IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Shape Control for Modular Continuum Soft Arms: A Distributed Approach to Address Redundancy

Samuele Bordini, *Graduate Student Member, IEEE*, Daniele Caradonna, *Graduate Student Member, IEEE*, Antonio Bicchi, *Life Fellow, IEEE*, and Egidio Falotico, *Member, IEEE*

***Abstract*—Modular continuum soft arms represent an emerging class of robotic systems characterized by flexible, highly deformable structures. Designing shape controllers for these arms poses significant challenges due to their modeling complexity and hyper-redundant nature. Our goal is to develop a scalable control framework for modular arms, where each module is self-contained. Starting from distributed control theory, we assign a collaborative controller to each soft module. Through collaboration among modules, the framework enables the system to achieve the desired tip position and shape. Each controller relies on the minimal model, such as the Constant Curvature, of its self-contained module and the local transformation shared by adjacent modules. We present three kinematic control strategies - Consensus, Bipartite Consensus, and Formation Control - for a modular continuum soft arm, that progressively relax constraints to achieve more complex, adaptable shapes. In addition, we develop a decentralized curvature-based dynamic controller to manage dynamic coupling among modules. The validation is carried out through numerical analysis and dynamic simulations of soft arms with varying numbers of modules.**

***Index Terms*—Modeling, control, and learning for soft robots, distributed robot systems.**

I. Introduction

SOFT robotics [1] aims to use soft and smart materials to create more compliant robot structures, enhancing the safety of physical interactions with the environment. The field of continuum soft arms is particularly expanding, representing a new generation of flexible arms inspired by muscular hydrostats [2]. These arms utilize continuous deformation to achieve novel capabilities compared to traditional rigid robots: dexterity, interaction ability, and environmental adaptability. Achieving this requires effective shape control in real-world applications, particularly in unstructured and cluttered environments where adaptability is necessary. However, designing a shape controller for these arms presents significant challenges due to their complex modeling, material properties, and wide range of potential technological solutions [3], [4]. One such technological solution is modular continuum soft robots [5], where the model and control are inherently dependent on the number of modules.

We aim to design a scalable control framework for a generic number of soft arm modules, i.e., independent of the module count. To achieve this, we create self-contained modules in terms of control that operate autonomously and communicate locally to ensure the desired overall behavior. This allows modules to be added or removed without altering the modeling and control phases. We present a distributed control approach for modular continuum soft arms. The distributed framework [6], summarized in Fig. 1, consists of the linear topology graph representing the controller network. The generic agent A_i controls the self-contained soft module M_i. Agents collaborate through communication channels c_i to achieve the global behavior, i.e., the desired shape. Each module must know the local position of its adjacent ones. Moreover, each controller requires only its module's model, ensuring the self-contained property and allowing the robot to be represented with a minimal model, such

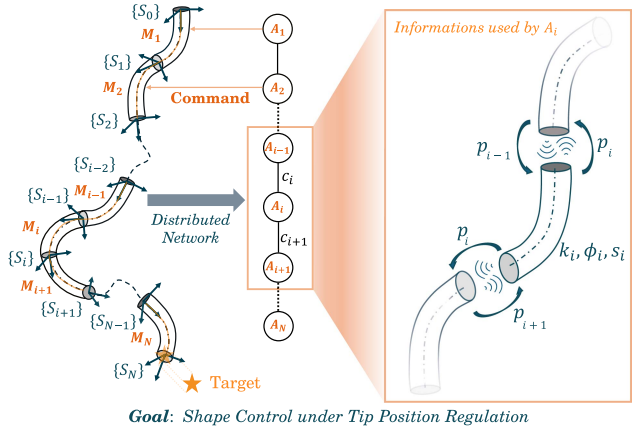

Fig. 1. Distributed framework for shape control of modular continuum soft arms. The left side illustrates the modular soft arm achieving Shape Control under Tip Position Regulation. Each module (M_i) collaborates with adjacent controllers (A_i) via communication channels (c_i). The zooms shows the information used by the A_i: the geometrical model of i^{th} module, denoted by the variables k_i, ϕ_i, and s_i, and the shared local position $\boldsymbol{p}_{i-1}$ and $\boldsymbol{p}_{i+1}$.

Received 25 March 2025; accepted 4 August 2025. Date of publication 11 August 2025; date of current version 20 August 2025. This article was recommended for publication by Associate Editor L. Pimenta and Editor M. A. Hsieh upon evaluation of the reviewers' comments. This work was supported in part by Fondazione Pisa - Scientific and Technological Research through the SHARE-CP Project under Grant CUP J83C24000530007 and in part by MAE (Italy) and A*STAR (Singapore) through the DESTRO Project "Dextrous, strong yet soft robots," under Grant R22I0IR124. *(Corresponding author: Samuele Bordini.)*

Samuele Bordini is with the Centro di Ricerca "Enrico Piaggio", Università di Pisa, 56126 Pisa, Italy, and also with the Dipartimento di Ingegneria dell'Informazione, Università di Pisa, 56126 Pisa, Italy (e-mail: samuele.bordini@phd.unipi.it).

Daniele Caradonna and Egidio Falotico are with the The BioRobotics Institute, Scuola Superiore Sant'Anna, 56127 Pisa, Italy, and also with the Department of Excellence in Robotics and AI, Scuola Superiore Sant'Anna, 56127 Pisa, Italy.

Antonio Bicchi is with the Centro di Ricerca "Enrico Piaggio", Università di Pisa, 56126 Pisa, Italy, and with the Dipartimento di Ingegneria dell'Informazione, Università di Pisa, 56126 Pisa, Italy, and also with the Soft Robotics for Human Cooperation and Rehabilitation, Fondazione Istituto Italiano di Tecnologia, 16163 Genova, Italy.

Digital Object Identifier 10.1109/LRA.2025.3597852

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

as Constant Curvature (CC) [7]. This significantly minimizes the modeling effort and simplifies the control law, reducing the control algorithm's computational complexity. Finally, our distributed algorithm requires only the local relative tip position of the modules, allowing us to achieve the agreement directly on the configuration vector. We strive to achieve a Shape Control (SC) under Tip Position Regulation (TPR), i.e., leveraging the redundancy degrees of the soft robot to create a feasible shape with respect to the desired tip position. To accomplish the task, we introduce a comprehensive control framework composed of a kinematic controller and a dynamic controller. The former control approach consists of adapting three well-known distributed kinematic strategies from the multi-vehicle community - Consensus, Bipartite Consensus, and Formation Control - to a modular continuum soft arm. They progressively relax the constraints to allow more complex and adaptable shapes. The three developed control laws enable the same tip position to be reached with different shapes. They are applied incrementally during the validation process, with each law building on the previous one to progressively enhance shape complexity, maintaining the same desired tip position. The latter is a decentralized curvature-based dynamic controller that guarantees desirable behavior during the transient state.

