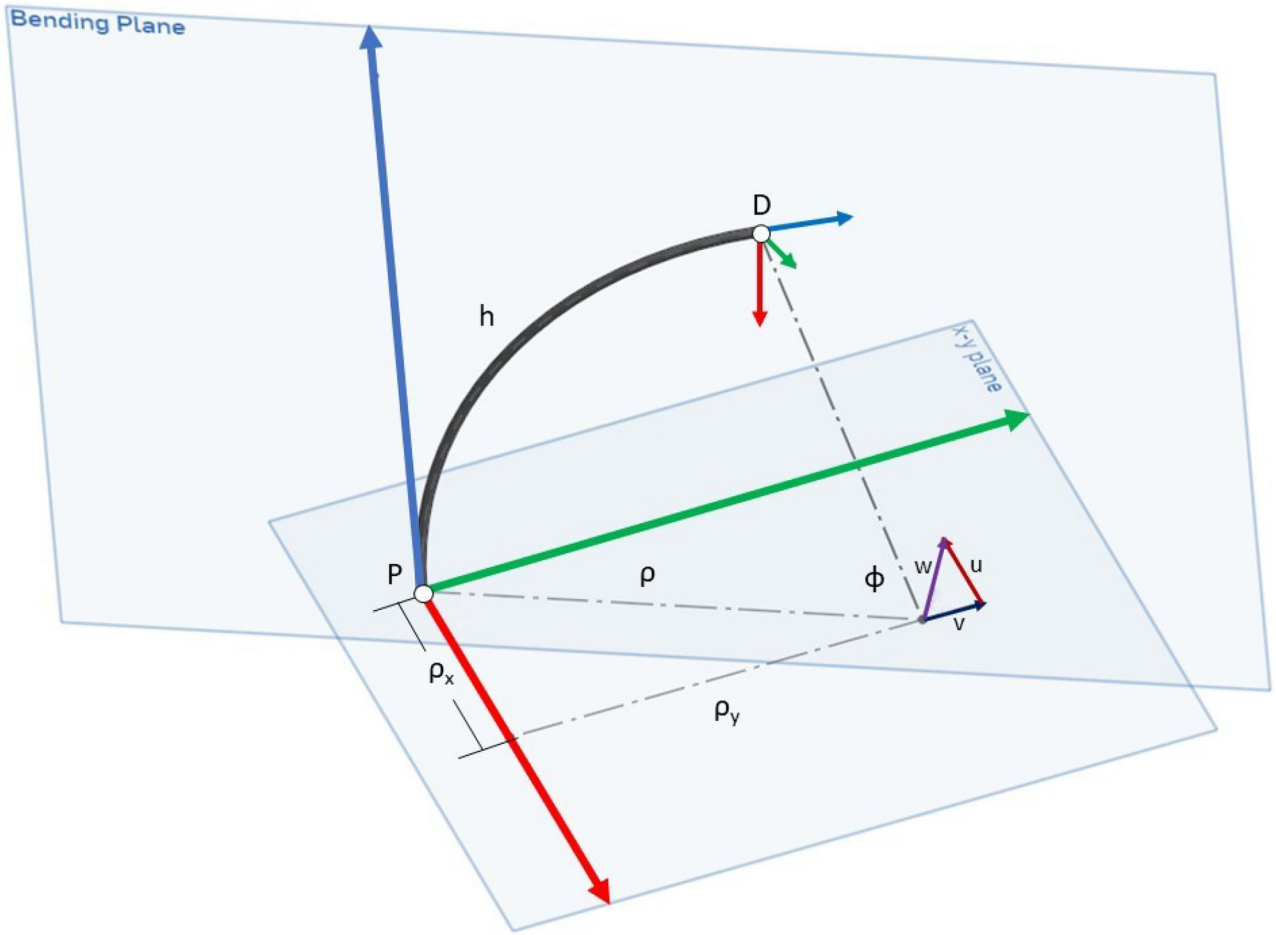


FIGURE 2 | The parametrization variables used to describe a constant curvature continuum segment in our work. Developed by Allen et al. (2020).

2.1. Continuum Joint Configuration

For a general continuum joint there are three degrees of freedom, bending about two orthogonal axes and extension along an axis orthogonal to the two bending axes. As long as one bending degree of freedom exists, in which the center of rotation stays constant for the bending range, the joint can be considered a continuum joint.

In our work, we focus on continuum joints that have a fixed length/height and two bending degrees of freedom. The continuum joint hardware shown in **Figure 1**, is used as the basis for models in this paper, is made up of bending segments of a fixed height. We assume that these segments bend with piece-wise constant curvature. The theory is that the curvature change in one segment is small enough that it can be assumed to have constant curvature. It should be noted that the methods discussed in this paper can be adapted for joints that are not actually made of smaller constant curvature segments by splitting the joint into virtual segments.

Due to our fixed length assumption as the joint bends there exists a neutral axis or spine at the center of the joint that does not change lengths. This is represented by the black line in **Figure 2**.

We use the u, v, and h states developed by Allen et al. (2020) to describe the configuration space and pose of a single continuum joint segment under the constant curvature assumption (displayed graphically in **Figure 2**). This parameterization is based on Screw Theory. The full configuration of the continuum joint is described by the u, v, and h parameters of the series of smaller segments that make up the joint. The parameter u describes bending about the local x-axis and v describes bending about the local y-axis for each segment. The variable h is the length of the neutral axis which we keep constant for the purposes of this paper. According to Allen et al., the arc angle, ϕ, is equal to the magnitude of the rotation axis, $w = [u, v, 0]$.

2.2. Sensor Arrangement

As stated previously, for this application we are simulating sensors that measure the change in length of the joint as it undergoes bending. By using constant curvature assumptions we can calculate the pose from the sensors length measurements as will be described in section 2.3.

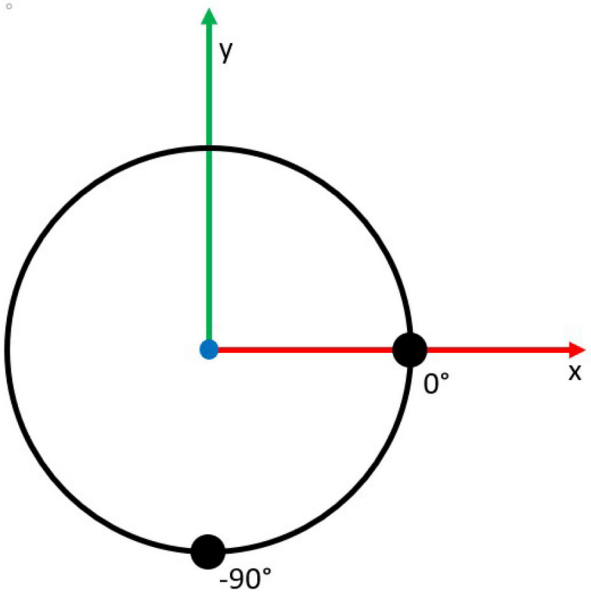


FIGURE 3 | A cross-sectional illustration of the sensor locations at 0° and −90° on the circumference of the joint.

The continuum joint has two sets of sensors that run the length of the joint and start at the base at 0 and −90° on the circumference of the joint as shown in **Figure 3**. These locations allow each set of sensors to independently measure the two degrees of freedom (u and v) as the mounting points correspond to the directions of the bending axis. Thus, bending about each axis will only be measured by a single set of sensors.

Each set of sensors in our simulations contain between one and six sensors which are aligned such that they are parallel to the neutral axis of the continuum joint in the unbent configuration. Additionally we modeled each joint using 12–48 segments of equal length. The sensors were placed such that they cover a series of consecutive segments. This series can be a minimum of one segment or a maximum of the total number of segments representing the joint. An example of a set of three sensors is shown in **Figure 4**.

For a sensor configuration to be considered for our simulated experiments, each segment must be covered by a minimum of one sensor. For the joints simulated in this paper, both degrees of freedom had identical sensor placements although this is not a requirement for successful configuration estimation of a two degree of freedom continuum joint.

2.3. Pose Estimation

As mentioned in sections 2.1 and 2.2, the bending section of the continuum joint is divided into smaller segments that are small enough that we can assume constant curvature. Additionally each of these segments is covered by at least one length sensor located at a fixed distance away from the neutral axis.

The work developed by Allen et al. (2020) also describes how to estimate the angle of bending for a continuum joint with constant curvature that is monitored by a length sensor. We apply this method to our discrete sections by using Equations (1) and (2) which convert the length of a tendon, l, located at a fixed radius from the neutral axis of the joint to a joint angle, u or v, given the height of the segment, h.

$$u = \frac{l_{tendon\ at\ 0^\circ} - h}{radius} \tag{1}$$

$$v = \frac{-h + l_{tendon\ at\ -90^\circ}}{radius} \tag{2}$$

As defined in Allen et al. (2020) w is defined as $[u, v, 0]$ and whose magnitude equals ϕ. Therefore, ϕ represents the total magnitude of the deflection angle as shown in Equation (3)

$$\phi = \sqrt{u^2 + v^2} \tag{3}$$

The full homogeneous transformation matrix for the uvh parametrization is described in Allen et al. (2020). We use this to compute the position of the end of each link along the kinematic chain of segments that makes up the complete pose of the bending section of the continuum joint.

This approach is used for each estimation method described in section 2.5. Although each sensor covers several constant curvature segments, these segments may not have the same curvature. Thus at least some error is introduced. The tendon length l of a segment is calculated by dividing the full sensor by the number of segments that it covers. This tendon length is referred to as a "virtual tendon length."

Given every segment's angle of deflection the length of a simulated sensor is calculated by summing the "virtual tendon lengths" for each segment that the sensor covers. The "virtual tendon lengths" are calculated by solving for the respective l found in Equations (1) and (2).

2.4. Loading Conditions

Since the motivation of this paper is to improve the estimation of continuum joint poses under real-world loading conditions, we examine four loading conditions that encapsulate the majority of situations experienced by a cantilevered continuum joint with a fixed mounting. The loading conditions are listed as follows:

- **End Force Load**: This loading condition simulates contact at the end of the joint (**Figure 5A**).
- **Uniformly Distributed Load**: This loading condition simulates joint deflection due to gravity (**Figure 5B**).
- **End Force Load with a Moment**: This loading condition simulates a load at the end of the joint with a torque created by the joints actuators resulting in an s shape (**Figure 5C**).
- **Constant Curvature**: While not explicitly a loading condition, this represents the joint being actuated such that all segments reach their maximum range of motion (ROM).

We treat the continuum joint as a cantilevered beam and apply each of the given loading conditions. The Finite Element Analysis (FEA) program ANSYS was used to simulate the resulting deformation of the modeled beam.

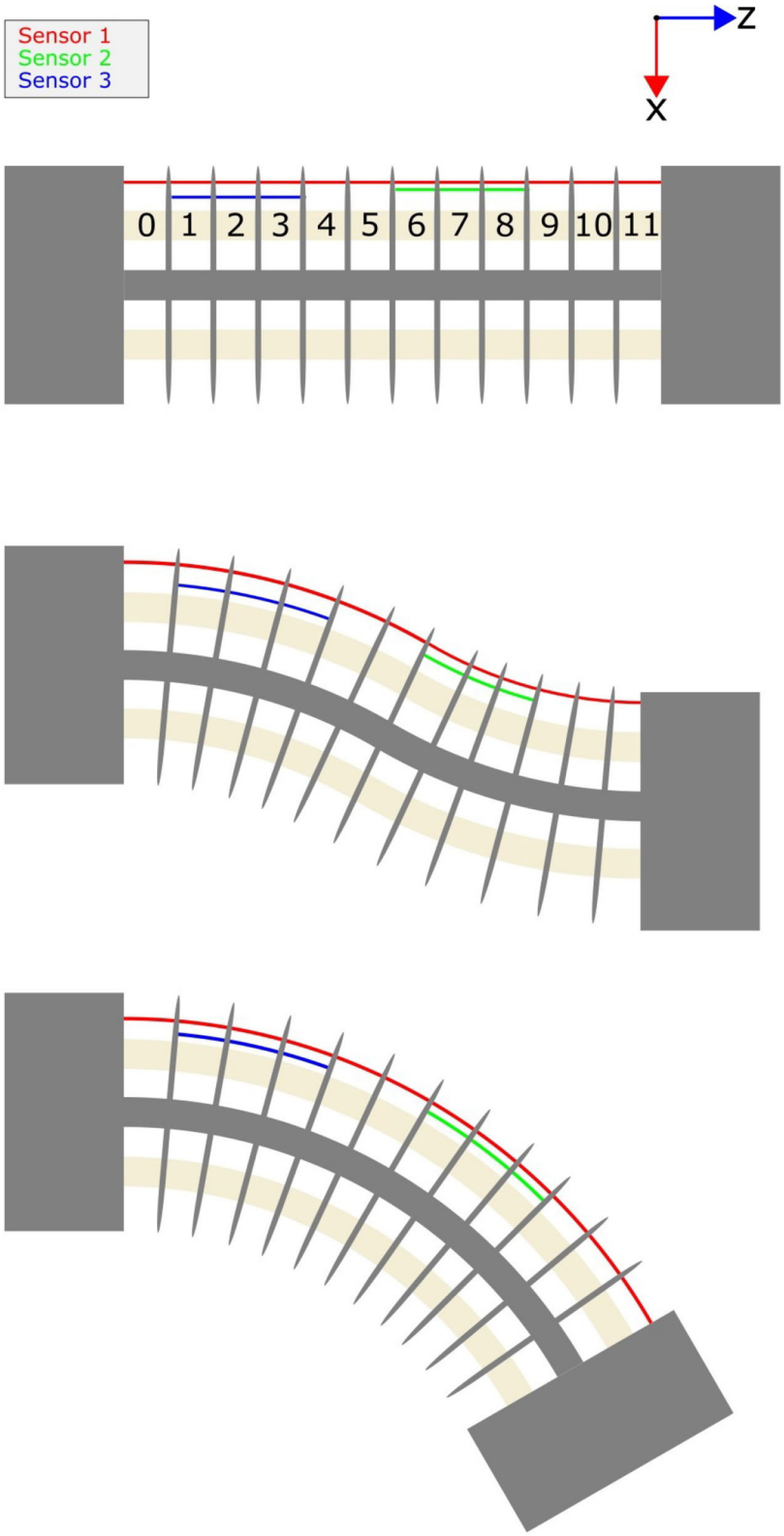


FIGURE 4 | An illustration of showing three possible bending states of a continuum joint, the segment indexing used in this paper, and a possible sensor configuration.