The main contribution of this work is the design of a scalable control framework based on self-contained modules to perform SC with TPR. This is achieved by introducing: i) three distributed kinematic control strategies for modular continuum soft arms in 3D space, relying solely on relative position as the communicated information, namely, Consensus, Bipartite Consensus, and Formation Control; ii) a decentralized, curvature-based dynamic controller. Finally, dynamic validation is performed in the Soft Robot Simulator (SoRoSim) [8] framework on a 3-module and 5-module soft robot arm.

II. Related Works

Unlike rigid robotics, the state of the art in modeling and control of soft robots has yet to reach a gold standard. In terms of modeling, the research community proposes many techniques to manage the infinite Degrees of Freedom (DoFs) and the strong non-linearities of the dynamics. In [3], the Authors grouped the modeling approaches into four classes: (i) Continuum Mechanics models, (ii) Geometrical models, (iii) Discrete Material models, and (iv) Surrogate models. The main goal is to address a trade-off between the accuracy and computation efficiency of the model.

With the advancement of models in recent years, numerous studies have emerged that address soft robot model-based control [2], [4], [9]. The design of kinematic and dynamic controllers is particularly challenging for soft robot applications, primarily due to the complexity of the model and the multitude of potential technological solutions.

Closed-loop kinematic controllers are based on differential kinematic inversion, and they allow for performing TPR or SC, considering only the steady-state behavior. In [10], [11] and [12], a controller based on a geometric model is presented to perform TPR. In [13], perform a tip pose trajectory control using the Geometrically Exact Model. Finally, in [14], SC is performed for a three-dimensional robot.

The dynamic controller also guarantees the desired behavior during the transient state. With the centralized approach, the controller needs the entire dynamic model. In [15], the authors design a three-dimensional control framework based on the Augmented Formulation of Piecewise Constant Curvature (PCC), performing a tip trajectory. Similarly, [16] introduces a sliding mode approach for TPR. Shape control is also performed directly with the dynamic model, as in [17], and [18]. Finally, [19] presents a TPR and SC separately, with the use of dynamics. These approaches are centralized, suffer from model complexity, and are not scalable for modular structures. Notably, the model complexity of the modular continuum soft arm can increase substantially as the dimensionality of the actuator and configuration spaces grows.

In recent years, decentralized and distributed strategies have emerged as a class of both kinematic and dynamic controllers, aimed at reducing the number of DoFs directly managed by a centralized controller. In [20], position control is performed with a decentralized kinematic controller for a modular soft structure. In [21], a distributed consensus strategy is used to perform a TPR using a kinematic PCC model with 2D Augmented Formulation. In [22] and [23], Doroudchi et al. design the dynamic distributed controller to regulate the tip position with Euler-Bernoulli and Cosserat models, respectively. Recently, a decentralized approach to the sensing system was proposed in [24], while in [25], the authors presented a distributed kinematic control strategy to robustly regulate the orientation of the soft arm tip.

Our solution is based on distributed system control [6]. This approach was born for multi-vehicle systems [26] or swarm robotics to ensure the desired behavior of the entire system through local interaction rules. In our work, the multi-robot systems are composed of N controllers, i.e., agents, that command each module of the soft arm. Graph theory [27] serves as a fundamental tool for systematically representing distributed systems, providing natural abstractions for how information is shared among agents in a network. The proposed strategies are based on Consensus [6, Chap. 3] and [21], Bipartite Consensus [28], [29], and Formation Control [6, Chap. 6].

III. Modeling Modular Continuum Soft Arms

The control method relies on a kinematic and dynamic model that satisfies the following assumptions: (i) the kinematic model must yield a Jacobian matrix such that its pseudo-inverse exists; (ii) it must provide an inverse kinematics solution that can uniquely reconstruct the agent's configuration (i.e., its state) from the relative position and orientation between adjacent agents; and (iii) the dynamic model must include an actuation matrix for which the pseudo-inverse exists. The proposed control strategies are applicable to any model [3] that meets these assumptions. An example of such a model is the PCC representation, which captures the dominant deformation modes - namely curvature and elongation - using a minimal parameterization [4].

A. PCC With Exponential Coordinates

The most commonly used steady-state model is the PCC, a geometric model [3]. PCC robots are modeled as consisting of a finite number of curved links described by arc parameters [7], [11], [15]. This model accurately approximates the robot when the manipulator is uniform, symmetrically actuated, with minimal external loading and negligible torsional and shear effects [2]. With our framework, each module is independently

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

modeled as a CC segment, ensuring a compact formulation that reinforces the practicality of this approach.

1) Kinematics: The most compact and numerically robust [30] representation [7] is that with exponential coordinates based on Lie Group Theory [31]. The configuration vector for each CC segment is defined as $\boldsymbol{q}_i = \begin{bmatrix} k_i & \phi_i & s_i \end{bmatrix}^\top \in \mathbb{R}^3$, where $k_i \in \mathbb{R}$ is the curvature, $\phi_i \in [-\pi, \pi]$ is the angle of the plane containing the arc, and $s_i \in \mathbb{R}^+$ is the arc length. The latter is inserted into the state vector to model the degree of elongation strain, which is very significant for actuated soft robots. The transformation [11] is parameterized by the twist vector $\boldsymbol{\zeta}_1$ and $\boldsymbol{\zeta}_2 \in \mathbb{R}^6$. The former refers to the transformation on the z axis by the ϕ_i angle, and the latter to the transformation about the local y axis by the θ_i angle, recalling that $\theta_i = k_i s_i$. Finally, each homogeneous transformation $\boldsymbol{T}_i \in SE(3)$ is defined as

$$\boldsymbol{T}_i = e^{\hat{\boldsymbol{\zeta}}_1 \phi_i} e^{\hat{\boldsymbol{\zeta}}_2 s_i} = e^{\hat{\boldsymbol{v}}_6 \phi_i} e^{(\hat{\boldsymbol{v}}_3 + \hat{\boldsymbol{v}}_4 k_i) s_i}, \tag{1}$$

for $i = 1, 2, \ldots, m$, and where $\boldsymbol{v}_j$ is the j^{th} element of the standard canonical basis for $\mathbb{R}^6$. We define $\hat{(\cdot)}$ as the *hat* operator [31]. Moreover, it is possible to extract the analytical expression of the Jacobian matrix $\boldsymbol{J}_i \in \mathbb{R}^{6\times 3}$ [11] for the i^{th} segment. It is important to note that the Jacobian matrix remains well-defined as the curvature approaches zero, meaning the limit is finite and no singularity occurs.