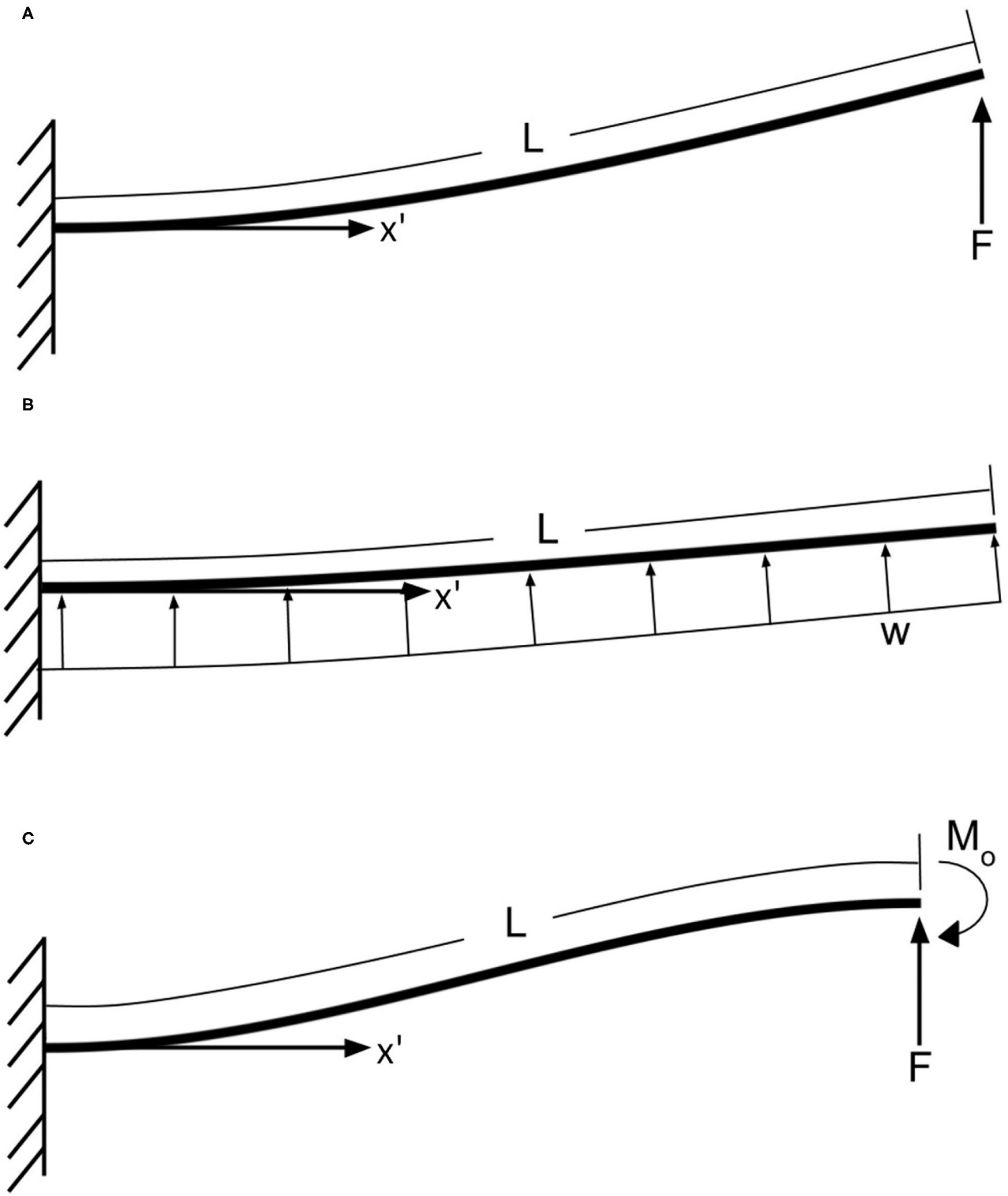


FIGURE 5 | (A) End force load, **(B)** uniformly distributed load, **(C)** beam with one end fixed and the other end guided. *L* is the arc length of the full joint, *F* is an end load amount, *w* is a distributed load, and M_0 is a moment.

We use Beam 188 elements which were divided into $m * 3$ elements, where m is the number of the constant curvature segments in the joint. For each loading condition the load and/or moment was incrementally increased until the desired deflection of the first segment was reached.

It is important to note that we used the FEA solution for all of the loading except for the constant curvature case as the angles of deflection for each segment are all the same and thus known.

We then adapt these nonconstant curvature simulations to our actual Piecewise Constant Curvature (PCC) joint model. This is done by recording the total deflections from the FEA model at the beginning and end of each segment. Then the difference between the deflection at the beginning of a segment and the end of the segment is calculated. This difference is then set as the bending angle for that Constant Curvature segment as shown in Equation (4).

$$\phi_i = \theta_{i+1} - \theta_i \; \forall \; i = 1, \ldots, m \tag{4}$$

Figure 6 demonstrates a continuum joint undergoing a force load and the deflection angles, θ_i that can be used to calculate the relevant joint angles ϕ_i

Due to the computational demands of the finite element analysis (FEA), we found solutions for a specific set of joint deflections. This was performed for every loading condition where the first segment was set to the maximum ROM which was incremented from -8 to $+8^\circ$ in increments of $\frac{1}{8}^\circ$. For any modeled deflections that were between the original FEA solutions a linear interpolation was used. Using this method, a maximum per segment error bound of 0.0285° was calculated for the linear interpolation. This was calculated using the worst case scenario (maximum difference between two points used for interpolation) in terms of error. Specifically, the error bound was found by summing the difference between joint angle FEA solutions used for interpolation and then dividing by the number of segments. We did not need to use the interpolation method for the constant curvature loading case as each segment's angle would be the maximum ROM.

2.5. Estimation Methods

In this section we describe the estimation methods used in the simulated experiments. We experimented with simulating four different sensor configurations for gathering state data and used two different methods for state estimation.

2.5.1. Sensor Configurations

When describing the configuration of sensors along the length of a continuum joint, we use a pair of two numbers inside square brackets to represent the sensor's starting segment and ending segment as such [starting segment–ending segment]. **Figure 4** shows the indexing of the joint segments on this 12 segment continuum joint. The segment numbering is started at the most proximal segment which is labeled segment 0 and the rest of the segments are incrementally labeled until the last segment. Using our method of describing a sensor configuration on a joint, the red sensor is [0–11], the blue sensor is [1–3], and the green sensor is [6–8].

2.5.1.1. Single Sensor

The Single Sensor configuration, henceforth abbreviated as SS, involves using a single sensor that spans the entire length of a joint, to measure the overall joint angle of a continuum joint. This method relies on the assumption of constant curvature along the entire length of the joint for state estimation. This method represents the bare minimum amount of sensing that a continuum joint can have for state estimation with length sensors.

2.5.1.2. End-to-End

The End-to-End Sensor configuration, henceforth abbreviated as EE, involves multiple sensors that are placed along the length of the joint with every segment covered and no overlap. The method for algorithmically determining the sensor placement involves dividing the number of segments by the number of sensors and rounding down. That is the default number of segments each sensor will cover. If there is a remainder from dividing the number of segments by the number of sensors, that remainder is evenly distributed among the sensors closest to the distal end of the joint. For example, a 12 segment joint with five sensors would have sensors that cover the following segments [0–1], [2–3], [4–5], [6–8], and [9–11].

2.5.1.3. Heuristic Overlap

The Heuristic Overlap configuration, henceforth abbreviated as HO, involves multiple sensors aligned in a regular pattern along the joint, with each sensor overlapping with its neighboring sensors for two segments. The sensor placement is determined by first finding the EE sensor configuration and expanding each sensor's starting and ending index by one segment. Note, sensors that cover the first or last segment on the joint are not expanded past the ends of the joint. For example, a 12 segment joint with five sensors would have sensors that cover the following segments [0–2], [1–4], [3–6], [5–9], and [8–11].

2.5.1.4. Optimized Overlap

The Optimized Overlap configuration, henceforth abbreviated as OO, is a sensor configuration that is determined by an evolutionary algorithm we developed. The evolutionary algorithm is described in section 2.6. This configuration represents the best possible sensor configuration for a given number of sensors.

2.5.2. State Estimation Methods

For each estimation method, we estimate the angle of deflection for each individual segment of the continuum joint which then allows for the estimation of the full configuration of the joint. This is accomplished by estimating the length that a single sensor would be if it was monitoring just that individual segment, henceforth known as the"virtual sensor length." We have developed three methods for performing this estimation.

2.5.2.1. DEM (Direct Estimation Method)

For sensor configurations that have no overlap, we estimate the virtual sensor length of a segment by simply dividing the length of the sensor covering it by the number of segments that sensor

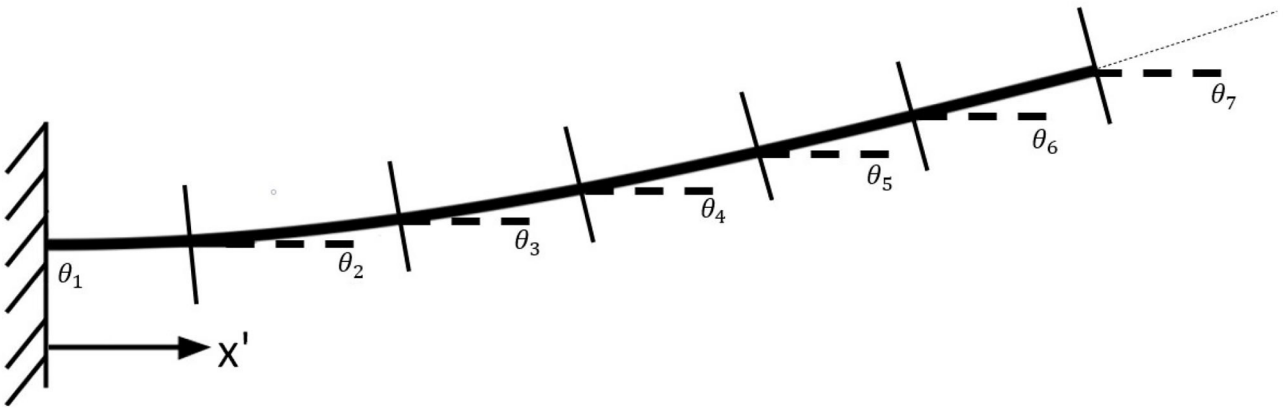


FIGURE 6 | Illustration of the method used for adapting a continuous bending model to a Piece-wise Constant Curvature model.

covers. This method assumes that all of the segments covered by a sensor have the same angle of deflection.

2.5.2.2. E-WAM (Equally-Weighted Averaging Method)

For sensor configurations that have overlap, we have two methods of estimating the virtual sensor length of a joint segment, the first of which we call E-WAM. The E-WAM method takes the per segment lengths of all the sensors covering a segment and averages them to estimate the length of the virtual sensor for that segment.

2.5.2.3. WAM (Weighted Averaging Method)

The second method for estimating the virtual sensor length, l_{est}, of a joint segment on a robot that has overlapping sensors is WAM. This method uses a weighted linear combination of all of the sensors on the robot to find the virtual sensor length for each segment, the hypothesis being that the sensors that do not cover the segment still provide additional information about its state. Each segment has a separate weight for each sensor on the joint as shown in Equation (5).

$$l_{est,i} = \sum_{j=0}^{n} w_{i,j} * \frac{l_j}{p_j} \tag{5}$$

where i is the *ith* segment, n is the number of sensors on a joint, $w_{i,j}$ is the weighting on the *ith* segment for the *jth* sensor, l_j is the full length of the *jth* sensor, and p_j is the number of segments the *jth* sensor spans.

We find these weights by applying the robust linear regression algorithm from Scipy (Virtanen et al., 2020) to deflection angle data we simulated from the continuum joint under 30 different loading samples (s) for each of the 4 loading conditions (c) for a total of 120 data points per joint segment. The 30 different loading samples are calculated by varying the ROM used in the loading conditions as describe in the following equation.

$$ROM_{loading,i} = -ROM_{max} + i\frac{2ROM_{max}}{s} \;\forall\; i = 1, \ldots, s \tag{6}$$

We use the scipy *least_squares* function with the loss condition set to "*soft_l1*" and the "*f_scale*" condition set to 0.1. Our residuals function can be seen in Equation (7) where S is the matrix of collected sensor data, w_i is a vector of the weights for which we are solving, and l_i is the length of a virtual sensor covering that segment.

$$residual = Sw - l_i \tag{7}$$

Matrix S takes the form shown in Equation (8). Each row is made up of the sensor values from one of the simulated loading cases. The sensor data in the matrix is normalized and denoted as $\bar{s}$, where $\bar{s}_j = \frac{l_j}{p_j}$.

$$S = \begin{bmatrix} \bar{s}_{1,case\,1} & \bar{s}_{2,case\,1} & \cdots & \bar{s}_{n,case\,1} \\ \bar{s}_{1,case\,2} & \bar{s}_{2,case\,2} & \cdots & \bar{s}_{n,case\,2} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{s}_{1,case\,m} & \bar{s}_{2,case\,m} & \cdots & \bar{s}_{n,case\,m} \end{bmatrix} \tag{8}$$

Vector w_i takes the form shown in Equation (9).

$$w_i = \begin{bmatrix} w_{i,sensor\,1} \\ w_{i,sensor\,2} \\ \vdots \\ w_{i,sensor\,n} \end{bmatrix} \tag{9}$$

Vector l_i takes the form seen in Equation (10).

$$l_i = \begin{bmatrix} l_{i,case\,1} \\ l_{i,case\,2} \\ \vdots \\ l_{i,case\,m} \end{bmatrix} \tag{10}$$

2.6. Evolutionary Algorithm

To find the optimal sensor placement on a continuum joint, we implemented an Evolutionary Algorithm (EA) from the DEAP (Distributed Evolutionary Algorithms in Python) Library (Fortin et al., 2012).

The goal of our algorithm is to find the optimized sensor placement for a given continuum joint with a fixed number of sensors. Prior to running the EA, we define the continuum joint on which we will be optimizing the sensor placement by setting the total length of the joint, the number of segments, and the total range of motion of the continuum joint.

The EA itself is the eaSimple function from the DEAP library, which handles iterating over the specified number of generations, selection, mating and mutation with built-in options or the ability to define your own functions. We chose to do 10 generations and discuss our choices for selection, mating, and mutation in section 2.6.4.

2.6.1. Defining an Individual

To represent an individual we used a list of integers with a length of two times the number of sensors. For example, a continuum joint with 12 segments and two sensors could be represented as [0, 7, 4, 11]. In this list, each sensor is represented by a pair of numbers. The first two numbers represent the starting segment index of the first sensor and the ending segment index of the first sensor. The second two numbers represent the starting and ending index of the second sensor. For a given sensor number i, the starting index is $2i$ and the ending index is $2i+1$. If the ending index is lower than the starting index, they are automatically swapped to be in the correct order by our algorithm. The sensors cover the full segments of both the starting and ending segment. In other words the sensor starts on the bottom of the starting segment and ends at the top of the end segment.

2.6.2. Creating the Population

To create the population, we create 500 individuals each with an attribute list that is two times the number of sensors long with random integers generated at every index of the attribute list.

We experimented with seeding the population with individuals that have sensors lined up end to end or start with a Heuristic overlap but found no noticeable improvement in the EA's performance.

2.6.3. Evaluating the Fitness

To evaluate the fitness of an individual we use a cost function that sums the deflection angle error of all m joint segments, for all s loading samples of a loading conditions, for all c loading conditions giving us the cost function seen in Equation (11). Our goal is to minimize the cost of an individual.

$$cost = \sum_{i}^{c}\sum_{j}^{s}\sum_{k}^{m}(|\phi_{actual,i,j,k} - \phi_{estimated,i,j,k}|) \qquad (11)$$

Additionally, when evaluating an individual, we first determine whether or not a sensor configuration is a valid configuration. For our purposes, valid means that each segment on the joint is observable i.e., covered by at least one sensor. If this criteria is not met, the individual's fitness score is set to the maximum which is the maximum joint error possible [$(2 \times ROM)$ multiplied by c, s, and m]. Intuitively, this means that the estimation was off by the maximum possible amount for each segment in each loading simulated.

We also experimented with including the Cartesian position and orientation of the end effector of the joint in the fitness score. However, due to its direct correlation with the individual deflection angles we found that this did not improve overall performance for the optimization.

2.6.4. Selection, Mating, and Mutation

Selection is performed though a tournament selection process as provided by the DEAP library, *deap.tools.selTournament(individuals, k, tournsize, fit_attr="fitness")*, where the method is passed a list of individuals (individuals) and the size of the tournament (tournsize).

The mating is performed by using a one point crossover algorithm provided by the DEAP library, *deap.tools.cxOnePoint(ind1, ind2)*, where "*ind1*" and "*ind2*" are two individuals that are to be mated. The algorithm randomly chooses a place for crossover to happen. Crossover then occurs by swapping the elements between the two individuals that are right of the selected element. This method cannot choose the last element so there will always be some crossover. We set the crossover probability to 0.7.

Mutation occurs using the method *deap.tools.mutUniformInt(individual, low, up, indpb)* found in the DEAP library where "*individual*" is the individual to be mutated, "*low*" and "*up*" are the lower and upper bound, respectively, that an attribute can be set to, and "*indpb*" is the independent probability that each element of the attribute will be mutated. Therefore, if an individual is selected for mutation each element of the individual's attribute (the sensor list) has a chance to randomly mutate to a value in the closed set ["*low*", "*up*"] based on a uniform distribution. We set the mutation probability to 0.5

2.7. Experiments

We had four hypotheses that we tested and analyzed for general trends.

1. Increasing the number of sensors on a joint for a given placement method and estimation method will improve the accuracy of the state estimation.
2. Overlapping the sensors can provide more information about the configuration of the joint and will therefore improve configuration estimation for continuum joints.
3. Using a weighted linear combination of the overlapping sensor data can decrease the state variable estimation as compared to an equally weighted linear combination. Additionally, the weights can be found using linear regression.
4. An evolutionary algorithm can be used to determine the optimal placement of overlapping sensors that will further improve state estimation for continuum joints.

To prove generality of our solutions and to test the hypotheses being proposed, we generated 80 different joints by varying the number of segments and the max ROM per segment. We varied the two variables as shown in **Table 1** to generate the 80 different joints. From here on in this paper, when we mention ROM, we are referring to the range of motion of the segment, not of the whole joint, unless explicitly stated otherwise.

For each of the hypotheses presented above, we perform simulated experiments that compare the performance of a sensor placement or state estimation method on all 80 joints. We compare the performance of the methods by simulating the joints in 40 different poses and comparing the aggregate error of the joint segments angle error (our cost function) normalized for number of segments m, max ROM, loading conditions c, and loading samples, s. The 40 different poses come from the 4 different loading cases ($c = 4$) and ten sample poses ($s = 10$). We then multiply by 100 to get the average joint segment angle error as a percent of ROM for a given joint (see Equation 12).

$$Average\ \%\ Error\ of\ ROM = \frac{cost}{c \cdot s \cdot m \cdot ROM_{max}} \tag{12}$$

With the exception of the base case of a single sensor, we performed all of our simulated experiments with two, three, four, five, and six sensors to study how the results change as more sensors are added. For the first hypothesis, we compare the average percent error of ROM when using the DEM on the simulated joints for the SS placement and two to six sensors in the EE placement. For the second hypothesis we study the effects of overlap by comparing the results of the EE placement method with the DEM estimation method vs. the HO placement method with the E-WAM estimation method. The third hypothesis tests

TABLE 1 | The different continuum joint parameters and their values that were simulated.