2) Dynamics: We focus on the dynamics from the CC formulation for a single segment, aligned with our objectives. Similar to the general case, obtaining the dynamic model combining differential kinematics with the system's physical characteristics using the Lagrangian formalism:

$$\begin{aligned} \boldsymbol{B}(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i + \boldsymbol{C}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i)\dot{\boldsymbol{q}}_i + \boldsymbol{G}(\boldsymbol{q}_i) + \boldsymbol{K}(\boldsymbol{q}_i) \\ + \boldsymbol{D}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) = \boldsymbol{A}(\boldsymbol{q}_i)\boldsymbol{\tau}_i \,. \end{aligned} \tag{2}$$

$\boldsymbol{B}(\boldsymbol{q}_i) \in \mathbb{R}^{3\times 3}$, $\boldsymbol{C}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) \in \mathbb{R}^{3\times 3}$, and $\boldsymbol{G}(\boldsymbol{q}_i) \in \mathbb{R}^3$ are inertia, Coriolis, and gravitational terms, respectively. Unlike rigid robots, additional terms such as stiffness $\boldsymbol{K}(\boldsymbol{q}_i) \in \mathbb{R}^3$ and damping $\boldsymbol{D}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i) \in \mathbb{R}^3$ are included. These matrices are approximated linearly, assuming constant stiffness and damping factors along the robot's length. Finally, $\boldsymbol{A}(\boldsymbol{q}_i) \in \mathbb{R}^{3\times 3}$ is the actuation matrix and $\boldsymbol{\tau}_i \in \mathbb{R}^n$ is the generalized control forces. Considering the actuation system described in [32], we assume the system is fully actuated and derive the differential map from the configuration space to the actuation space. The interactions with the external environment are not considered in this work.

IV. Control Architecture

A. Control Framework Definition

The soft arm consists of self-contained modules at the hardware level. Each module features independent actuation, such as pneumatic or tendon-driven systems, and is equipped with embedded local sensors, e.g., IMUs [24], [25]. Modules must be capable of exchanging relative position information with neighboring units, for instance, via wireless communication. These modules are controlled by agents that know the local position of the module tip and share this information with adjacent ones.

The modules are described by a base frame and a tip frame, where the base frame of agent i coincides with the tip frame of agent $(i-1)$. This ensures continuity of the transformation between modules, as shown in Fig. 1. The agent Nth is the *leader*. It knows the tip pose with respect to the $\{S_0\}$ frame and leads the other agents accordingly with the position error.

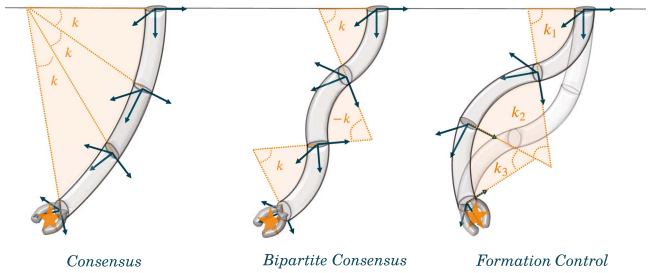

Fig. 2. A three-module robot reached the desired final tip position with different shapes. We incrementally generalize the obtainable shape: Arc-like using Consensus, S-shape with Bipartite Consensus, and any feasible shape with Formation Control. The variable k represents the curvature of the i^{th} module, and the star represents the desired tip position.

This control framework is scalable and does not depend on the number of agents. The modular kinematic chain can be expanded or reduced without altering the agents' control law. Moreover, the agents can be different: the only aspect of the model considered is the differential kinematics and the dynamic model of the associated module. It should be underlined that the model in the control law refers only to one module, strongly reducing the computational cost of a single control law, as shown in detail in Section IV-G.

In Fig. 2, an example of the final configuration of the soft arm is presented, with generalization and softening of the constraints moving from left to right. Indeed, with Consensus, each module is coplanar and reaches the same curvature. With Bipartite Consensus, the coplanarity constraint is removed, i.e., ϕ and k variables, resulting in adjacent modules having equal magnitudes but opposite signs. In Formation Control, we obtain the generalization of the shape control strategy. The only requirement is the feasibility of the shape given the desired tip position.

B. Communicated Information Reduction

Simplifying the control law in a distributed control framework shifts complexity to communication, which depends on network topology and message minimization. A global fixed frame is commonly used for distributed control, though it is not the only solution. Adopting a strategy without a global frame [26], [33] reduces communication and simplifies the physical interpretation of relative positions. This is possible because the CC model assures, given a robot tip pose, a single configuration vector, excluding singular configurations. This approach eliminates the need for rotation matrices, with positions expressed in local frames, reducing the number of shared messages among agents. With a model limited to elongation and curvature, the strategy ensures agreement on tip positions and configuration vectors. This allows us to recast the formulation within the framework of a SC task, extending beyond the previously introduced TPR approach.

C. Consensus Control for Arc-Like Shape With TPR

Building upon the work presented in [21], we extend the consensus formulation to a 3D framework for modular continuum soft robots, and generalize it to include additional strain degrees of freedom, such as the stretch component. Moreover, we adopt

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

the communication reduction strategy described in Section IV-B to achieve agreement directly on the configuration vector. This enables us to reinterpret the formulation in the context of a SC task, going beyond the previously presented TPR approach.

We start from the differential kinematics $\boldsymbol{J}_i\dot{\boldsymbol{q}}_i = (\dot{\boldsymbol{T}}_i\boldsymbol{T}_i^{-1})^\vee$, where $(\cdot)^\vee$ is the *vee* operator [31]. To determine the linear velocity $\dot{\boldsymbol{p}}_i$ of the tip of the ith segment expressed in the local base frame $\{S_{i-1}\}$, we pass through the explicit expression of $\dot{\boldsymbol{T}}_i$, inverting the previous equation. Hence, the $\dot{\boldsymbol{p}}_i$ expression [11] is equal to

$$\dot{\boldsymbol{p}}_i = \begin{bmatrix}\boldsymbol{I}_{3\times3} & -\hat{\boldsymbol{p}}_i\end{bmatrix}\begin{bmatrix}\boldsymbol{J}_{v,i}\\ \boldsymbol{J}_{\omega,i}\end{bmatrix}\dot{\boldsymbol{q}}_i = \boldsymbol{G}_i\begin{bmatrix}\boldsymbol{J}_{v,i}\\ \boldsymbol{J}_{\omega,i}\end{bmatrix}\dot{\boldsymbol{q}}_i = \boldsymbol{J}_{p,i}\,\dot{\boldsymbol{q}}_i\,. \tag{3}$$

where $\boldsymbol{J}_{p,i} \in \mathbb{R}^{3\times3}$ is the Jacobian on which the proposed kinematic control laws are based. The expression of the i^{th} controller, based on the Consensus strategy, is equal to

$$\dot{\boldsymbol{q}}_i = \boldsymbol{J}_{p,i}^{\dagger}\left[-\sum_{j\in\mathcal{N}_i}(\boldsymbol{p}_i - \boldsymbol{p}_j) + \boldsymbol{\mathcal{K}}_i\boldsymbol{E}\right], \tag{4}$$

where $\boldsymbol{J}_{p,i}^{\dagger}$ denote the Moore–Penrose pseudo-inverse of the Jacobian, and $\mathcal{N}_i$ is the set of neighbors of the i^{th} module. Moreover, $\boldsymbol{p}_i$ represents the position of the i^{th} agent expressed in its base local frame. We define the tip position error $\boldsymbol{E} = \boldsymbol{p}_N^0 - \boldsymbol{p}_{\text{d}}^0$, and the control gain $\boldsymbol{\mathcal{K}}_i = -\boldsymbol{R}_0^{N-1}\boldsymbol{K}_p$, if $i = N$, and $\boldsymbol{0}_{3\times3}$, otherwise. $\boldsymbol{R}_0^{N-1}$ is the rotation matrix from the world frame $\{S_0\}$ to last module base frame $\{S_{N-1}\}$, and $\boldsymbol{K}_p$ is the control gain. The control law for the Nth agent includes a proportional term to correct the tip position error. Starting from [21], we introduce the following proposition.

Proposition IV.1: Let $\boldsymbol{p}_i$ denote the local position vector of segment i^{th} in its local base frame. The distributed control law in (4) establishes a consensus protocol for the vectors $\boldsymbol{p}_i$, with an input term defined by the error $\boldsymbol{E}$ between the robot's tip and the target position.

Proof: Starting from the differential kinematic definition in (3), we write the i^{th} velocity vector in $\{S_{i-1}\}$ frame as

$$\dot{\boldsymbol{p}}_i = \boldsymbol{G}_i\boldsymbol{J}_i\,\dot{\boldsymbol{q}}_i = \boldsymbol{J}_{p,i}\,\dot{\boldsymbol{q}}_i\,, \tag{5}$$

Now, we substitute the control law defined in (4), in (5) to obtain

$$\dot{\boldsymbol{p}}_i = -\sum_{j\in\mathcal{N}_i}(\boldsymbol{p}_i - \boldsymbol{p}_j) + \boldsymbol{\mathcal{K}}_i\boldsymbol{E}, \tag{6}$$

where it has been used the identity $\boldsymbol{J}_{p,i}\boldsymbol{J}_{p,i}^{\dagger} = \boldsymbol{I}_{3\times3}$. To achieve a compact matrix formulation, it is necessary to define the collection $\boldsymbol{P} := \begin{bmatrix}(\boldsymbol{p}_1)^\top & (\boldsymbol{p}_2)^\top & \dots & (\boldsymbol{p}_N)^\top\end{bmatrix}^\top$, and $\boldsymbol{\mathcal{K}} := \begin{bmatrix}\boldsymbol{0}_{3\times3}^\top & \boldsymbol{0}_{3\times3}^\top & \dots & \boldsymbol{\mathcal{K}}_N^\top\end{bmatrix}^\top$. Now, it is possible to write the compact expression in matrix form as

$$\dot{\boldsymbol{P}} = -\boldsymbol{\mathcal{L}}\boldsymbol{P} + \boldsymbol{\mathcal{K}}\boldsymbol{E}\,, \tag{7}$$

where it has been defined $\boldsymbol{\mathcal{L}} = \boldsymbol{L}\otimes\boldsymbol{I}_{3\times3}$, where $\boldsymbol{L} \in \mathbb{R}^{N\times N}$ is a Laplacian matrix of the graph associated with the communication network of the robot's segments, and $\otimes$ is the Kronecker product. It is a linear consensus system driven by signals $\boldsymbol{E}$. □

With the Consensus without the global frame, the magnitudes of all the position vectors $\boldsymbol{p}_i$ converge to the agreement value. Furthermore, considering the CC model, the agreement is reached also in terms of the configuration vector. This observation ensures the convergence to the Arc-like Shape of the controlled distributed system. An example of the obtained final configuration is shown on the left side of Fig. 2.

Proposition IV.2: The distributed control law (4) ensures that the position of the robot tip $\boldsymbol{p}_N^0$ globally exponentially converges to the target position $\boldsymbol{p}_{\text{d}}^0$.

Proof: Starting from (7), we extract the last row of the dynamics to explicitly write the expression of the $\dot{\boldsymbol{p}}_N$

$$\dot{\boldsymbol{p}}_N = -(\boldsymbol{p}_N - \boldsymbol{p}_{N-1}) + \boldsymbol{\mathcal{K}}_N\boldsymbol{E}\,. \tag{8}$$

Obtaining consensus between the modules, we delete the first hand-right term; while remembering the definition of $\boldsymbol{\mathcal{K}}_N$, the resulting equation is $\dot{\boldsymbol{p}}_N^0 = -\boldsymbol{K}_p\boldsymbol{E}$. Given a fixed target, i.e., $\dot{\boldsymbol{p}}_{\text{d}}^0 = \boldsymbol{0}$, we rewrite the previous equation as $\dot{\boldsymbol{E}} + \boldsymbol{K}_p\boldsymbol{E} = \boldsymbol{0}$, which is exponentially stable to the equilibrium ($\boldsymbol{E} = \boldsymbol{0}$) for any positive gains. □

D. Bipartite Consensus Control for S-Shape With TPR

Starting from the formulation presented in [28], we generalize the Consensus strategy with the Bipartite Consensus. It allows for achieving an alternative feasible shape of the soft arm, specifically, the *S-shape*, characterized by a serpentine curve. This occurs due to a form of disagreement among agents, explicitly referred to as an *'agreed dissensus'*, obtained through the generalization of classic graph theory: signed graph [29]. In terms of control, it represents an extension of the previous strategy obtained through the assignment of negative weights to the graph edges. This is possible because the graph is connected and structurally balanced [28]. Hence, considering the state of the network as the local position of the i^{th} agent and the choice of the frames, each weight of the network is equal to $w_{ij} = \begin{bmatrix}1 & -1 & 1\end{bmatrix}^\top$. It is possible to write the control law

$$\dot{\boldsymbol{q}}_i = \boldsymbol{J}_{p,i}^{\dagger}\left[-\sum_{j\in\mathcal{N}_i}(\boldsymbol{p}_i - \boldsymbol{w}_{ij}\odot\boldsymbol{p}_j) + \boldsymbol{\mathcal{K}}_i\boldsymbol{E}\right], \tag{9}$$

where the $\odot$ operator represents the product element by element. The structurally balanced condition allows us to conclude that the controlled system with the Bipartite Consensus strategy has the same spectral properties [28] as the controlled system with the Consensus strategy, and then, the tip position error converges exponentially.

In a network of three agents, agent 2 has the same magnitude of k and ϕ as agents 1 and 3, but with the opposite sign in the final configuration, as shown in the middle of Fig. 2. This highlights that the system's constraints are relaxed with this strategy, removing the coplanarity constraint, and allowing the modules to achieve an opposite curvature and form the desired S-shape.

E. Formation Control Shape Control Under TPR

In this section, we present the main contribution of our work. Building on the results from [6], and leveraging the control framework introduced in Sections IV-A and IV-B, we extend the Consensus approach by adopting a Formation Control strategy. This strategy enables us to perform SC under the sole assumption that the desired shape is feasible for the target tip position. Generalization is achieved at the expense of a large amount of

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

requested information. The task is described in terms of the final tip position and the relative position between the neighboring agents. The desired shape must belong to the feasible set of arm shapes, maintaining the desired tip position, defining the set of translational and rotational invariant formations [6]. The formation control law obtained is equal to

$$\dot{\boldsymbol{q}}_i = \boldsymbol{J}_{p,i}^{\dagger} \left[-\sum_{j \in \mathcal{N}_i} \left[(\boldsymbol{p}_i - \boldsymbol{p}_j) - \boldsymbol{z}_{ij,\text{ref}} \right] + \boldsymbol{\mathcal{K}}_i \boldsymbol{E} \right], \tag{10}$$

where $\boldsymbol{z}_{ij,\text{ref}} \in \mathbb{R}^3$ is the desired relative configuration between two neighbor agents defined in the local frame. The desired relative configurations define the constraints in terms of shape and can be defined as $\boldsymbol{z}_{ij,\text{ref}} = \boldsymbol{p}_{i,\text{ref}} - \boldsymbol{p}_{j,\text{ref}}$.

Theorem IV.1: The distributed control law in (10) establishes a formation control strategy with an input term defined by the error $\boldsymbol{E}$ between the robot's tip and the target position, assuring the asymptotic convergence to a translational invariant desired shape.

Proof: Starting from (3), we write the i^{th} velocity vector in the local frame as in (5). Substituting the control law defined in (10), we obtain

$$\dot{\boldsymbol{p}}_i = -\sum_{j \in \mathcal{N}_i} \left[(\boldsymbol{p}_i - \boldsymbol{p}_j) - \boldsymbol{z}_{ij,\text{ref}} \right] + \boldsymbol{\mathcal{K}}_i \boldsymbol{E}, \tag{11}$$

where we used the identity $\boldsymbol{J}_{p,i} \boldsymbol{J}_{p,i}^{\dagger} = \boldsymbol{I}_{3\times 3}$. It is necessary to use the previously defined vectors $\boldsymbol{P}$ and $\boldsymbol{\mathcal{K}}$, with the addition of $\boldsymbol{Z}_{\text{ref}}$, the collection of relative position vectors. Now, it is possible to write a compact expression in matrix form:

$$\dot{\boldsymbol{P}} = -\boldsymbol{\mathcal{L}} \boldsymbol{P} + \boldsymbol{\mathcal{D}} \boldsymbol{Z}_{\text{ref}} + \boldsymbol{\mathcal{K}} \boldsymbol{E}\,, \tag{12}$$

where it has just been defined $\boldsymbol{\mathcal{L}}$. Likewise, $\boldsymbol{\mathcal{D}} = \boldsymbol{D} \otimes \boldsymbol{I}_{3\times 3}$, where $\boldsymbol{D}$ is the incidence matrix of the directed graph network associated with the relative configuration definition. Compared to the traditional definition of formation control [6], we conclude that it is a linear formation control strategy driven by signals $\boldsymbol{E}$. Considering (11), and recalling the definition of $\boldsymbol{z}_{ij,\text{ref}}$, we rewrite the term inside the sum as $\sum_{j\in\mathcal{N}_i}[(\boldsymbol{p}_i - \boldsymbol{p}_{i,\text{ref}}) - (\boldsymbol{p}_j - \boldsymbol{p}_{j,\text{ref}})]$. Now, defining the new variable $\boldsymbol{\gamma}_i = \boldsymbol{p}_i - \boldsymbol{p}_{i,\text{ref}}$, it is trivial that the term expressed in (11) is a consensus on the variable $\boldsymbol{\gamma}_i$. It can be write as $\boldsymbol{\mathcal{L}}\boldsymbol{\Gamma}$, where $\boldsymbol{\Gamma}$ is the collection of $\boldsymbol{\gamma}_i$ for $i = 1, \ldots, N$. It is possible to conclude that the collaborative term has zero when the consensus is reached, obtaining the desired shape of the modular continuum soft arm. □

Theorem IV.2: Assuming that the desired shape is feasible with respect to the target tip position, the distributed control (10) law ensures that the position of the robot tip $\boldsymbol{p}_N^0$ globally exponentially converges to the target position $\boldsymbol{p}_{\text{d}}^0$.

Proof: To demonstrate the convergence of the tip position error, we start from (12), extracting the last row of the dynamics to explicitly write the expression of the $\dot{\boldsymbol{p}}_N$

$$\dot{\boldsymbol{p}}_N = -(\boldsymbol{\gamma}_N - \boldsymbol{\gamma}_{N-1}) + \boldsymbol{\mathcal{K}}_N \boldsymbol{E}\,. \tag{13}$$

Obtaining the desired translation invariant formation between the modules, we delete the first hand-right term; while remembering the definition of $\boldsymbol{\mathcal{K}}_N$, the resulting equation is $\dot{\boldsymbol{p}}_N^0 = -\boldsymbol{K}_P \boldsymbol{E}$. Given a fixed target, i.e., $\dot{\boldsymbol{p}}_{\text{d}}^0 = \boldsymbol{0}$, we rewrite the previous equation as $\dot{\boldsymbol{E}} + \boldsymbol{K}_p \boldsymbol{E} = \boldsymbol{0}$, which is exponentially stable to the equilibrium ($\boldsymbol{E} = \boldsymbol{0}$) for any positive gains. □

F. Decentralized Curvature-Based Dynamic Controller

Kinematic control does not account for dynamic behaviors like gravity and module interactions, requiring a low-level dynamic controller. Thus, a dynamic model for each CC segment is used, and a decentralized controller rejects disturbances, such as unmodeled nonlinear interactions between agents. The decentralized approach can be adopted [34] under two main assumptions: i) each module is equipped with an independent actuation system, and ii) the configuration vector exhibits low velocities and accelerations, i.e., $\dot{q}$ and $\ddot{q}$. The first assumption holds due to the self-contained nature of the actuation system in each module along the chain [5]. The second assumption is justified by the fact that soft robots typically operate with lower bandwidth during quasi-static tasks, mainly due to their inherently lower stiffness and damping compared to rigid systems. This latter condition supports the hypothesis that the dominant coupling effect arises from the gravitational forces exerted by the downstream modules. These disturbances change slowly, and they can be treated as step disturbances. To mitigate them, integral terms are added to the control law, effectively nullifying errors. The decentralized curvature-based dynamic control law is

$$\boldsymbol{\tau}_i = \boldsymbol{A}^{-1}(\bar{\boldsymbol{q}}_i) \left[\boldsymbol{K}(\bar{\boldsymbol{q}}_i) + \boldsymbol{G}(\bar{\boldsymbol{q}}_i) + f_{\text{PID}}(\boldsymbol{e}_{q,i}, \dot{\boldsymbol{e}}_{q,i}) \right], \tag{14}$$

where $f_{\text{PID}}(\boldsymbol{e}_{q,i}, \dot{\boldsymbol{e}}_{q,i}) = \boldsymbol{\mathcal{K}}_P \boldsymbol{e}_{q,i} + \boldsymbol{\mathcal{K}}_D \dot{\boldsymbol{e}}_{q,i} + \Gamma \boldsymbol{\mathcal{K}}_I \int_0^t \boldsymbol{e}_{q,i} \text{d}t$. The term $\bar{\boldsymbol{q}}_i$ is the desired configuration of the module i^{th}, and $\boldsymbol{e}_{q,i} = \bar{\boldsymbol{q}}_i - \boldsymbol{q}_i$ is the configuraton vector error. The terms $\boldsymbol{K}(\bar{\boldsymbol{q}}_i)$ and $\boldsymbol{G}(\bar{\boldsymbol{q}}_i)$ are, respectively, the stiffness and the gravitational compensation term. The matrix $\boldsymbol{A}(\bar{\boldsymbol{q}}_i)$ is the actuation matrix. Furthermore, $\boldsymbol{\mathcal{K}}_P$, $\boldsymbol{\mathcal{K}}_D$, and $\boldsymbol{\mathcal{K}}_I$ are the positive defined control gain matrices. The term Γ is equal to $N - i$ for each agent. The idea is to apply a larger integral gain to the first module, gradually decreasing for subsequent modules since the disturbance from the weight of downstream modules is greater for those closer to the base of the arm. In [4], it is shown the control law, in a centralized case, without the integral term, has the same structure as a classic robot, controlled by a nonlinear proportional-derivative regulator. Adding an integral term preserves stability if the gain is sufficiently low. This approach does not explicitly compensate for unmodeled nonlinear interactions between agents but allows the error to converge to zero due to the integral term.

G. Computational Complexity Analysis

This section highlights the theoretical computational complexity of our control framework compared to the centralized approach. We focus on the number of basic operations required to compute the control input: vector summation, matrix-vector multiplication, and matrix inversion via Singular Value Decomposition (SVD). The variables m, n, and N represent the number of controlled spatial coordinates, the number of DoFs, and the number of modules, respectively. For kinematic control, we compare the reduced communication-based framework (Section IV-B) with the classic centralized law [11], neglecting the invariant terms. The computational complexity for the centralized kinematic control is $\mathcal{O}(\min\{m(nN)^2, m^2nN\}) + \mathcal{O}(mnN)$, where the first term is the Jacobian pseudo-inversion, and the second term is matrix-vector products. For the distributed, the complexity is $\mathcal{O}(\min\{mn^2, m^2n\}) + \mathcal{O}(mn) +$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

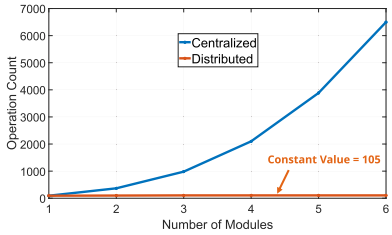

Fig. 3. A comparison of computational complexity between the proposed and centralized control. The y-axis is the operation count, i.e., the number of basic operations. The analysis considers an increasing number of modules N, with $m = 3$ and $n = 3$.

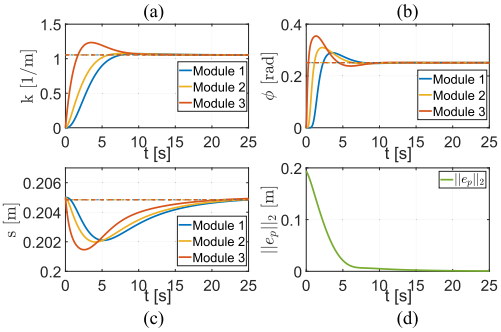

Fig. 4. Arc-like shape control using the Consensus. The results show the convergence of the configuration vector to the same value for three state components (a), (b), and (c) and the norm of the tip position error (d).

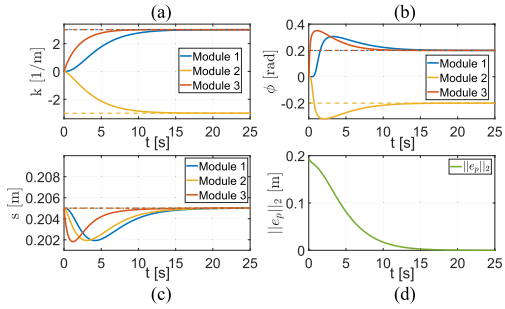

Fig. 5. S-shape control using the Bipartite Consensus. The results show the convergence to the agreed dissensus behavior (a), (b), and (c) and the norm of the tip position error (d).

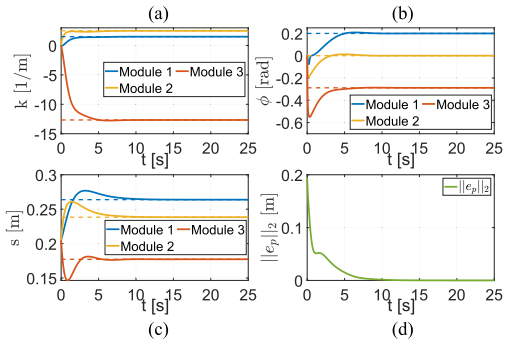

Fig. 6. Shape control using the Formation Control strategy. The results show the convergence to the desired configuration vectors (a), (b), and (c) and the norm of the tip position error (d).

$\mathcal{O}(2, m)\gamma$, where the additional term accounts for the collaborative component, with $\gamma = 0$ for a single module arm, 1 for two modules, and 2 for more. For dynamic control, the centralized complexity is $\mathcal{O}(bnN) + \mathcal{O}(nN(nN)^2) + \mathcal{O}((nN)^2)$, where the terms correspond to vector sums, pseudo-inversion, and multiplication of resulting terms. In our case $b = 7$. In the decentralized case, the complexity is the same but without dependence on N.

Our scalable framework maintains constant complexity (equal to 105) as N increases, while the centralized approach has linear and quadratic contributions for kinematic and dynamic control, respectively. The results are shown in Fig. 3.

V. Simulation Results

A. Numerical Validation

These results are significant for validating the theoretical tools and discussing performance. Figs. 4, 5, and 6 show the results using Consensus, Bipartite Consensus, and Formation Control, respectively. The three control strategies progressively enable the robot to achieve increasingly complex shapes - Arc-like, S-shape, and more generally, any feasible configuration - highlighting improvements in generalization through incremental experiments. For this reason, the desired tip position remains the same across all strategies, while the resulting shape varies, and is given by $\begin{bmatrix}0.0723, & -0.175, & 0.574\end{bmatrix}^\top$. For the Consensus, we obtain an Arc-like shape through agreement among agents in terms of configuration vectors and convergence to the desired tip position, as shown in Fig. 4. Interestingly, the modules follow a similar path, with the first and the second following the leader, i.e., the third module, with a certain delay. To minimize this type of behavior, which can lead to an overshoot, the idea is to maintain a low value of the proportional gain, equal to 0.20, maintaining the balance between the agreement and the proportional terms. The gain is tuned considering maintaining agreement, measured by the Laplacian potential [6]. This ensures a lower overshoot and a good convergence time.

For the Bipartite Consensus, we show the results in Fig. 5. Therefore, the final configuration vectors make the previously described *agreed dissensus* behavior clear for the adjacent modules. The variables k and ϕ reach equal absolute values but with opposite signs. This results in a greater required curvature compared to the Consensus Strategy. Fig. 5(d) shows the convergence to the desired tip position.

Fig. 6 shows the Formation Control results with the convergence of the configuration vectors and the tip position. This strategy allows for the performance of more general and complex shapes, thanks to the addition of relative positions among modules. In this example, they are equal to $z_{12,\text{ref}} = [-0.043, \quad -0.001, \quad 0.044]^\top$ and $z_{23,\text{ref}} = [0.036, \quad -0.257, \quad 0.327]^\top$.

B. Simulations

This section shows the result obtained with the SoRoSim MATLAB toolbox [8]. This second validation is also performed to validate the dynamic control shown in Section IV-F with the path of the configuration vector planned by the kinematic controllers shown in Section IV-C, IV-D, and IV-E. The parameters of each module are: $L = 0.205$ m, $D = 0.06$ m, $E = 10$ MPa,

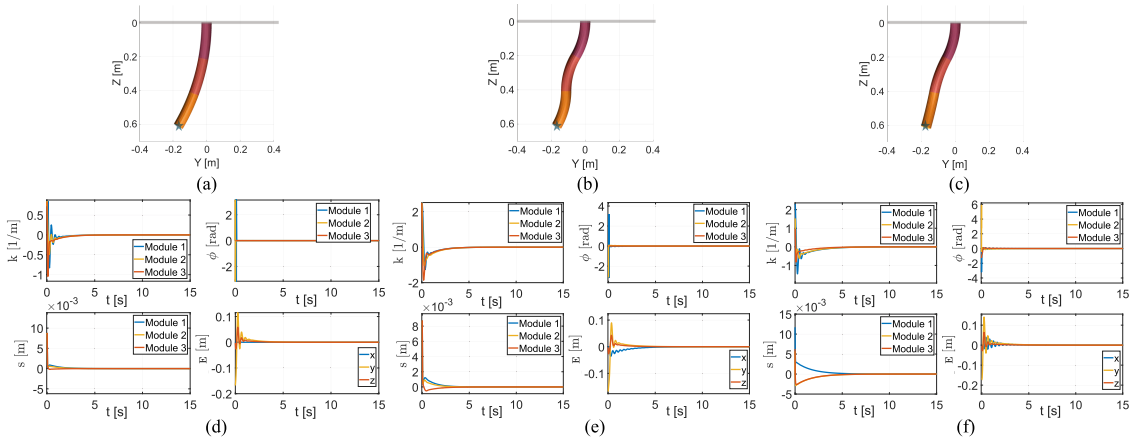

Fig. 7. Three-module robot simulations in SoRoSim with desired tip position equal to $[0.0, \quad -0.168, \quad 0.611]^\top$. Figures (a), (b), and (c) show the final shape for Consensus, Bipartite Consensus, and Formation Control, respectively, with the same final tip position. The figures (d), (e), and (f), show the shape error in terms of the configuration vector and the tip position error.

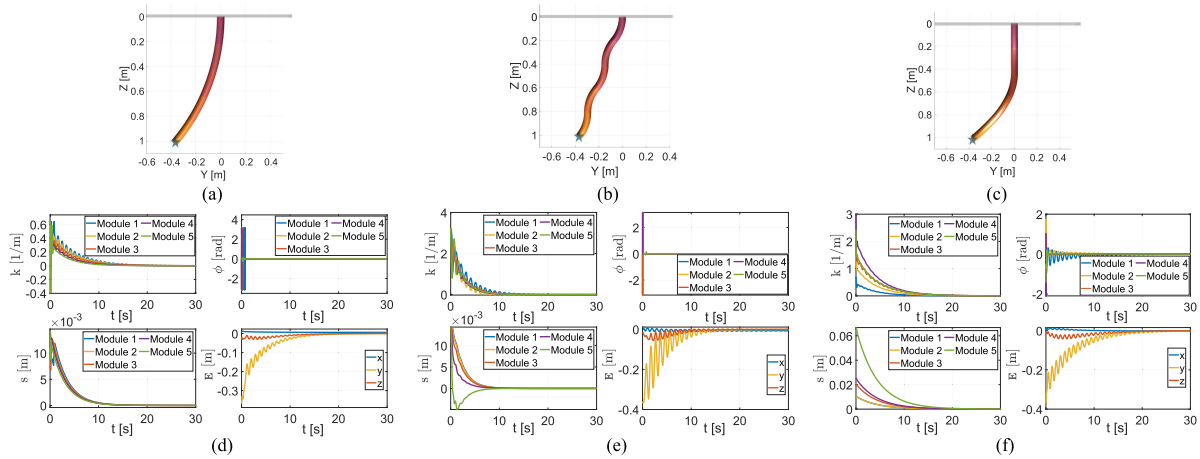

Fig. 8. Five-module robot simulations in SoRoSim with desired tip position equal to $[0.011, \quad -0.370, \quad 1.008]^\top$. Figures (a), (b), and (c) show the final shape for Consensus, Bipartite Consensus, and Formation Control, respectively, with the same final tip position. The figures (d), (e), and (f), show the shape error in terms of the configuration vector and the tip position error.

$\rho = 1100$ kg/m^3, and $\eta = 0.05$MPa $\cdot$ s. They represent, respectively, the length of the module, the diameter of the cross section, the Young modulus, the density, and the damping coefficient. The gain matrices of the dynamic controller are chosen such that their diagonal elements are of the same order of magnitude as those of the stiffness matrix. The simulation environment is based on a different model [8] - namely, the Geometric Variable Strain (GVS) approach - compared to the one used in our model-based controller. It considers the interaction among modules and includes inertial and Coriolis terms, which our controller assumes to be unmodeled dynamics and aims to reject as disturbances. This aspect makes the validation of the dynamic controller more significant. Moreover, the idea is to validate the control framework for different soft arm setups, to analyze how dynamic control rejects unmodeled interactions as they increase. The validation is performed with two different robot sets: 3-module and 5-module robots, both with identical modules in terms of physical parameters.

For the former, the final configurations are shown in Fig. 7(a), (b), and (c). The convergence to the desired shape is present in Fig. 7, (d), (e), and (f), respectively, for Consensus, Bipartite Consensus, and Formation Control. All the performed tasks underline the shape's convergence to the desired one. The adopted controllers enable rapid convergence to the desired target while effectively rejecting unmodeled dynamics.

For 5-module robots, the oscillations increase, as expected. With this setup, the interaction force between modules increases in amplitude, leading to greater oscillations. Moreover, as established in classical beam theory [35], the oscillation is inversely proportional to the square of the beam length. This results in a longer time required for convergence. Fig. 8(a), (b), and (c) show the final configuration for the three strategies, while Fig. 8(d), (e), and (f) illustrate the convergence to the desired shape and tip position. The system achieved good performance, converging to the desired behavior even under this setup. This highlights that the dynamic interactions between modules, though not explicitly known to the controller, are effectively compensated for by the integral term of the control.

VI. Conclusion and Future Works

In this work, we present a novel, scalable control framework for modular continuum soft arms. This framework enables the creation of self-contained modules, each governed by a collaborative controller to achieve the desired shape. Each controller relies on a minimal model, such as the Constant Curvature assumption, for its own module, while integrating local position information shared by neighboring modules. We presented three kinematic control strategies - Consensus, Bipartite Consensus, and Formation Control - to achieve more complex and adaptable shapes while reaching the same tip position. The control is performed in three-dimensional space and relies solely on the relative local positions communicated among agents. Additionally, we present a decentralized curvature-based dynamic controller to manage the unmodeled nonlinear dynamic coupling among modules. In conclusion, we validated the proposed control framework through extensive numerical simulations.

Future investigations will focus on extending the proposed framework to incorporate distributed dynamic control and to generalize the treatment to models with additional strain modes, such as the GVS model. Finally, experimental validation will be carried out to demonstrate the effectiveness of the approach in real-world scenarios.

References

[1] C. D. Santina et al., "Soft robots," *Encyclopedia Robot.*, vol. 489, pp. 1–14, 2020.

[2] T. George Thuruthel et al., "Control strategies for soft robotic manipulators: A survey," *Soft Robot.*, vol. 5, no. 2, pp. 149–163, 2018.

[3] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1728–1748, Jun. 2023.

[4] C. D. Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Syst. Mag.*, vol. 43, no. 3, pp. 30–65, Jun. 2023.

[5] C. Zhang et al., "Modular soft robotics: Modular units, connection mechanisms, and applications," *Adv. Intell. Syst.*, vol. 2, no. 6, 2020, Art. no. 1900166.

[6] M. Mesbahi, *Graph Theoretic Methods in Multiagent Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2010.

[7] R. J. Webster III et al., "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, 2010.

[8] A. T. Mathew, I. B. Hmida, C. Armanini, F. Boyer, and F. Renda, "SoRoSim: A MATLAB toolbox for hybrid rigid–soft robots based on the geometric variable-strain approach," *IEEE Robot. Autom. Mag.*, vol. 30, no. 3, pp. 106–122, Sep. 2023.

[9] E. Falotico et al., "Learning controllers for continuum soft manipulators: Impact of modeling and looming challenges," *Adv. Intell. Syst.*, vol. 7, 2024, Art. no. 2400344.

[10] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 935–949, Aug. 2014.

[11] R. J. Webster et al., "Closed-form differential kinematics for concentric-tube continuum robots with application to visual servoing," in *Proc. Exp. Robot.: 11th Int. Symp.*, 2009, pp. 485–494.

[12] A. S. Lafmejani et al., "Kinematic modeling and trajectory tracking control of an octopus-inspired hyper-redundant robot," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3460–3467, Apr. 2020.

[13] F. Renda, C. Armanini, A. Mathew, and F. Boyer, "Geometrically-exact inverse kinematic control of soft manipulators with general threadlike actuators' routing," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7311–7318, Jul. 2022.

[14] J. Zhu, H. Wang, W. Chen, and L. Xie, "The three-dimensional shape control for a soft robot," in *Proc. 13th IEEE Int. Conf. Control Automat.*, 2017, pp. 385–390.

[15] R. K. Katzschmann, C. D. Santina, Y. Toshimitsu, A. Bicchi, and D. Rus, "Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model," in *Proc. 2nd IEEE Int. Conf. Soft Robot.*, 2019, pp. 454–461.

[16] D. Papageorgiou et al., "Sliding-mode control of a soft robot based on data-driven sparse identification," *Control Eng. Pract.*, vol. 144, 2024, Art. no. 105836.

[17] X. Shao et al., "Model-based control for soft robots with system uncertainties and input saturation," *IEEE Trans. Ind. Electron.*, vol. 71, no. 7, pp. 7435–7444, Jul. 2024.

[18] P. Pustina, P. Borja, C. D. Santina, and A. De Luca, "P-satI-D shape regulation of soft robots," *IEEE Robot. Autom. Lett.*, vol. 8, no. 1, pp. 1–8, Jan. 2023.

[19] F. Renda, A. Mathew, and D. F. Talegon, "Dynamics and control of soft robots with implicit strain parametrization," *IEEE Robot. Autom. Lett.*, vol. 9, no. 3, pp. 2782–2789, Mar. 2024.

[20] C. N. Cho et al., "A modular control scheme for hyper-redundant robots," *Int. J. Adv. Robotic Syst.*, vol. 12, no. 7, 2015, Art. no. 91.

[21] A. S. Lafmejani, H. Farivarnejad, A. Doroudchi, and S. Berman, "A consensus strategy for decentralized kinematic control of multi-segment soft continuum robots," in *Proc. 2020 Amer. Control Conf.*, 2020, pp. 909–916.

[22] A. Doroudchi et al., "Decentralized control of distributed actuation in a segmented soft robot arm," in *Proc. 2018 IEEE Conf. Decis. Control*, 2018, pp. 7002–7009.

[23] A. Doroudchi et al., "Configuration tracking for soft continuum robotic arms using inverse dynamic control of a Cosserat rod model," in *Proc. IEEE 4th Int. Conf. Soft Robot. (RoboSoft)*, 2021, pp. 207–214.

[24] J. Hughes et al., "Sensing soft robot shape using imus: An experimental investigation," in *Proc. Exp. Robot.: 17th Int. Symp.*, 2021, pp. 543–552.

[25] G. Pei et al., "Plant-inspired decentralized controller for robust orientation control of soft robotic manipulators," *Bioinspiration Biomimetics*, vol. 20, 2025, Art. no. 026019.

[26] Z. Yang et al., "Leader-follower formation control of nonholonomic mobile robots with bearing-only measurements," *J. Franklin Inst.*, vol. 357, no. 3, pp. 1628–1643, 2020.

[27] J. A. Bondy et al., *Graph Theory*. Berlin, Germany: Springer Publishing Company, Incorporated, 2008.

[28] C. Altafini, "Achieving consensus on networks with antagonistic interactions," in *Proc. IEEE 51st IEEE Conf. Decis. Control*, 2012, pp. 1966–1971.

[29] Y. Hou et al., "On the Laplacian eigenvalues of signed graphs," *Linear Multilinear Algebra*, vol. 51, no. 1, pp. 21–30, 2003.

[30] D. Caradonna, M. Pierallini, C. D. Santina, F. Angelini, and A. Bicchi, "Model and control of R-soft inverted pendulum," *IEEE Robot. Autom. Lett.*, vol. 9, no. 6, pp. 5102–5109, Jun. 2024.

[31] R. M. Murray et al., *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 2017.

[32] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 43–55, Feb. 2006.

[33] E. Montijano, D. Zhou, M. Schwager, and C. Sagues, "Distributed formation control without a global reference frame," in *Proc. 2014 Amer. Control Conf.*, 2014, pp. 3862–3867.

[34] L. Sciavicco et al., *Modelling and Control of Robot Manipulators*. Berlin, Germany: Springer Science & Business Media, 2012.

[35] W. Weaver Jr. et al., *Vibration Problems in Engineering*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 1991.