Parameters	Values
Number of segments	12, 16, 20, 24, 28, 32, 36, 40, 44, 48
Max ROM per segment (degrees)	±1, ±2, ±3, ±4, ±5, ±6, ±7, ±8

TABLE 2 | Average segment error as a percent of the range of motion, normalized over all of the deflection modes used for evaluation of performance.

	# Sensors	1	2	3	4	5	6
SS, DEM	Median	16.71	–	–	–	–	–
	3rd Quart.	16.92	–	–	–	–	–
	1st Quart.	16.16	–	–	–	–	–
EE, DEM	Median	–	8.309	5.423	4.033	3.178	2.634
	3rd Quart.	–	8.321	5.465	4.077	3.212	2.672
	1st Quart.	–	8.281	5.305	3.931	3.105	2.592
HO, E-WAM	Median	–	7.561	4.504	3.090	2.342	1.914
	3rd Quart.	–	7.735	4.720	3.230	2.538	1.993
	1st Quart.	–	7.277	4.280	2.878	2.257	1.784
HO, WAM	Median	–	1.552	0.262	0.0915	0.0434	0.0366
	3rd Quart.	–	2.086	0.364	0.112	0.0538	0.0475
	1st Quart.	–	1.403	0.181	0.0653	0.0289	0.0228
OO, WAM	Median	–	1.542	0.238	0.0836	0.0368	0.0315
	3rd Quart.	–	2.063	0.362	0.106	0.0506	0.0437
	1st Quart.	–	1.389	0.175	0.0427	0.0220	0.0215

SS, single sensor; EE, end to end; HO, heuristic overlapping; OO, optimized overlapping; DEM, direct estimation method; E-WAM, non-weighted averaging method; WAM, weighted averaging method.

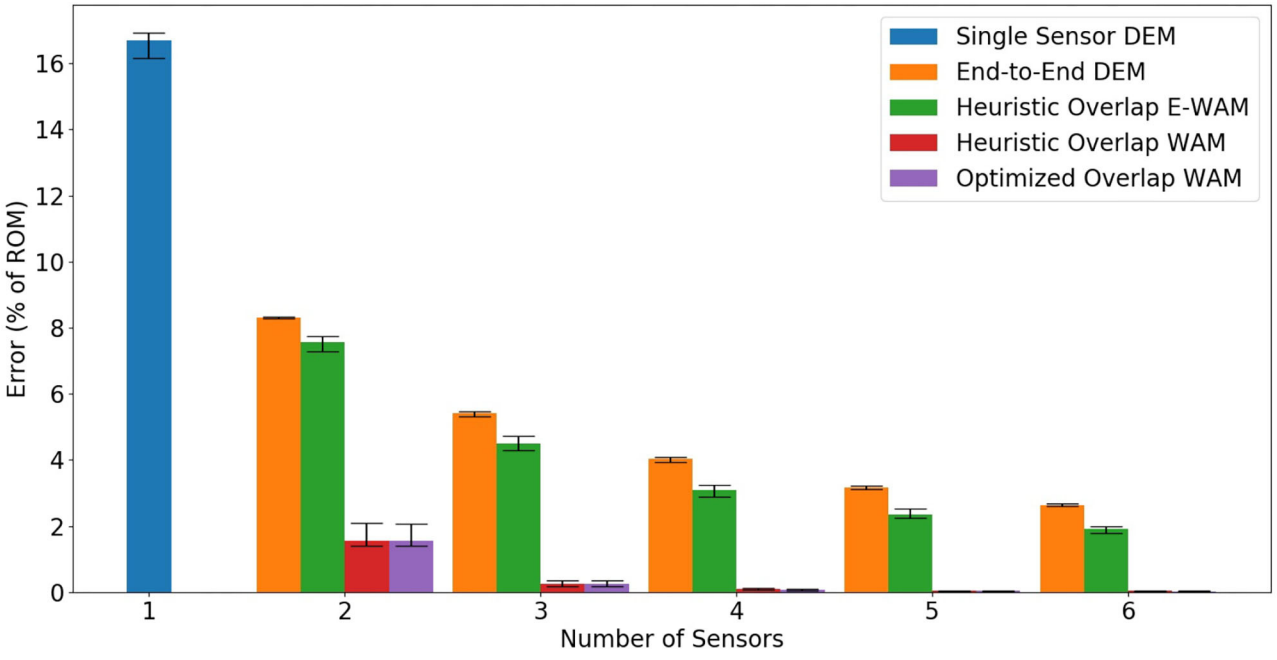


FIGURE 7 | This bar graph shows the median results for all of the simulated joints' "Average % Error of ROM" (defined in Equation 12) of all the joint's segments. Quartile bars are included to show the spread of the results.

our WAM method by comparing the WAM method and the E-WAM method on the joints with HO sensor placement. The final hypothesis tests our evolutionary algorithm by comparing the HO and OO sensor placement methods while using the WAM estimation method. Section 3 presents and discusses the results of the simulated experiments.

3. RESULTS

This section reports the results of the tests described in section 2.7. To make it easier to compare all of our results, we created a bar graph that summarizes the tested sensor placement and estimation methods, shown in **Figure 7**. We also report the original data used to generate the bar graph in **Table 2**.

We have also included a case study to help visually show the effectiveness of the different estimation methods for the different loading cases. **Figure 8** shows how well different estimation methods are able to reconstruct the actual configuration of the joint under the three real-world loading scases.

4. DISCUSSION

The results of these simulated experiments strongly support our first, second, and third hypotheses. The results also loosely support our fourth hypothesis. For all of the analyses listed in this discussion, the data behind **Figure 7** can be found in **Table 2**.

Our first hypothesis, that increasing the number of sensors on a joint for a given placement and estimation method will improve the accuracy of the state estimation, is somewhat intuitive. As we can see in **Figure 7**, as the number of sensors increases, the error, as a percent of the range of motion, generally decreases. This hypothesis is most strongly supported by the EE with DEM and HO with E-WAM state estimation methods. For these methods, the decrease in error resembles an exponential decay. While six sensors was the maximum number of sensors we used in our experiments, we expect the decrease in error for a given continuum joint to plateau when the number of sensors is greater than or equal to the number of segments. For example, a twelve segment joint using EE and DEM with 13 sensors would not be any more accurate than a joint with twelve sensors given our assumptions. In the real world, with imperfect sensors, this may not be true because having two sensors monitoring a single segment may allow filtering or averaging to get more accurate information out of the two sensors than a single sensor alone.

While the trend of increasing the number of sensors leading to a decrease in error is consistent across all the simulated experiments, we also noticed a steep decline in error for HO and OO using the WAM method when going from two sensors to three sensors, with an effective plateau in performance from four to six sensors. We attribute this plateau to the effectiveness of the WAM method to accurately estimate the state with a smaller number of sensors. Four or more sensors seems to add redundant information to the estimation method resulting in only minor decreases in error.

Our second hypothesis states that by overlapping length sensors on a continuum joint, we are able to obtain more information about its configuration and therefore better estimate said configuration. Referencing **Figure 7** again, we can see that all cases of HO or OO had lower errors than the EE placement method for a given number of sensors. This confirms that overlapping sensors does indeed allow us to more accurately estimate the configuration of the joint.

We first analyze why there is an improvement from using EE with DEM to HO with E-WAM. This is performed using the term "region of estimation," which refers to groups of segments on the continuum joint which are estimated to have the same deflection angle and therefore the same curvature. In a simplified example, a continuum joint with two sensors with EE placement only has two distinct regions of estimation, the segments covered by the first sensor and the segments covered by the second sensor. A continuum joint with two sensors using the HO placement has three distinct regions of estimation, the segments covered exclusively by the first sensor, the segments covered exclusively by the second sensor and the segments covered by both sensors. The E-WAM method is essentially the DEM method but it averages the overlapping sensors that are covering a segment. This creation of additional estimation regions is what allows the HO method to have lower error than the EE method, even when using a simple estimation method such as E-WAM.

Our third hypothesis, which is the main contribution of this paper, is that a weighted linear combination of overlapping sensor data can significantly reduce state estimation error when compared to simpler estimation methods such as E-WAM and DEM. The reduction in error from HO with E-WAM to HO with WAM shown in **Figures 7**, **8** is dramatic. This data is highlighted in **Table 3**. We can easily see how overlapping sensors creates additional regions of estimation with the simple estimation method E-WAM. The WAM method takes that one step further by using linear regression to derive unique sensor value weights for estimating the state of each segment, thus creating a distinct region of estimation for each segment. This means that each joint segment can have a unique estimated deflection angle with minimum of two sensor. To achieve this with E-WAM $m - 1$ sensors are needed, where m is the number of constant curvature segments of the joint. For example, the proximal most segment is always bent at an angle that is greater than or equal to the bending angle of the next most proximal segment. This can be expressed by the WAM method when it calculates slightly different weights for segments zero and one, even though they may be covered by the same set of sensors.

Furthermore, the WAM method allows for sensors that are not covering a segment to provide information about the robot state. By using a linear combination of all the sensor measurements on the joint, not just the ones covering the segment, WAM is able to significantly reduce the deflection angle estimation error as compared to E-WAM. For example, if the proximal segments have a sensor reading associated with a negative bending angle and the distal segments have a sensor reading associated with a positive bending angle, that information can be captured by the weights of the WAM method to determine that there will be a point of inflection in the middle of the joint and therefore middle segments will have small deflection angles in this situation.

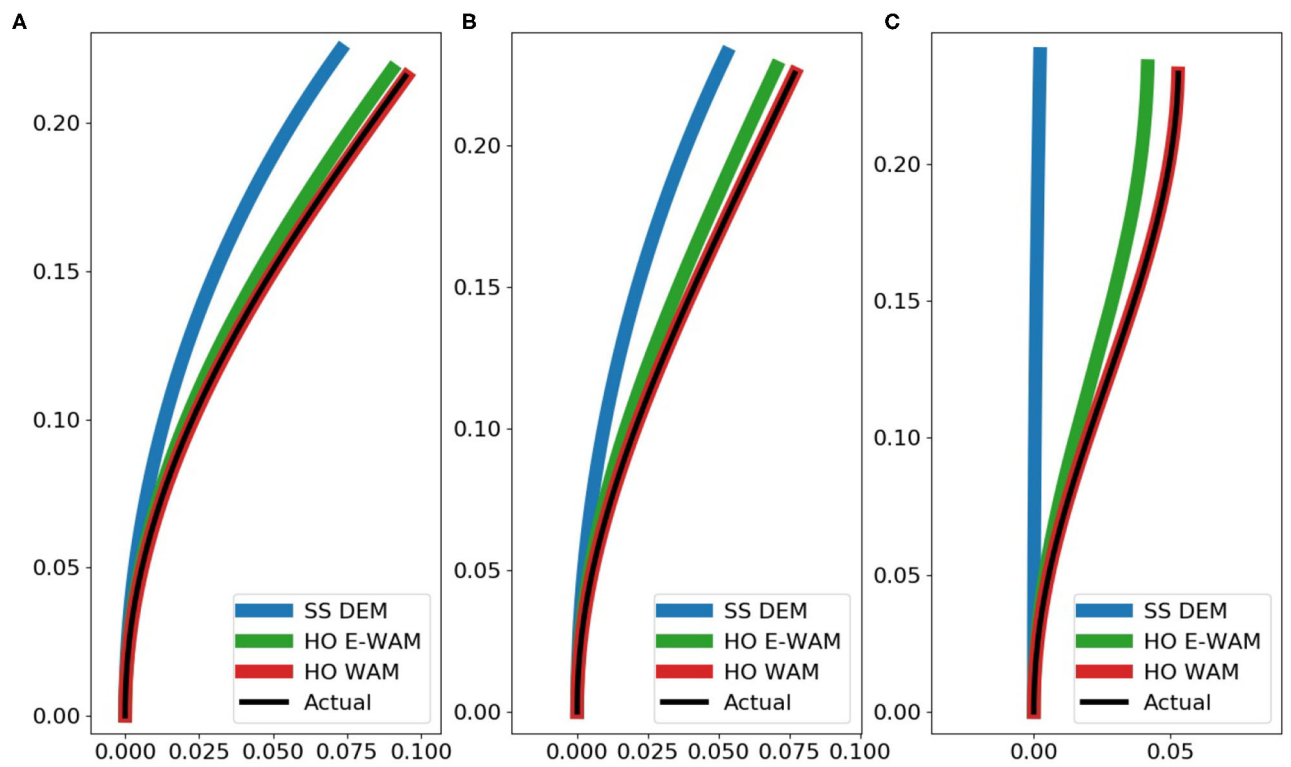


FIGURE 8 | Case study of effectiveness of estimation methods for a single case. Sensors placed using HO. Number of segments = 12, max ROM = 3 deg, and number of sensors = 3 placed at [0,4,3,8,7,11]. The plots show the following loading conditions: **(A)** end force load, **(B)** uniformly distributed load, **(C)** moment and end force load. All units in the plots are reported in meters.

TABLE 3 | Table highlighting the difference in median error as a % of ROM between HO with E-WAM and HO with WAM.

Number of sensors	2	3	4	5	6
HO with E-WAM median error as % of ROM	7.561	4.504	3.090	2.342	1.914
HO with WAM median error as % of ROM	1.552	0.261	0.0915	0.0434	0.0366
Decrease in median error as % of ROM	**6.009**	**4.243**	**2.9985**	**2.2986**	**1.8774**

The bold values shown in the table highlight the improvements between the two methods being compared.

TABLE 4 | Table highlighting the difference in median error as a % of ROM between HO with WAM and OO with WAM.

Number of sensors	2	3	4	5	6
HO with WAM median error as % of ROM	1.552	0.262	0.0915	0.0434	0.0366
OO with WAM median error as % of ROM	1.542	0.238	0.0836	0.0368	0.0315
Decrease in median error as % of ROM	**0.010**	**0.0240**	**0.00790**	**0.0066**	**0.0051**

The bold values shown in the table highlight the improvements between the two methods being compared.

Our final hypothesis was that an evolutionary algorithm could be used to determine the optimal placement of overlapping sensors such that state estimation will be further improved than using WAM with the Heuristic Overlap. This hypothesis is only loosely supported by the data collected in our simulated experiments. Since the bars in **Figure 7** are so small, the data comparing HO and OO with WAM are highlighted in **Table 4**. There is always a reduction in error when using OO instead of HO, however that reduction is very small. We mainly attribute this to the WAM method being able to estimate the shape so accurately that it is difficult to reduce the error even further using "optimal" sensor placements. We also believe the HO placement method already provides a fairly optimal, even coverage of all the segments on the joint. The largest reduction in error observed

occurs when using three sensors. We believe the benefits of OO peak at three sensors because it is when using more than three sensors with HO there is already excellent coverage of the segments and when using two sensors there aren't many possible configurations so there is only a modest reduction in error from optimizing.

In conclusion, we have shown that state estimation of a continuum joint can be significantly improved by using the WAM estimation method on overlapping sensors which are placed on the continuum joint according to a simple heuristic. Using this method with only three sensors yielded a median joint angle error (as a percentage of the range of motion) of 0.262%. Increasing the number of sensors further reduced the state estimation error to under 0.1%. We have also shown that the simple heuristic overlap performs almost as well as an optimized overlapping arrangement determined by an evolutionary algorithm with the median error (as a percent range of motion), being $<0.025\%$ for all cases tested.

Some sources of error in this work could come from the shapes of the joints in the real world not being as ideal as the simulated ones we used for testing. This would mean that the median errors determined in this paper would be slightly higher when implemented on hardware even with ideal sensors. Even with this introduction of uncertainty, we are confident the reduction in error seen from using WAM in simulation will translate to large, real world reductions in error. A simple way to improve the estimation would be to collect test data from the hardware and perform linear regression on that data rather than simulated data. Nonetheless, future work will entail implementing these sensor placement and configuration estimation methods on hardware and testing their capabilities for a non-idealized sensor. Given noise or other possible sources of error introduced by the hardware, this will be important to prove that the approach is as effective in the real world as is predicted by these kinematic and static loading simulations.

DATA AVAILABILITY STATEMENT

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

AUTHOR CONTRIBUTIONS

LR and TD developed the concepts presented here with support from MK. All authors contributed equally to the manuscript.

FUNDING

This work was partly supported by NASA contract #80NSSC19C0637, the Office of Naval Research contract #N68335-17-C-0045, and ARPA-E contract #DE-AR0001241. This material is based upon work that was also supported by the National Science Foundation under Grant no. 1935312.

REFERENCES

Abbas, A., and Zhao, J. (2017). "Twisted and coiled sensor for shape estimation of soft robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, CA), 482–487. doi: 10.1109/IROS.2017.8202197

Allen, T. F., Rupert, L., Duggan, T. R., Hein, G., and Albert, K. (2020). "Closed-form non-singular constant-curvature continuum manipulator kinematics," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)* (New Haven, CT), 410–416. doi: 10.1109/RoboSoft48309.2020.9116015

Anderson, P. L., Mahoney, A. W., and Webster, R. J. (2017). Continuum reconfigurable parallel robots for surgery: shape sensing and state estimation with uncertainty. *IEEE Robot. Autom. Lett.* 2, 1617–1624. doi: 10.1109/LRA.2017.2678606

Avery, J., Runciman, M., Darzi, A., and Mylonas, G. P. (2019). "Shape sensing of variable stiffness soft robots using electrical impedance tomography," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC), 9066–9072. doi: 10.1109/ICRA.2019.8793862

Bilodeau, R. A., Yuen, M. C., Case, J. C., Buckner, T. L., and Kramer-Bottiglio, R. (2018). "Design for control of a soft bidirectional bending actuator," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid), 1–8. doi: 10.1109/IROS.2018.8594293

Borum, A., Matthews, D., and Bretl, T. (2014). "State estimation and tracking of deforming planar elastic rods," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong King), 4127–4132. doi: 10.1109/ICRA.2014.6907459

Case, J. C., Booth, J., Shah, D. S., Yuen, M. C., and Kramer-Bottiglio, R. (2018). "State and stiffness estimation using robotic fabrics," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)* (Livorno), 522–527. doi: 10.1109/ROBOSOFT.2018.8405379

Chen, W., Xiong, C., Liu, C., Li, P., and Chen, Y. (2019). Fabrication and dynamic modeling of bidirectional bending soft actuator integrated with optical waveguide curvature sensor. *Soft Robot.* 6, 495–506. doi: 10.1089/soro.2018.0061

Cianchetti, M., Renda, F., Licofonte, A., and Laschi, C. (2012). "Sensorization of continuum soft robots for reconstructing their spatial configuration," in *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)* (Rome), 634–639. doi: 10.1109/BioRob.2012.6290788

Clark, E., Brunton, S. L., and Kutz, J. N. (2020). Multi-fidelity sensor selection: Greedy algorithms to place cheap and expensive sensors with cost constraints. *IEEE Sensors J.* 21, 600–611. doi: 10.1109/JSEN.2020.3013094

Day, N. M. (2018). *Tactile sensing and position estimation methods for increased proprioception of soft-robotic platforms* (Master's Thesis). Brigham Young University, Provo, UT, United States.

Denasi, A., Khan, F., Boskma, K. J., Kaya, M., Hennersperger, C., Göbl, R., et al. (2018). "An observer-based fusion method using multicore optical shape sensors and ultrasound images for magnetically-actuated catheters," in *2018 IEEE International Conference on Robotics and Automation (ICRA)* (Brisbane), 50–57. doi: 10.1109/ICRA.2018.8462695

Deutschmann, B., Chalon, M., Reinecke, J., Maier, M., and Ott, C. (2019). Six-DoF pose estimation for a tendon-driven continuum mechanism without a deformation model. *IEEE Robot. Autom. Lett.* 4, 3425–3432. doi: 10.1109/LRA.2019.2927943

DeVries, L., and Paley, D. A. (2013). "Observability-based optimization for flow sensing and control of an underwater vehicle in a uniform flowfield," in *2013 American Control Conference* (Washington, DC), 1386–1391. doi: 10.1109/ACC.2013.6580030

Dobrzynski, M. K., Pericet-Camara, R., and Floreano, D. (2011). "Contactless deflection sensor for soft robots," in *IEEE International Conference on Intelligent Robots and Systems* (San Francisco, CA), 1913–1918. doi: 10.1109/IROS.2011.6094845

Elgeneidy, K., Lohse, N., and Jackson, M. (2016). Data-driven bending angle prediction of soft pneumatic actuators with embedded flex sensors. *IFAC-PapersOnLine* 49, 513–520. doi: 10.1016/j.ifacol.2016.10.654

Elgeneidy, K., Neumann, G., Jackson, M., and Lohse, N. (2018). Directly printable flexible strain sensors for bending and contact feedback of soft actuators. *Front. Robot. AI* 5:2. doi: 10.3389/frobt.2018.00002

Felt, W., Lu, S., and Remy, C. D. (2018). Modeling and design of "smart braid" inductance sensors for fiber-reinforced elastomeric enclosures. *IEEE Sensors J.* 18, 2827–2835. doi: 10.1109/JSEN.2018.2802640

Felt, W., Suen, M., and Remy, C. D. (2016). "Sensing the motion of bellows through changes in mutual inductance," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon), 5252–5257. doi: 10.1109/IROS.2016.7759772

Felt, W., Telleria, M. J., Allen, T. F., Hein, G., Pompa, J. B., Albert, K., et al. (2019). An inductance-based sensing system for bellows-driven continuum joints in soft robots. *Auton. Robots* 43, 435–448. doi: 10.1007/s10514-018-9769-7

Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M., and Gagné, C. (2012). DEAP: evolutionary algorithms made easy. *J. Mach. Learn. Res.* 13, 2171–2175. doi: 10.5555/2503308.2503311

Gerboni, G., Diodato, A., Ciuti, G., Cianchetti, M., and Menciassi, A. (2017). Feedback control of soft robot actuators via commercial flex bend sensors. *IEEE/ASME Trans. Mechatron.* 22, 1881–1888. doi: 10.1109/TMECH.2017.2699677

Gibbs, P. T., and Asada, H. H. (2005). Wearable conductive fiber sensors for multi-axis human joint angle measurements. *J. NeuroEng. Rehabil.* 2:7. doi: 10.1186/1743-0003-2-7

Godaba, H., Vitanov, I., Aljaber, F., Ataka, A., and Althoefer, K. (2020). "A bending sensor insensitive to pressure: soft proprioception based on abraded optical fibres," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)* (New Haven, CT), 104–109. doi: 10.1109/RoboSoft48309.2020.9115984

Hannan, M. W., and Walker, I. D. (2003). Kinematics and the Implementation of an Elephant's trunk manipulator and other continuum style robots. *J. Robot. Syst.* 20, 45–63. doi: 10.1002/rob.10070

He, Y., Zhang, X., Zhu, L., Sun, G., Lou, X., and Dong, M. (2019). Curvature and force measurement of soft manipulator based on stretchable helical optic fibre. *Opt. Fiber Technol.* 53:102010. doi: 10.1016/j.yofte.2019.102010

Hyatt, P., Kraus, D., Sherrod, V., Rupert, L., Day, N., and Killpack, M. D. (2019). Configuration estimation for accurate position control of large-scale soft robots. *IEEE/ASME Trans. Mechatron.* 24, 88–99. doi: 10.1109/TMECH.2018.2878228

Katzschmann, R. K., Thieffry, M., Goury, O., Kruszewski, A., Guerra, T.-M., Duriez, C., et al. (2019). "Dynamically closed-loop controlled soft robotic arm using a reduced order finite element model with state observer," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)* (Seoul), 717–724. doi: 10.1109/ROBOSOFT.2019.8722804

Kim, B., Ha, J., Park, F. C., and Dupont, P. E. (2014). "Optimizing curvature sensor placement for fast, accurate shape sensing of continuum robots," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 5374–5379. doi: 10.1109/ICRA.2014.6907649

Krener, A. J., and Ide, K. (2009). "Measures of unobservability," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference* (Shanghai), 6401–6406. doi: 10.1109/CDC.2009.5400067

Li, T., Qiu, L., and Ren, H. (2020). Distributed curvature sensing and shape reconstruction for soft manipulators with irregular cross sections based on parallel dual-FBG arrays. *IEEE/ASME Trans. Mechatron.* 25, 406–417. doi: 10.1109/TMECH.2019.2949151

Lun, T. L. T., Wang, K., Ho, J. D. L., Lee, K.-H., Sze, K. Y., and Kwok, K.-W. (2019). Real-time surface shape sensing for soft and flexible structures using fiber bragg gratings. *IEEE Robot. Autom. Lett.* 4, 1454–1461. doi: 10.1109/LRA.2019.2893036

Mahoney, A. W., Bruns, T. L., Swaney, P. J., and Webster, R. J. (2016). "On the inseparable nature of sensor selection, sensor placement, and state estimation for continuum robots or 'where to put your sensors and how to use them'," in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (Stockholm), 4472–4478. doi: 10.1109/ICRA.2016.7487646

Marchese, A. D., Komorowski, K., Onal, C. D., and Rus, D. (2014). "Design and control of a soft and continuously deformable 2D robotic manipulation system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong), 2189–2196. doi: 10.1109/ICRA.2014.6907161

Melingui, A., Escande, C., Benoudjit, N., Merzouki, R., and Mbede, J. B. (2014). Qualitative approach for forward kinematic modeling of a Compact Bionic Handling Assistant trunk. *IFAC Proc. Vol.* 19, 9353–9358. doi: 10.3182/20140824-6-ZA-1003.01758

Merzouki, R., Melingui, A., and Mbede, J. (2014). Compact bionic handling arm control using neural networks. *Electron. Lett.* 50, 979–981. doi: 10.1049/el.2014.1136

Monet, F., Sefati, S., Lorre, P., Poiffaut, A., Kadoury, S., Armand, M., et al. (2020). "High-resolution optical fiber shape sensing of continuum robots: a comparative study," in *2020 IEEE International Conference on Robotics and Automation (ICRA)* (Paris), 8877–8883. doi: 10.1109/ICRA40945.2020.9197454

Nakagawa, N., and Mochiyama, H. (2018). "Real-time shape estimation of an elastic rod using a robot manipulator equipped with a sense of force," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid), 8067–8073. doi: 10.1109/IROS.2018.8593946

Oliveira, J., Ferreira, A., and Reis, J. C. (2020). Design and experiments on an inflatable link robot with a built-in vision sensor. *Mechatronics* 65:102305. doi: 10.1016/j.mechatronics.2019.102305

Ozel, S., Skorina, E. H., Luo, M., Tao, W., Chen, F., Pan, Y., et al. (2016). "A composite soft bending actuation module with integrated curvature sensing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)* (Stockholm), 4963–4968. doi: 10.1109/ICRA.2016.7487703

Qi, J., Sun, K., and Kang, W. (2015). Optimal PMU placement for power system dynamic state estimation by using empirical observability gramian. *IEEE Trans. Power Syst.* 30, 2041–2054. doi: 10.1109/TPWRS.2014.2356797

Rone, W. S., and Ben-tzvi, P. (2013). "Multi-segment continuum robot shape estimation using passive cable displacement," in *2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE)* (Washington, DC). doi: 10.1109/ROSE.2013.6698415

Rucker, D. C., and Webster, R. J. (2011). "Deflection-based force sensing for continuum robots: a probabilistic approach," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Francisco, CA), 3764–3769. doi: 10.1109/IROS.2011.6094526

Sadati, S. M. H., Shiva, A., Herzig, N., Rucker, C. D., Hauser, H., Walker, I. D., et al. (2020). Stiffness imaging with a continuum appendage: real-time shape and tip force estimation from base load readings. *IEEE Robot. Autom. Lett.* 9, 2824–2831. doi: 10.1109/LRA.2020.2972790

Scimeca, L., Hughes, J., Maiolino, P., and Iida, F. (2019). Model-free soft-structure reconstruction for proprioception using tactile arrays. *IEEE Robot. Autom. Lett.* 4, 2479–2484. doi: 10.1109/LRA.2019.2906548

She, Y., Li, C., Cleary, J., and Su, H.-J. (2015). Design and fabrication of a soft robotic hand with embedded actuators and sensors. *J. Mech. Robot.* 7:021007. doi: 10.1115/1.4029497

Sheng, J., Deaton, N. J., and Desai, J. P. (2019). "A large-deflection FBG bending sensor for SMA bending modules for steerable surgical robots," in *2019 International Conference on Robotics and Automation (ICRA)* (Montreal, QC), 900–906. doi: 10.1109/ICRA.2019.8794302

Sklar, E. I., Sareh, S., Secco, E. L., Faragasso, A., and Althoefer, K. (2016). A non-linear model for predicting tip position of a pliable robot arm segment using bending sensor data. *Sensors Transduc.* 199, 52–61.

Song, S., Li, Z., Yu, H., and Ren, H. (2015). Electromagnetic positioning for tip tracking and shape sensing of flexible robots. *IEEE Sensors J.* 15, 4565–4575. doi: 10.1109/JSEN.2015.2424228

Takano, R., Mochiyama, H., and Takesue, N. (2017). "Real-time shape estimation of kirchhoff elastic rod based on force/torque sensor," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Marina Bay Sands), 2508–2515. IEEE. doi: 10.1109/ICRA.2017.7989292

Tapia, J., Knoop, E., Mutný, M., Otaduy, M. A., and Bächer, M. (2020). MakeSense: automated sensor design for proprioceptive soft robots. *Soft Robot.* 7, 332–345. doi: 10.1089/soro.2018.0162

Trivedi, D., and Rahn, C. D. (2009). "Shape sensing for soft robotic manipulators," in *Volume 7: 33rd Mechanisms and Robotics Conference, Parts A and B* (San Diego, CA: ASMEDC), 335–343. doi: 10.1115/DETC2009-87598

Trivedi, D., and Rahn, C. D. (2014). Model-based shape estimation for soft robotic manipulators: the planar case. *J. Mech. Robot.* 6:021005. doi: 10.1115/1.4026338

Truby, R. L., Santina, C. D., and Rus, D. (2020). Distributed proprioception of 3D configuration in soft, sensorized robots via deep learning. *IEEE Robot. Autom. Lett.* 5, 3299–3306. doi: 10.1109/LRA.2020.2976320

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in python. *Nat. Methods* 17, 261–272. doi: 10.1038/s41592-019-0686-2

Wang, H., Zhang, R., Chen, W., Liang, X., and Pfeifer, R. (2016). Shape detection algorithm for soft manipulator based on fiber bragg gratings. *IEEE/ASME Trans. Mechatron.* 21, 2977–2982. doi: 10.1109/TMECH.2016.2606491

Wang, X., Fang, G., Wang, K., Xie, X., Lee, K.-H., Ho, J. D. L., et al.. (2020). Eye-in-hand visual servoing enhanced with sparse strain measurement for soft continuum robots. *IEEE Robot. Autom. Lett.* 5, 2161–2168. doi: 10.1109/LRA.2020.2969953

Wurdemann, H. A., Sareh, S., Shafti, A., Noh, Y., Faragasso, A., Chathuranga, D. S., et al. (2015). "Embedded electro-conductive yarn for shape sensing of soft robotic manipulators," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Milan), 8026–8029. doi: 10.1109/EMBC.2015.7320255

You, X., Zhang, Y., Chen, X., Liu, X., Wang, Z., Jiang, H., et al. (2017). "Model-free control for soft manipulators based on reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Vancouver, CA), 2909–2915. doi: 10.1109/IROS.2017.8206123

Yuan, Y., Wu, G., Li, X., Fan, Y., and Wu, X. (2011). Effects of twisting and bending on LP_21 mode propagation in optical fiber. *Opt. Lett.* 36:4248. doi: 10.1364/OL.36.004248

Yuen, M. C., Tonoyan, H., White, E. L., Telleria, M., and Kramer, R. K. (2017). "Fabric sensory sleeves for soft robot state estimation," in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (Marina Bay Sands: IEEE), 5511–5518. doi: 10.1109/ICRA.2017.7989649

Yuen, M. C.-S., Lear, T. R., Tonoyan, H., Telleria, M., and Kramer-Bottiglio, R. (2018). Toward closed-loop control of pneumatic grippers during pack-and-deploy operations. *IEEE Robot. Autom. Lett.* 3, 1402–1409. doi: 10.1109/LRA.2018.2800079

Zhao, J., and Abbas, A. (2016). "A low-cost soft coiled sensor for soft robots," in *Volume 2: Mechatronics; Mechatronics and Controls in Advanced Manufacturing; Modeling and Control of Automotive Systems and Combustion Engines; Modeling and Validation; Motion and Vibration Control Applications; Multi-Agent and Networked Systems; Path Planning and Motion Control; Robot Manipulators; Sensors and Actuators; Tracking Control Systems; Uncertain Systems and Robustness; Unmanned, Ground and Surface Robotics; Vehicle Dynamic Controls; Vehicle Dynamics and Traffic Control* (Minneapolis, MN: American Society of Mechanical Engineers), V002T26A006. doi: 10.1115/DSCC2016-9916

Zhou, J., Chen, Y., Chen, X., Wang, Z., Li, Y., and Liu, Y. (2020). A proprioceptive bellows (PB) actuator with position feedback and force estimation. *IEEE Robot. Autom. Lett.* 5, 1867–1874. doi: 10.1109/LRA.2020.2969920

Zhuang, W., Sun, G., Li, H., Lou, X., Dong, M., and Zhu, L. (2018). FBG based shape sensing of a silicone octopus tentacle model for soft robotics. *Optik* 165, 7–15. doi: 10.1016/j.ijleo.2018.03.087

Conflict of Interest: TD is employed by company Otherlab Inc., while LR was formerly employed by company Otherlab Inc.

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

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership