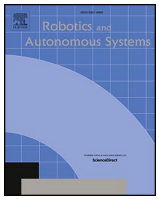
A Safe, high-precision reinforcement learning-based optimal control of surgical continuum robots: A monotone tube boundary approach with prescribed-time control capability

Robotics and Autonomous Systems 190 (2025) 104992

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



A Safe, high-precision reinforcement learning-based optimal control of surgical continuum robots: A monotone tube boundary approach with prescribed-time control capability

Mohammad Jabari [a,b,1,*], Andrea Botta [b], Luigi Tagliavini [b], Carmen Visconte [b], Giuseppe Quaglia [b]

[a] Department of Informatics, Bioengineering, Robotics and Systems Engineering (DIBRIS), Università di Genova, Genova, Italy
[b] Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

ARTICLE INFO

Keywords:
Surgical Continuum Robot
Safe Operation
Precise Control
Monotone Tube Boundary
Prescribed-time Control
Reinforcement Learning

ABSTRACT

This paper introduces a novel approach to the prescribed-time control of continuum surgical robots, focusing on four key areas: enhanced system safety, tailored transient tracking, steady-state tracking enhancement, and optimal learned control. The main contribution is the application of system state constraints on tracking error, transforming these constraints into an unconstrained problem using a monotone tube boundary. This method avoids the complexity of Model Predictive Control (MPC) and Control Barrier Functions (CBF) techniques, as well as the conservatism and fixed-boundary issues associated with the Barrier Lyapunov Function (BLF) method. By using a monotone tube boundary, the approach allows for the pre-assignment of transient characteristics for tracking error, avoiding excessive overshoot and lack of adjustability seen with the Prescribed Performance Function (PPF). The prescribed-time control philosophy enables pre-determination of settling time, enhancing precision and convergence rates essential for surgical applications. Additionally, an optimized prescribed-time control strategy using an actor-critic neural network-based Reinforcement Learning (RL) approach ensures controller optimality, reducing control effort, power consumption, and heat generation in the robot's actuators. The method adapts to dynamic environments, ensuring robust performance in various surgical scenarios. Simulation results on a two-segment continuum robot demonstrate the proposed method's advantages over state-of-the-art techniques.

1. Introduction

The influence of control science spans various engineering fields such as robotics, aerospace, electric vehicles, and traffic management. In medicine, it has driven significant advancements in surgical robots, rehabilitation devices, wearable sensors, and breathing aids [1]. While medical science has improved diagnostic capabilities with tools like X-rays, CT scans, and various blood tests, treatment challenges remain, particularly for surgeries in hard-to-reach areas. For instance, aspirating blood or infectious material from intracavities with small incision ports (5-15 mm) presents a challenge [2]. Natural Orifice Transluminal Endoscopic Surgery (NOTES) helps reduce surgical trauma by passing instruments through narrow, curved cavities to perform precise operations [3,4]. However, the blood suctioning procedure in NOTES still requires manual operation by a bedside assistant [6,7]. Researchers have turned to continuum robots to perform complex, minimally invasive procedures in unstructured surgical environments due to their adaptability, safety, and dexterity [8,9]. Ensuring the precision and reliability of these robots is crucial for patient safety, necessitating advanced control methods to maintain peak performance amidst disturbances and uncertainties.

In recent years, many controllers for continuum robots have been designed, with PID controls being among the first due to their simplicity and efficiency. For example, in [10] and [11], the PID controller is utilized for continuum robot control; [10] employs the Grey Wolf Optimizer (GWO), and [11] uses fuzzy logic to fine-tune the PID parameters. The literature review reveals that the Model Predictive Control (MPC) method is also widely used in robotic systems, similar to the

* Corresponding author.
E-mail address: mohammad.jabari@polito.it (M. Jabari).
[1] Mohammad Jabari has been fully supported by the PNRR-NGEU project, which has received funding from the MUR – DM 351/2022.

https://doi.org/10.1016/j.robot.2025.104992
Received 7 December 2024; Received in revised form 11 March 2025; Accepted 17 March 2025
Available online 20 March 2025

PID controller. For instance, references [12] and [13] demonstrate effective angle control and deformation management in continuum robots via MPC. However, it is important to note that both PID and MPC are inherently suitable for linear systems and may not be highly accurate for nonlinear systems such as continuum robots, especially in the presence of uncertainties [14]. Therefore, for nonlinear systems, sliding mode controllers are often employed, which are also known for their robustness against uncertainties [15–18].

It should be noted, however, that classic sliding mode controllers are not suitable for surgical continuum robots for two primary reasons. Firstly, the complex kinematic and dynamic models of these robots, which arise from their structural flexibility, additional degrees of freedom, and nonlinear factors such as internal drive friction and backlash, make them more intricate than traditional robots [5,9]. Secondly, their low stiffness poses control challenges during precise position interactions with the environment [19]. Additionally, the chattering phenomenon inherent in sliding mode controllers can induce vibrations and significant transient errors [18,20,21], which is critical during surgery where there is a risk of damaging vital tissues and endangering human life.

To enhance the performance of sliding mode controllers in surgical robotics, the integration of finite-time, fixed-time, and prescribed performance methodologies is essential. These approaches significantly expedite goal attainment, and minimize overshoot, ripple, and steady-state error, thereby optimizing operational precision. However, each of these methods has advantages and disadvantages that are briefly mentioned here. In the intricate domain of control theory, the nuances distinguishing finite-time sliding mode controllers from fixed-time counterparts are pivotal, particularly in their convergence properties [22–24]. Finite-time controllers assure system stability within a time frame that is not only finite but also varies based on the system's initial state. In contrast, fixed-time controllers offer a guarantee of stability within a fixed upper bound on convergence duration, meaning that the upper bound of the settling time is notably independent on the system's initial conditions, thus facilitating a more predictable and swift path to convergence [25,26]. Diverging from these temporal-focused strategies, Prescribed Performance Sliding Mode Control (PPSMC) delineates a sophisticated framework wherein the system's output or error trajectory is meticulously guided to conform to a pre-established transient behavior, ensuring strict adherence to predefined transient and steady-state performance specifications [27,28]. However, in the delicate surgical robotics field, the prescribed performance method, while offering precise control over critical parameters such as overshoot and settling time, introduces risks associated with its less accurate control over the transient state, which may inadvertently result in damage to sensitive bodily tissues. To circumvent these potential hazards, this article advocates for a robust alternative known as the monotone tube approach [29]. This method is designed to mitigate the aforementioned risks and substantially enhance control precision. It is further augmented by Monotone Tube Sliding Mode Control (MTSMC), which employs strategically defined tube boundaries to reinforce PPSMC, thereby maintaining the system's performance within specified constraints despite the presence of disturbances or uncertainties. Utilizing the monotone tube method enables to accurately determine the settling time and overshoot, thereby resolving the challenging issue of safety in surgical robotics.

In this paper, the prescribed-time method within MTSMC is proposed to achieve an even higher level of safety [30]. Unlike finite-time control and fixed-time control, prescribed-time control is a method in control systems that guarantees stabilization within a preassigned time frame completely independent from the initial condition, which is crucial for tasks where timing is critical. Prescribed-time control is adaptable for various systems and ensures stability without inducing saturation or triggering unwanted dynamics. The primary rationale for employing the prescribed-time method in this research is to ensure that the tracking error converges to zero with remarkable speed and to render the system completely independent from the initial conditions, thereby achieving an unprecedented level of safety in surgical robotics.

In this article, the reinforcement learning (RL) method is also used in the controller to create different capabilities in the continuum surgical robot, one of the most important of which is to achieve optimal performance in unknown and different environments [31,32]. RL has emerged as a promising approach in this context, offering solutions for sequential decision-making problems in healthcare. By leveraging RL, continuum robots can optimize their actions based on feedback from the environment, improving their performance in dynamic and uncertain surgical settings. Studies have shown RL's potential in enhancing precision and adaptability in tasks such as object grasping and autonomous learning in robotics, as well as optimizing treatment regimes in chronic diseases and critical care. The essence of this technology lies in its ability to not only excel in routine, well-known scenarios but also to maintain high performance in novel and unpredictable settings. Consider, for instance, a continuum robot typically deployed for renal procedures; through the advancements discussed herein, such a robot can be exploited for cardiac surgeries or other body organs. This capability allows it to have good performance even in the presence of disturbances and unwanted uncertainties, such as pulmonary or blood pressure. Moreover, the system's efficiency is epitomized by its minimal energy demands during operations. At the heart of this efficiency is the Actor-critic reinforcement learning approach, which has been instrumental in formulating an optimal control law. This sophisticated strategy enables the controller to nullify tracking errors with minimal exertion of control forces. Achieving this objective necessitated the resolution of the Hamilton-Jacobi-Bellman (HJB) equation, a task adeptly accomplished through the reinforcement learning algorithm. The implications of solving the HJB equation are profound, as it paves the way for the robot to execute precise maneuvers with unparalleled accuracy and reliability.

In order to prove the efficacy of the proposed method, numerical simulations on a two-segment continuum robot, previously studied in [33] and [34], were carried out. Results were compared to those reported in [33], where an alternative control approach was proposed.

In summary, these contributions together emphasize the potential of the paper to significantly increase the accuracy, safety, and efficiency of continuum robot manipulators and pave the way for their wider application in complex and critical environments, which can be summarized as follows.

- **Enhanced System Safety:** The paper presents a monotone tube boundary to enforce system state constraints on tracking error, transforming them into an unconstrained problem. This method provides a more efficient alternative to the complexity of Model Predictive Control (MPC) and Control Barrier Functions (CBF), and avoids the conservatism and fixed-boundary issues of Barrier Lyapunov Functions (BLF).
- **Tailored Transient Tracking:** The monotone tube boundary enables precise pre-assignment of transient tracking characteristics, offering improved control over tracking behavior compared to Prescribed Performance Functions (PPF), which often result in excessive overshoot and limited adjustability.
- **Steady-State Tracking Enhancement:** The integration of a prescribed-time control approach ensures that settling time is predetermined and independent of initial conditions. This results in faster convergence and higher accuracy, which is critical for applications like robotic surgery.
- **Optimized and Adaptive Control:** The paper introduces an actor-critic neural network to optimize the prescribed-time control strategy. This reduces control effort, lowers power consumption, and minimizes heat generation in actuators, important for safety in medical applications. Additionally, the adaptive nature of the approach allows for effective operation in dynamic and unstructured environments.

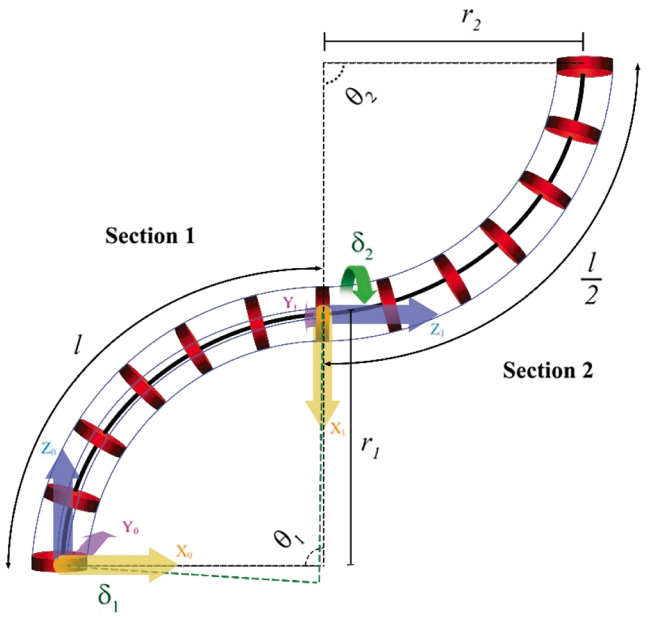


Fig. 1. Two Segment Continuum Robot Structure [34].

Table 1
Parameter values of the considered continuum robot [34].

Parameter	Designation	Value
l	Bending section length	0.3 m
m	Disk mass	0.01 kg
g	Gravity constant	9.81 m/s^2
E	Young's modulus	$2.1{\cdot}10^{11}$ Pa
I_b	Inertia moment of disk	$3.97{\cdot}10^{-12}$ m^4
I_{xx}	Inertia moment of elastic backbone	$3.06{\cdot}10^{-7}$ m^4

The paper is outlined as follows. The mathematical model of the continuum robot is described in section 2. Section 3 presents the procedure of controller design and its stability analysis through the Lyapunov function strategy. In section 4 the proposed controller is applied on the continuum robot and several scenarios are considered in the simulation to evaluate the performance of the proposed method against the method in [33]. Finally, in section 5, the conclusion and discussion are expressed.

2. Mathematical modeling of the continuum robot

Continuum robots are flexible robotic systems first designed by Victor Scheinman and Larry Leifer at Stanford University in 1965. These robots can continuously deform, achieving enhanced flexibility and precision in movement, making them particularly suitable for complex tasks in confined spaces. They operate by bending at their sections, allowing navigation through restricted environments while minimizing the risk of damage to surrounding areas. This paper focuses on the design of a controller for a specific continuum robot, segmented into two distinct sections, with four degrees of freedom (DoF).

Each section is controlled by three non-elastic cables, facilitated by intermediary support disks as depicted in Fig. 1 [34]. Within this configuration, the bending angles along the y-axis are denoted as θ_1 and θ_2, while the rotational angles about the z-axis are represented by δ_1 and δ_2. Additionally, the radii corresponding to the bending angles are r_1 and r_2, and the length of the central supporting structure, or backbone, for each section is designated as l.

Table 1

2.1. Kinematic model

To construct the kinematic model for the specified continuum robot, which follows the constant curvature assumption, it is necessary to determine the endpoint's local coordinates and the corresponding rotation matrix for each segment. Subsequently, the robot's global coordinates are computed. The local coordinates for the endpoint position vector in each segment of the continuum robot are specified in the following manner [34].

$$P_i^{i-1} = [x_i \quad y_i \quad z_i]^T, \ i = 1, 2 \tag{1}$$

Such as:

$$\begin{aligned} x_i &= \frac{l}{\theta_i}(1 - \cos(\theta_i))\cos(\delta_i), \\ y_i &= \frac{l}{\theta_i}(1 - \cos(\theta_i))\sin(\delta_i), \\ z_i &= \frac{l}{\theta_i}\sin(\theta_i) \end{aligned}$$

The rotation matrix for each segment of the continuum robot can be articulated through the subsequent matrix:

$$\begin{aligned} R_i^{i-1} &= Rot(z, \delta_i)Rot(y, \theta_i)Rot(z, -\delta_i) \\ &= \begin{bmatrix} c^2\delta_i c\theta_i + s^2\delta_i & c\delta_i s\delta_i c\theta_i - c\delta_i s\delta_i & c\delta i \ s\theta i \\ c\delta_i s\delta_i c\theta_i - c\delta_i s\delta_i & s^2\delta_i c\theta_i + c^2\delta_i & s\delta i \ s\theta i \\ -c\delta_i s\theta_i & -s\delta i \ s\theta i & c\theta i \end{bmatrix} \end{aligned} \tag{2}$$

Where c and s denote cosine and sine function respectively. The process for determining the global coordinates of the endpoint position of the continuum robot is as follows:

$$P_2^0 = P_1^0 + R_1^0.P_2^1 = [x \quad y \quad z]^T \tag{3}$$

Where

$$\begin{aligned} x &= l\left(\frac{c\delta_1 s\delta_2 s\theta_1 - c\delta_2(c\theta_2 - 1)(c^2\delta_1 c\theta_1 - c^2\delta_1 + 1)}{\theta_2} - \frac{s\delta_1 s\delta_2(c\theta_1 - 1)(c\theta_2 - 1)}{\theta_2} - \frac{c\delta_1(c\theta_1 - 1)}{\theta_1}\right) \\ y &= l\left(\frac{s\delta_1 s\delta_2 s\theta_1 - s\delta_2(c\theta_2 - 1)(c^2\delta_1 + c\theta_1 s^2\delta_1)}{\theta_2} - \frac{c\delta_1 c\delta_2 s\delta_1(c\theta_1 - 1)(c\theta_2 - 1)}{\theta_2} - \frac{s\delta_1(c\theta_1 - 1)}{\theta_1}\right) \\ z &= l\left(\frac{c\theta_1 s\delta_1 + c\delta_1 c\delta_2 s\theta_1(c\theta_2 - 1)}{\theta_2} + \frac{s\delta_1}{\theta_1} + \frac{s\delta_1 s\delta_2 s\theta_1(c\theta_2 - 1)}{\theta_2}\right) \end{aligned}$$

The computation of the rotation matrix within the global frame for the robot in question is derived using the Eq. 4.

$$R_2^0 = R_1^0.R_2^1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{4}$$

Such as:

$$
\begin{aligned}
a_{11} &= \left(c^2\delta_1 c\theta_1 - c^2\delta_1 + 1\right)\left(c^2\delta_2 c\theta_2 - c^2\delta_2 + 1\right) - c\delta_1 c\delta_2 s\theta_1 s\theta_2 + s\delta_1 s\delta_2 (c\theta_1 - 1)(c\theta_2 - 1)\\
a_{12} &= \frac{2s\delta_2(c\theta_2 - 1)(c^2\delta_1 c\theta_1 - c^2\delta_1 + 1)}{2} + \frac{2s\delta_1(c^2\delta_2 + c\theta_2 s^2\delta_2)(c\theta_1 - 1)}{2} - c\delta_1 s\delta_2 s\theta_1 s\theta_2\\
a_{13} &= c\delta_2 s\theta_2\left(c^2\delta_1 c\theta_1 - c^2\delta_1 + 1\right) + c\delta_1 c\theta_2 s\theta_1 + \frac{2s\delta_1 s\delta_2 s\theta_2(c\theta_1 - 1)}{2}\\
a_{21} &= \frac{2s\delta_1(c\theta_1 - 1)(c^2\delta_2 c\theta_2 - c^2\delta_2 + 1)}{2} + \frac{2s\delta_2(c^2\delta_1 + c\theta_1 s^2\delta_1)(c\theta_2 - 1)}{2} - c\delta_2 s\delta_1 s\theta_1 s\theta_2\\
a_{22} &= \left(c^2\delta_1 + s^2\delta_1 c\theta_1 s^2\delta_1\right)\left(c^2\delta_2 + c\theta_2 s^2\delta_2\right) - s\delta_1 s\delta_2 s\theta_1 s\theta_2 + c\delta_1 c\delta_2 s\delta_1 s\delta_2 (c\theta_1 - 1)(c\theta_2 - 1)\\
a_{23} &= c\theta_2 s\delta_1 s\theta_1 + s\delta_2 s\theta_2\left(c^2\delta_1 - c\theta_1 s^2\delta_1\right) + c\delta_1 c\delta_2 s\delta_1 s\theta_2(c\theta_1 - 1)\\
a_{31} &= -c\delta_1 s\theta_1\left(c^2\delta_2 c\theta_2 - c^2\delta_2 + 1\right) - c\delta_2 c\theta_1 s\theta_2 - \frac{2s\delta_2 s\delta_1 s\theta_1(c\theta_2 - 1)}{2}\\
a_{32} &= -c\theta_1 s\delta_2 s\theta_2 - s\delta_1 s\theta_1\left(c^2\delta_2 + c\theta_2 s^2\delta_2\right) - c\delta_1 c\delta_2 s\delta_2 s\theta_1(c\theta_2 - 1)\\
a_{33} &= c\theta_1 c\theta_2 - c\delta_1 c\delta_2 s\theta_1 s\theta_2 - s\delta_1 s\delta_2 s\theta_1 s\theta_2
\end{aligned}
$$

To formulate the dynamic model for the continuum robot, the method involving kinetic energy is employed. Consequently, the calculation of velocities and positions for each segment is executed in following manner:

$$v_i = [\dot{x}_i \quad \dot{y}_i \quad \dot{z}_i]^T \tag{5}$$

Where:

$$
\begin{cases}
\dot{x}_i = \dfrac{1}{\theta_i}\left(s\theta_i c\delta_i - \dfrac{1}{\theta_i}(1 - c\theta_i c\delta_i)\right)\dot{\theta}_i - \dfrac{1}{\theta_i}((1 - c\theta_i)s\delta_i)\dot{\delta}_i\\
\dot{y}_i = \dfrac{1}{\theta_i}\left(s\theta_i s\delta_i - \dfrac{1}{\theta_i}(1 - c\theta_i s\delta_i)\right)\dot{\theta}_i + \dfrac{1}{\theta_i}((1 - c\theta_i)c\delta_i)\dot{\delta}_i\\
\dot{z}_i = \dfrac{1}{\theta_i}\left(c\theta_i - \dfrac{1}{\theta_i}s\theta_i\right)\dot{\theta}_i
\end{cases}
$$

Through the derivation of Eq. (3), the endpoint velocity of the robot in the global frame is determined in the following way:

$$v = [\dot{x} \quad \dot{y} \quad \dot{z}]^T \tag{6}$$

Such as:

$$
\begin{aligned}
\dot{x} &= l\left(\frac{c\delta_1 s\theta_1}{\theta_1} + \frac{c\delta_1(c\theta_1 - 1)}{\theta_1^2} + \frac{s2\delta_1 s\delta_2 s\theta_1(c\theta_2 - 1)}{2\theta_2}\right.\\
&\left. + \frac{2c\delta_1 c\theta_1 s\delta_2 + 2c^2\delta_1 c\delta_2 s\theta_1(c\theta_2 - 1)}{2\theta_2}\right)\dot{\theta}_1\\
&+ l\left(\frac{2c\delta_2(c\theta_2 - 1)(c^2\delta_1 c\theta_1 - c^2\delta_1 + 1) - 2c\delta_1 s\delta_2 s\theta_1}{2\theta_2^2}\right.\\
&\left. - \frac{s2\delta_1 s\delta_2(c\theta_1 - 1)(c\theta_2 - 1)}{2\theta_2^2} + \frac{s2\delta_1 s\delta_2 s\theta_2(c\theta_1 - 1)}{2\theta_2}\right)\dot{\theta}_2\\
&+ \frac{2c\delta_2 s\theta_2(c^2\delta_1 c\theta_1 - c^2\delta_1 + 1)}{2\theta_2} + \frac{s\delta_1(c\theta_1 - 1)}{\theta_1}\Big)\dot{\delta}_1\\
&+ l\Big(-\frac{s\delta_1 s\delta_2 s\theta_1}{\theta_2} + \frac{s\delta_2(c\theta_1 - 1)(c\theta_2 - 1)(2c^2\delta_1 - 1)}{\theta_2}\\
&- \frac{2c\delta_1 c\delta_2 s\delta_1(c\theta_1 - 1)(c\theta_2 - 1)}{\theta_2}
\end{aligned}
$$

$$
\begin{aligned}
&+ l\left(\frac{2s\delta_2(c\theta_2 - 1)(c^2\delta_1 c\theta_1 - c^2\delta_1 + 1) + 2c\delta_1 c\delta_2 s\theta_1}{2\theta_2}\right.\\
&\left. - \frac{s2\delta_1 c\delta_2(c\theta_1 - 1)(c\theta_2 - 1)}{2\theta_2}\right)\dot{\delta}_2\\
\dot{y} &= l\left(\frac{s\delta_1 s\theta_1}{\theta_1} + \frac{s\delta_1(c\theta_1 - 1)}{\theta_1^2} + \frac{s^2\delta_1 s\delta_2 s\theta_1(c\theta_2 - 1)}{\theta_2}\right.\\
&\left. + \frac{c\theta_1 s\delta_1 s\delta_2 + c\delta_1 c\delta_2 s\delta_1 s\theta_1(c\theta_2 - 1)}{\theta_2}\right)\dot{\theta}_1\\
&+ l\left(\frac{s\delta_2 s\theta_2(c^2\delta_1 + c\theta_1 s^2\delta_1) + c\delta_1 c\delta_2 s\delta_1 s\theta_2(c\theta_1 - 1)}{\theta_2}\right.\\
&\left. + \frac{s\delta_2(c^2\delta_1 + c\theta_1 s^2\delta_1) + c\delta_1 c\delta_2 s\delta_1(c\theta_1 - 1)(c\theta_2 - 1)}{\theta_2^2}\right.\\
&\left. - \frac{s\delta_1 s\delta_2 s\theta_1}{\theta_2^2}\right)\dot{\theta}_2 + l\left(\frac{c\delta_2 s^2\delta_1(c\theta_1 - 1)(c\theta_2 - 1)}{\theta_2}\right.\\
&\left. + \frac{c\delta_1 s\delta_2 s\theta_1 - c^2\delta_1 c\delta_2(c\theta_1 - 1)(c\theta_2 - 1)}{\theta_2}\right.\\
&\left. - \frac{c\delta_1(c\theta_1 - 1)}{\theta_1} - \frac{2c\delta_1 s\delta_1 s\delta_2(c\theta_1 - 1)(c\theta_2 - 1)}{\theta_2}\right)\dot{\delta}_1\\
&+ l\left(\frac{c\delta_2 s\delta_1 s\theta_1 - c\delta_2(c^2\delta_1 + c\theta_1 s^2\delta_1)(c\theta_2 - 1)}{\theta_2}\right.\\
&\left. + \frac{c\delta_1 s\delta_1 s\delta_2(c\theta_1 - 1)(c\theta_2 - 1)}{\theta_2}\right)\dot{\delta}_2\\
\dot{z} &= l\left(\frac{c\delta_1 c\delta_2 c\theta_1(c\theta_2 - 1) - s\delta_2 s\theta_1 + c\theta_1 s\delta_1 s\delta_2(c\theta_2 - 1)}{\theta_2}\right.\\
&\left. - \frac{s\delta_1}{\theta_1^2}\right)\dot{\theta}_1 + l\left(-\frac{c\theta_1 s\delta_2 + c\delta_1 c\delta_2 s\theta_1(c\theta_2 - 1)}{\theta_2^2}\right.\\
&\left. - \frac{s\delta_1 s\delta_2 s\theta_1(c\theta_2 - 1)}{\theta_2^2} - \frac{c\delta_1 c\delta_2 s\theta_1 s\theta_2 - s\delta_1 s\delta_2 s\theta_1 s\theta_2}{\theta_2}\right)\dot{\theta}_2\\
&+ l\left(\frac{c\delta_1}{\theta_1} + \frac{c\delta_1 s\delta_2 s\theta_1(c\theta_2 - 1) - c\delta_2 s\delta_1 s\theta_1(c\theta_2 - 1)}{\theta_2}\right)\dot{\delta}_1\\
&+ l\left(\frac{c\delta_2 c\theta_1 - c\delta_1 s\delta_2 s\theta_1(c\theta_2 - 1) + c\delta_2 s\delta_1 s\theta_1(c\theta_2 - 1)}{\theta_2}\right)\dot{\delta}_2
\end{aligned}
$$

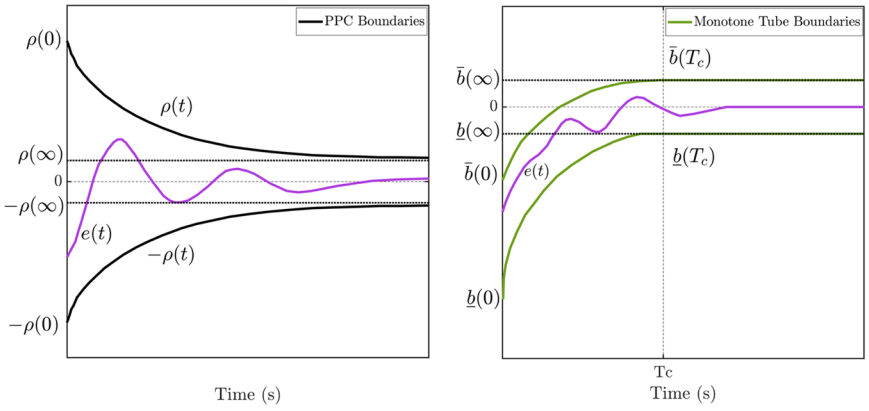


Fig. 2. Comparison between standard PPC boundaries and monotone tube boundaries.

2.2. Dynamic model

The creation of the dynamic model is essential during the controller's design phase. In this study, the Euler-Lagrange approach is utilized to construct the dynamic model for the continuum robot under consideration, which it is assumed orientation angles are zero (where $\delta_1 = \delta_2 = 0$). The Euler-Lagrange (Eq. (7)) is presented as follows :

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_i} - \frac{\partial T}{\partial \theta_i} + \frac{\partial U}{\partial \theta_i} = Q_i \tag{7}$$

In this context, T represents the sum of kinetic energy, U denotes the aggregate potential energy, and Q_i refers to the system's generalized forces. For the robot being analyzed, and by applying Eq. (7), the dynamic model is articulated as follows [34]:

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1^2 \\ \dot{\theta}_1\dot{\theta}_2 \\ \dot{\theta}_2^2 \end{bmatrix} + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} \tag{8}$$

With

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},\ C = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \end{bmatrix},$$
$$K = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \text{ and } Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}.$$

M, C and K indicate mass, Coriolis and gravitional matrix respectively.

The dynamic model, as outlined by Eq. (8), is expressed in the following general model typically used in robotic systems:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + K(\theta) = F \tag{9}$$

Where $\theta,\ F \in \mathbb{R}^2$.

3. Preliminaries, Controller Design and Stability Analysis

The control objective is to design an approximately optimized controller to force the bending angles θ to track the desired trajectory in a prescribed-time by considering system safety and state constraints. Prior to commencing controller design, relevant definitions and lemmas that enhance the comprehension of concepts related to monotone tube boundaries, prescribed-time control, reinforcement learning-based optimal control and designed controller stability analysis are introduced.

3.1. Monotone Tube Boundary

Monotone tube boundaries are a class of boundary functions designed for prescribed performance control, establishing a quantitative relationship between control performance (both transient and steady-state) and user-defined indices such as overshoot, accuracy, and convergence time, with reduced conservatism. Inspired by dynamic tube model predictive control (MPC), these boundaries optimize invariant tube geometry and open-loop trajectories simultaneously. Monotone tube boundary functions offer a constructive solution for controller design, ensuring the desired tracking performance by transforming the prescribed performance control (PPC) problem into one of forcing forward invariance for the closed loop. This approach provides alternative technical routes for studying prescribed performance using tools for analyzing invariance in nonlinear systems. If the tracking error remains within these boundaries, performance can be adjusted in both transient and steady-state stages. In contrast, the standard PPC approach uses two-sided boundaries, making it challenging to constrain tracking error overshoot. Additionally, the exponential-type function in traditional PPC only reaches preselected values as time approaches infinity, preventing accurate pre-determination of the convergence set after a pre-assigned convergence time.

Monotone tube boundaries link their parameters to widely-used quantitative indices in industry, offering several advantages over traditional PPC methods:

- **Reduced Conservatism:** Monotone tube boundaries provide a less conservative approach, allowing for more precise control adjustments.
- **Enhanced Adjustability:** These boundaries enable better control over transient and steady-state performance, avoiding the excessive overshoot common in traditional PPC.
- **Accurate Convergence:** The method allows for accurate pre-determination of the convergence set, enhancing control precision and reliability.

By leveraging these advantages, monotone tube boundaries offer a promising solution for improving prescribed performance control in various applications. It is worth mentioning we can transform constraints on the bending angles of the continuum robot to the tracking error and use monotone tubes for safety considerations. To streamline the process of monotone tube-based controller design for the Continuum Robot, the following Lemma is presented.

Lemma 1. [36]: Consider a nonlinear system as:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= g(x)u + f(x) \\ y &= x_1 \end{aligned} \tag{10}$$

Where $x_1 \in \mathbb{R}^n,\ x_2 \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ are the states and control inputs of the system and $x = [x_1, x_2]^\top$. Define tracking error as $e = y - y_r$ and

transform it as

$$z = \left[\ln\left(\frac{\varepsilon_1}{1-\varepsilon_1}\right), \ldots, \ln\left(\frac{\varepsilon_n}{1-\varepsilon_n}\right)\right]^\top = [z_1, \ldots, z_n]^\top \tag{11}$$

Where e_i is the *ith* element of e,

$$\begin{aligned}
\varepsilon_i &= \frac{1}{\Delta b_i}\left(e_i - \underline{b}_i\right),\\
\underline{b}_i &= \text{sign}(e_i(0))(q_i - q_{\infty i}) - \underline{\lambda}_i q_i,\\
\overline{b}_i &= \text{sign}(e_i(0))(q_i - q_{\infty i}) + \overline{\lambda}_i q_i,\\
q_i &= \begin{cases} \text{csch}\left(q_{0i} + \frac{\alpha_i t}{T_{ci} - t}\right) + q_{\infty i} & ,0 \leqslant t < T_{ci} \\ q_{\infty i} & ,t \geq T_{ci} \end{cases},\\
\Delta b_i &= \overline{b}_i - \underline{b}_i.
\end{aligned}$$

There is a control input as $u = \varpi\left(x, e, z, \dot{y}_r, \ddot{y}_r\right)$ stabilizes the system (10) and results in the boundedness of z and consequently brings the following properties:

P1 (Boundedness): The entire system state x remains bounded.

P2 (Forward Invariance): The system maintains forward invariance within the following time-varying monotone tube set.

$$\mathscr{E} := \left\{x_1 \in \mathbb{R}^n \middle| \underline{b} < y - y_r < \overline{b}, t \geq 0\right\}, \underline{b} = \left[\underline{b}_1, \ldots, \underline{b}_n\right]^\top, \overline{b}(t) = [\overline{b}_1, \ldots, \overline{b}_n]^\top \tag{12}$$

Proof. For more details about the proof of this Lemma and the control input u refer to [35].

Fig. 2 illustrates the characteristics of monotone tube boundaries and provides a qualitative comparison of its performance with the PPC method. It is important to note that $\underline{b}_i(0)$, $\underline{b}_i(\infty)$, $\underline{b}_i(T_c)$, $\overline{b}_i(0)$, $\overline{b}_i(\infty)$ and $\overline{b}_i(T_c)$ are obtained by substituting the values 0, ∞, and T_c) into the definitions of $\underline{b}$ and $\overline{b}_i$. The Prescribed Performance Function (PPF) used in the PPC is defined as: $\rho = (\rho_0 - \rho_\infty)e^{-at} + \rho_\infty$, where ρ_0 and ρ_∞ are positive constants representing the initial value, final boundary, and decay rate of the Prescribed Performance Boundary (PPB), respectively. Note, In control methods where the goal is to guarantee tracking performance within a specific range, the selection of this range is based on constraints on the system's output or tracking error. Specifically, it is predetermined within what bounds the transient and steady-state errors should lie. This is similar to classical linear control methods, where parameters such as maximum overshoot, steady-state error, settling time, and rise time are used to determine controller gains. Similarly, in our approach, the values of b_ and b¯ are determined based on these specifications, ensuring the system performs within the desired error limits. A detailed explanation of how these parameters are defined can be found in reference [35].

3.2. Prescribe-time Control

Finite-time, fixed-time, and prescribed-time control methods each offer distinct advantages for system stabilization. Finite-time control ensures that the system state converges to a desired region within a finite but generally unknown time, which depends on the initial conditions. Fixed-time control extends this property by ensuring convergence time upper bound independent of initial conditions. Prescribed-time control further strengthens this framework by allowing explicit predefinition of the convergence time, making it particularly useful in applications where strict timing constraints are required.A key stability property related to prescribed-time control is Prescribed-Time Input-to-State Stability (PT-ISS). PT-ISS extends the conventional Input-to-State Stability (ISS) framework by ensuring that the system state is bounded and converges to a small residual set within a predefined time, irrespective of initial conditions or external disturbances. This property is crucial in control applications where stability must be guaranteed within a strict time frame. The following Lemma discusses prescribed-time control.

Lemma 2. [37]: If for a nonlinear system $\dot{x} = f(x, t, d)$ there is a smooth positive definite radially unbounded function $V : \mathbb{R}^n \to \mathbb{R}^+$ satisfies:

$$\dot{V}(t) \leqslant -2k^* \mu_1(t, t_0)V(t) + \mu_1(t, t_0)C_1^* + C_2 \tag{13}$$

for positive constants k^*, C_1 and C_2, then

$$V(t) \leqslant V(t_0)e^{\frac{-2k^* T_p}{p-1}\left(\left(\frac{T_p}{T_p + t_0 - t}\right)^{p-1} - 1\right)} + \frac{C^*}{2k^*} \tag{14}$$

and consequently it can be concluded that the system is prescribed-time input-to-state stable (PT-ISS) in time T_p.Where $\mu_1(t, t_0) \geqslant 1$ and $C = C_1 + C_2^*$. The definition of PT-ISS can be found in [37].

3.3. A General Point of View of Reinforcement Learning Based Optimal Control

Let's contemplate a nonlinear system, represented as follows:

$$\dot{x}(t) = f(x(t), u(x(t))) \tag{15}$$

In this equation, $x(t) \in \mathbb{R}^n$ is the state variable, $u(x(t)) \in \mathbb{R}^m$ is the control input, and $f(x(t), u(x(t))) \in \mathbb{R}^n$ is a continuous function satisfying the condition $f(0) = 0$. To derive the optimal control law for this system, we introduce the integral performance index over an infinite horizon, also known as the value function:

$$J(x, u) = \int_t^\infty c(x(\tau), u(x(\tau)))d\tau \tag{16}$$

Here, $c(x, u) = x^T P_1 x + u^T(x)P_2 u(x) \in \mathbb{R}$ is the cost function. The matrices $P_1 = P_1^T \in \mathbb{R}^{n \times n}$ and $P_2 = P_2^T \in \mathbb{R}^{m \times m}$ are positive semi-definite and definite, respectively.

Definition 3. (Admissible control) [38]: The control law $u(x)$ in system (15) is considered to be admissible within the set Ω_u under the following prerequisites: The control law $u(x)$ must exhibit continuity, the equation $u(0_n) = 0_m$ must hold, the control law should ensure the stability of system (15) and the control law ensure the boundedness of value function $J(x)$. It is noteworthy that the admissible control law within the set Ω_u is represented as $u(x) \in \Psi(\Omega_u)$.

The HJB equation is a cornerstone of optimal control theory. It offers a method to ascertain the best control policy for a dynamic system by solving a specific partial differential equation. The derivation of the HJB equation is based on the optimality principle, which posits that irrespective of the initial state and decision, the subsequent decisions must constitute an optimal control policy with respect to the state that results from the initial decision. For a system represented by Eq. (15) and a value function given by equation (6), the HJB equation is defined as follows [39]:

$$\min_{u \in U}\left(c(x, u) + \left(\frac{J(x, u)}{\partial x}\right)^T f(x, u)\right) = 0 \tag{17}$$

u denotes the set of admissible control inputs. The solution to the HJB equation yields the optimal value function $J^*(x, t)$, which can be employed to derive the optimal control policy. The optimal control policy can be determined by finding the value of u^* that minimizes the expression inside the minimum operator in the HJB Eq. (7), as shown below [39]:

$$u^*(x, t) = \arg\min_{u \in U}\left(c(x, u) + \left(\frac{J(x, u)}{\partial x}\right)^T f(x, u)\right) \tag{18}$$

Let's consider the dynamics of system (15) to be represented as follows:

$$\dot{x}(t) = f(x) + g(x)u(x) \tag{19}$$

For a system that takes the form of Eq. (19), the challenge of optimal control is to identify a suitable control that minimizes the performance index (16) to achieve the control objective. Let's denote the optimal control law for system (19) as $u^*(x)$. By replacing $u(x)$ with $u^*(x)$ in Eq. (16), we obtain the optimal performance index function:

$$J^*(x,u) = \int_t^\infty c(x(\tau), u^*(x(\tau)))d\tau = \min_{u\in\Psi(\Omega_x)}\left(\int_t^\infty c(x(\tau), u(x(\tau)))d\tau\right) \tag{20}$$

The HJB equation is derived by calculating the time derivative on both sides of the first part of Eq. (10):

$$\begin{aligned} H(x,u^*,J^*) &= x^T P_1 x + u^{*T}(x)P_2 u^*(x) + \left(\frac{J^*(x,u)}{dx}\right)^T \\ &\times (f(x) + g(x)u^*(x)) = 0 \end{aligned} \tag{21}$$

Given the relationship, the optimal control solution $u^*(x)$ associated with the optimal performance index function (20) is unique. Therefore, it is necessary for it to satisfy the HJB Eq. (21) as a unique control solution. In this case, $u^*(x)$ can be derived from the solution $\partial H(x,u^*,J^*)/\partial u^* = 0$ as follows:

$$u^*(x) = -\frac{1}{2}P_2^{-1}g^T(x)\frac{dJ^*(x,u)}{dx} \tag{22}$$

However, the optimal control (22) is not directly obtainable due to the uncertain term $\frac{dJ^*(x)}{dx}$. To acquire the available optimal control and to ensure (22) as the unique control solution (21), $\frac{dJ^*(x,u)}{dx}$ should be obtained. This can be achieved by solving the following HJB equation, which is derived by substituting (22) into (21):

$$\begin{aligned} H(x,u^*,J^*) &= x^T P_1 x + \frac{dJ^*(x)}{dx^T}f(x) - \\ &\frac{1}{4}\frac{dJ^*(x)}{dx^T}g(x) \times P_2^{-1}g^T(x)\frac{dJ^*(x)}{dx} = 0 \end{aligned} \tag{23}$$

Note that finding the analytical solution of the HJB Eq. (23) is challenging due to its strong nonlinearity. To address this issue, RL is employed as a practical method to find an approximate solution for optimal control [39]. Among the various reinforcement learning methods, the actor-critic reinforcement learning method has been extensively utilized in optimal control. To do this usually to reduce the uncertainty of the optimal control law (22) and guarantee the stability of the system, (22) can be rewritten as follows:

$$u^*(x) = \varpi(x) - J_0(x) \tag{24}$$

Where $\varpi(x)$ is a feedback control law proposed by the designer and known and $J_0(x)$ is an unknown term. $J_0(x)$ can be approximated solution by RL as follows.

$$J_0(x) = W^T\Phi(x) + \varepsilon(x) \tag{25}$$

Where $W \in R^l$ represent the RL actor-critic NN weights, where p is the number of neurons. The basis function vector $\Phi(x) = [\varphi_1(x),\ldots,\varphi_l(x)]^T$ is defined with $\varphi_i(x) = \exp\left[-(x-m_i)^T(x-m_i)/\varrho_i^2\right]$, where ϱ_i denotes the width of the Gaussian function and $m_i = [m_{i1},\ldots,m_{in}]^T$ are the centers of the receptive fields for $i = 1,2,\ldots,l$. Here, $\varepsilon(x) \in R$ represents the approximation error.

Let define a useful Lemma that will be used in the stability analysis named Young's inequality.

Lemma 3. [40] (Young's inequality): For any positive variables θ_1 and θ_2, the following inequality holds:

$$\theta_1\ \theta_2 \le \frac{1}{\beta_1}\theta_1^{\beta_1} + \frac{1}{\beta_2}\theta_2^{\beta_2} \tag{26}$$

where $\beta_1,\ \beta_2 > 1$ and $\frac{1}{\beta_1} + \frac{1}{\beta_2} = 1$. Based on (26), it is easy to prove that if θ_1 and θ_2 are vectors, the following inequality holds

$$\theta_1^T\ \theta_2 \le \frac{1}{\beta_1}\theta_1^{\beta_1} + \frac{1}{\beta_2}\theta_2^{\beta_2} \tag{27}$$

Where $\theta_1^{\beta_1}$ and $\theta_2^{\beta_2}$ indicate the element-wise power for vectors θ_1 and θ_2.

3.4. Safe High Precision Reinforcement Learning based Optimal Control

The proposed control scheme is demonstrated in Fig. 3.

By introducing the following variables the following state-space mode for (9) can be obtained.

$$\begin{aligned} x_1 &= \theta, \\ x_2 &= \dot{\theta}, \\ x &= \left[x_1^\top, x_2^\top\right]^\top, \\ u &= F \end{aligned}$$

And define:

$$\begin{aligned} g(x) &= M^{-1}(\theta), \\ f(x) &= -M^{-1}(\theta)(C(\theta,\dot{\theta}) + K(\theta)) \end{aligned}$$

Therefore (9) can be rewritten as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= g(x)u + f(x) \\ y &= x_1 \end{aligned} \tag{28}$$

Based on (11) the dynamics of the transformed error can be written as:

$$\dot{z}_i = r_i\dot{e}_i + \omega_i \tag{29}$$

With $r_i = \frac{1}{\varepsilon_i(1-\varepsilon_i)\Delta b_i}$, $\omega_i = \frac{r_i}{\Delta b_i}(\underline{b}_i\dot{\bar{b}}_i - \dot{\underline{b}}_i\bar{b}_i - e_i\dot{\Delta}b_i)$. Define $\omega_i = \frac{r_i}{\Delta b_i}(\underline{b}_i\dot{\bar{b}}_i - \dot{\underline{b}}_i\bar{b}_i - e_i\dot{\Delta}b_i)$, $\omega = [\omega_1,\omega_2]^\top$, so we have:

$$\dot{z} = r\dot{e} + \omega \tag{30}$$

We introduce a novel sliding surface which can ensure the prescribed-time stability of tracking errors.

$$s_i = \dot{z}_i + \mu z_i \tag{31}$$

Where μ is a scaling factor that provides the prescribed-time stability.

$$\mu = \begin{cases} \dfrac{\sigma}{T_p - t}, t < T_p \\ \delta, t \geqslant T_p \end{cases} \tag{32}$$

Where, σ and δ are positive constants and T_p is the prescribed settling time.

Define

$$s = [s_1, s_2]^\top \tag{33}$$

Therefore,

$$s = \dot{z} + \mu z \tag{34}$$

the time-derivative of s is equal to:

$$\dot{s} = \ddot{z} + \dot{\mu}z + \mu\dot{z} \tag{35}$$

Using (26) and (28), (32) yields:

$$\dot{s} = \dot{r}\dot{e} + r\ddot{e} + \dot{\omega} = rg(x)u + r\left(f(x) - \ddot{y}_r\right) + \dot{r}\dot{e} + \dot{\omega}\mu z + \mu\dot{z} \tag{36}$$

To assure the optimality of control law based on section 3.3 we define the following value function.

$$J(s) = \int_{t}^{+\infty} (s^{\top}(\tau)Ps(\tau) + u^{\top}(\tau)Ru(\tau))d\tau \tag{37}$$

Where $P \geqslant 0$ and $R > 0$.

The optimal value function is equal to:

$$J^{*}(s) = \int_{t}^{+\infty} (s^{\top}(\tau)Ps(\tau) + u^{*\top}(\tau)Ru^{*}(\tau))d\tau = \min_{u \in \psi(\Lambda_u)} \left(\int_{t}^{+\infty} (s^{\top}(\tau)Ps(\tau) + u^{\top}(\tau)Ru(\tau))d\tau \right) \tag{38}$$

To attain the optimal control law u^{*} we get the following HJB equation (Hamilton–Jacobi–Bellman).

$$H(s, u^{*}, J^{*}) = s^{\top}Ps + u^{*\top}Ru^{*} + \left(\frac{dJ^{*}(s)}{ds}\right)^{\top} \dot{s} \tag{39}$$

Substituting (34) into (37) yields:

$$\begin{aligned} H(s, u^{*}, J^{*}) &= s^{\top}Ps + u^{*\top}Ru^{*} \\ &+ \left(\frac{dJ^{*}(s)}{ds}\right)^{\top} \left(rg(x)u^{*} + r\left(f(x) - \ddot{y}_r\right) + \dot{re} + \dot{\omega} + \dot{\mu}z + \mu\dot{z} \right) \end{aligned} \tag{40}$$

To derive u^{*} we calculate the derivative of $H(s, u^{*}, J^{*})$ with respect to u^{*} and solve the following equation:

$$\frac{\partial H(s, u^{*}, J^{*})}{\partial u^{*}} = 0 \tag{41}$$

So we have:

$$u^{*} = -\frac{1}{2}R^{-1}g^{\top}(x)r\frac{dJ^{*}(s)}{ds} \tag{42}$$

We rewrite u^{*} in the following form:

$$u^{*} = -\mu G_1 s - G_1 J^{0}(s) = -G_2\frac{dJ^{*}(s)}{ds} \tag{43}$$

Where

$$G_1 = (rg(x))^{-1} \tag{44}$$

$$G_2 = \frac{1}{2}R^{-1}g^{\top}(\boldsymbol{x})r^{\top}$$

and

$$J^{0}(s) = -\mu s + G_1^{-1}G_2\frac{dJ^{*}(s)}{ds} \tag{45}$$

To attain $\frac{dJ^{*}(s)}{ds}$ we should substitute (43) into (40) and solve this highly nonlinear and complicated equation. Hence, we use the RL to overcome this difficulty and approximate $J^{0}(s)$ through Neural Network (NN).

$$J^{0}(s) = W^{*\top}\Phi(s) + \varepsilon(s) \tag{46}$$

Where $W^{*} \in \mathbb{R}^{l \times n}$ is the optimal weight of NN, $\Phi(s)\mathbb{R}^{l \times 1}$ is the NN activation function and $\varepsilon(s)$ is the NN approximation error. Therefore $\frac{dJ^{*}(s)}{ds}$ can be written as:

$$\frac{dJ^{*}(s)}{ds} = \mu G_2^{-1}G_1 s + G_2^{-1}G_1\ W^{*\top}\Phi(s) + G_2^{-1}G_1\varepsilon(s) \tag{47}$$

Therefore the optimal control law can be rewritten as:

$$u^{*} = -\mu G_1 s - G_1 W^{*T}\Phi(s) - G_1\varepsilon(s) \tag{48}$$

Moreover, the cost function can be estimated by the critic NN as:

$$\frac{d\widehat{J}^{*}(s)}{ds} = \mu G_2^{-1}G_1 s + G_2^{-1}G_1\widehat{W}_c^{\top}\Phi(s) \tag{49}$$

Since W^{*} is unknown we use its estimation in the control law. In the control law, we use the actor NN and $\widehat{W}_a$ is the estimation of the ideal weights of actor NN. $\widehat{W}_c$ is the estimation ideal weights of critic NN.

$$\widehat{u}^{*} = -\mu G_1 s - G_1\widehat{W}_a^{\top}\Phi(s) \tag{50}$$

Our approach employs online reinforcement learning, where the agent interacts with the surgical robot's environment in real-time, continuously updating the actor and critic neural network weights through adaptive laws (Equations 61 and 68). Unlike offline RL, which relies on a static dataset and typically evaluates convergence after a fixed number of training iterations, our method dynamically adapts without requiring a predefined number of iterations. Instead, convergence is evaluated through the stable evolution of weights, as demonstrated in Fig. 7. This aligns with the online RL paradigm, where performance continuously improves during direct interaction with the environment.

We construct the control input composed of two terms, where $\widehat{u}^{*}$ is the estimation of the optimal control and u_c which is a compensation term.

$$u = \widehat{u}^{*} + u_c \tag{51}$$

Define the actor and critic weights estimation errors as:

$$\widetilde{W}_a = \widehat{W}_a - W^{*}, \widetilde{W}_c = \widehat{W}_c - W^{*} \tag{52}$$

The actor-critic (AC) framework in this study fully embodies the fundamental RL property of the action-state-reward relationship. The state space is defined by the sliding surface $s = \dot{z} + \mu z$ (Equation 34), reflecting tracking error dynamics. The action space comprises admissible control inputs $u = \widehat{u}^{*} + u_c$ (Equation 51), where the actor network $\widehat{u}^{*} = -\mu G_1 s - G_1\widehat{W}_a^{\top}\Phi(s)$ (Equation 50) learns an optimal policy. The reward signal is embedded in the cost function $J(s) = \int_{t}^{+\infty} (s^{\top}(\tau)Ps(\tau) + u^{\top}(\tau)Ru(\tau))d\tau$ (Equation 37), which penalizes tracking errors $s^{\top}(\tau)$ and control effort $u^{\top}(\tau)Ru(\tau)$, guiding the agent to minimize both. The critic estimates $\frac{d\widehat{J}^{*}(s)}{ds}$ (Equation 49) to evaluate state-cost relationships, while the actor adjusts actions based on this feedback, with weights updated online via Equations 61 and 68. This process mirrors RL's iterative learning of an optimal policy through state-action-reward interactions, distinguishing it from mere adaptive updating by optimizing long-term performance in dynamic surgical environments.

To design u_c and adaptation laws we consider the following Lyapunov function.

$$V = \frac{1}{2}s^{\top}s + \frac{1}{2}trace\left(\widetilde{W}_c^{\top}\Gamma_c^{-1}\widetilde{W}_c\right) + \frac{1}{2}trace\left(\widetilde{W}_a^{\top}\Gamma_a^{-1}\widetilde{W}_a\right) \tag{53}$$

Γ_c and Γ_a are two positive definte matrices. Taking the time-

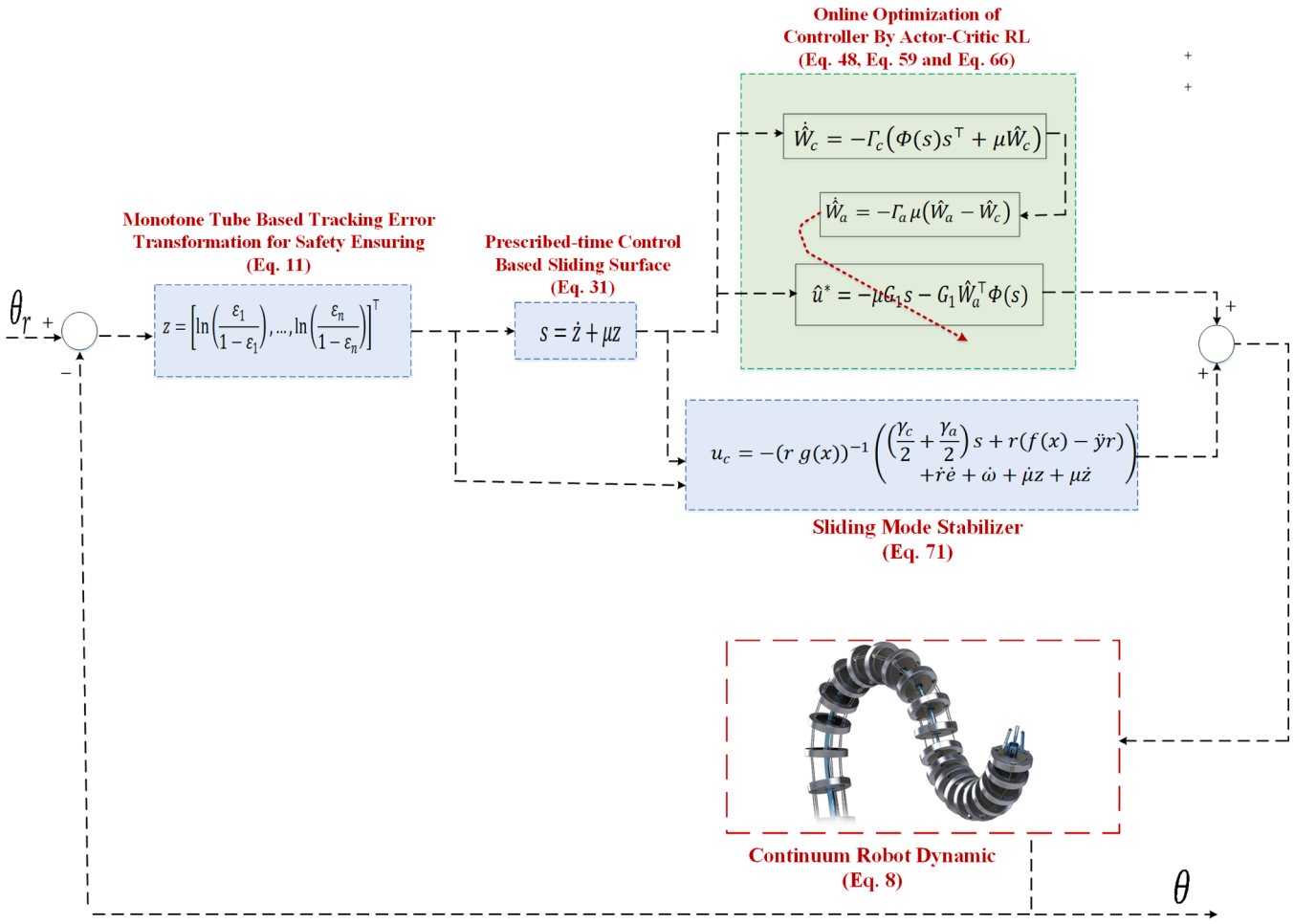


Fig. 3. Proposed method block diagram.

derivative of (53) yields:

$$\dot{V} = s^\top\dot{s} + trace\left(\widetilde{W}_c^\top\Gamma_c^{-1}\dot{\widehat{W}}_c\right) + trace\left(\widetilde{W}_a^\top\Gamma_a^{-1}\dot{\widehat{W}}_a\right) \tag{54}$$

Substituting (34) into (52) yields:

$$\begin{aligned}\dot{V} &= s^\top\left(rg(x)u + r\left(f(x) - \ddot{y}_r\right) + \dot{r}\dot{e} + \dot{\omega}\right) + trace\left(\widetilde{W}_c^\top\Gamma_c^{-1}\dot{\widehat{W}}_c\right)\\ &+ trace\left(\widetilde{W}_a^\top\Gamma_a^{-1}\dot{\widehat{W}}_a\right) = s^\top\left(-\mu s - \widehat{W}_a^\top\Phi(s) + rg(x)u_c + r\left(f(x) - \ddot{y}_r\right)\right.\\ &\left.+\dot{r}\dot{e} + \dot{\omega} + \dot{\mu}z + \mu\dot{z}\right) + trace\left(\widetilde{W}_c^\top\Gamma_c^{-1}\dot{\widehat{W}}_c\right) + trace\left(\widetilde{W}_a^\top\Gamma_a^{-1}\dot{\widehat{W}}_a\right)\end{aligned} \tag{55}$$

For the term $-s^\top\widehat{W}_a^\top\Phi(s)$ we have:

$$-s^\top\widehat{W}_a^\top\Phi(s) = -s^\top\widetilde{W}_a^\top\Phi(s) - s^\top\widehat{W}_c^\top\Phi(s) + s^\top\widetilde{W}_c^\top\Phi(s) \tag{56}$$

From the properties of the trace operator, we have:

$$s^\top\widetilde{W}_c^\top\Phi(s) = trace\left(\widetilde{W}_c^\top\Phi(s)s^\top\right) \tag{57}$$

Using Young's inequality in Lemma 3 term $-s^\top\widetilde{W}_a^\top\Phi(s)$ stands in the following inequality:

$$\begin{aligned}-s^\top\widetilde{W}_a^\top\Phi(s) &\leqslant \frac{\gamma_a}{2}s^\top s + \frac{1}{2\gamma_a}\Phi^\top(s)\widetilde{W}_a\widetilde{W}_a^\top\Phi(s)\\ &= \frac{\gamma_a}{2}s^\top s + \frac{1}{2\gamma_a}trace\left(\widetilde{W}_a^\top\Phi(s)\Phi^\top(s)\widetilde{W}_a\right) \leqslant \frac{\gamma_a}{2}s^\top s + \frac{\varrho}{2\gamma_a}trace\left(\widetilde{W}_a^\top\widetilde{W}_a\right)\end{aligned} \tag{58}$$

Where ϱ is the maximum Eigenvalue of $\Phi(s)\Phi^\top(s)$. Similarly we have:

$$-s^\top\widehat{W}_c^\top\Phi(s) \leqslant \frac{\gamma_c}{2}s^\top s + \frac{\varrho}{2\gamma_c}trace\left(\widehat{W}_c^\top\widehat{W}_c\right) \tag{59}$$

γ_a and γ_c are two positive constants. Therefore $\dot{V}$ stands in the following inequality:

$$\begin{aligned}\dot{V} \leqslant s^\top&\left(-\mu s + rg(x)u_c + r\left(f(x) - \ddot{y}_r\right) + \dot{r}\dot{e} + \dot{\omega} + \dot{\mu}z + \mu\dot{z}\right)\\ &+trace\left(\widetilde{W}_c^\top\left(\Gamma_c^{-1}\dot{\widehat{W}}_c + \Phi(s)s^\top\right)\right) + trace\left(\widetilde{W}_a^\top\Gamma_a^{-1}\dot{\widehat{W}}_a\right)\end{aligned} \tag{60}$$

Let propose the critic NN weights updating law as follows:

$$\dot{\widehat{W}}_c = -\Gamma_c(\Phi(s)s^\top + \mu\widehat{W}_c) \tag{61}$$

Substituting (59) into (58) yields:

$$\begin{aligned}V \leqslant s^\top&\left(-\mu s + rg(x)u_c + r\left(f(x) - \ddot{y}_r\right) + \dot{r}\dot{e} + \dot{\omega} + \dot{\mu}z + \mu\dot{z}\right)\\ &+trace\left(-\mu\widetilde{W}_c^\top\widehat{W}_c\right) + trace\left(\widetilde{W}_a^\top\Gamma_a^{-1}\dot{\widehat{W}}_a\right)\end{aligned} \tag{62}$$

The term $-\mu\widetilde{W}_c^\top\widehat{W}_c$ can be written as:

$$\begin{aligned}-\mu\widetilde{W}_c^\top\widehat{W}_c &= -\frac{\mu}{2}\widetilde{W}_c^\top\widehat{W}_c - \frac{\mu}{2}\widehat{W}_c^\top\widehat{W}_c + \frac{\mu}{2}W^{*\top}\widehat{W}_c\\ &= -\frac{\mu}{2}\widehat{W}_c^\top\widehat{W}_c - \frac{\mu}{2}\widetilde{W}_c^\top\widetilde{W}_c - \frac{\mu}{2}\widetilde{W}_c^\top W^* + \frac{\mu}{2}W^{*\top}\widehat{W}_c\end{aligned} \tag{63}$$

So

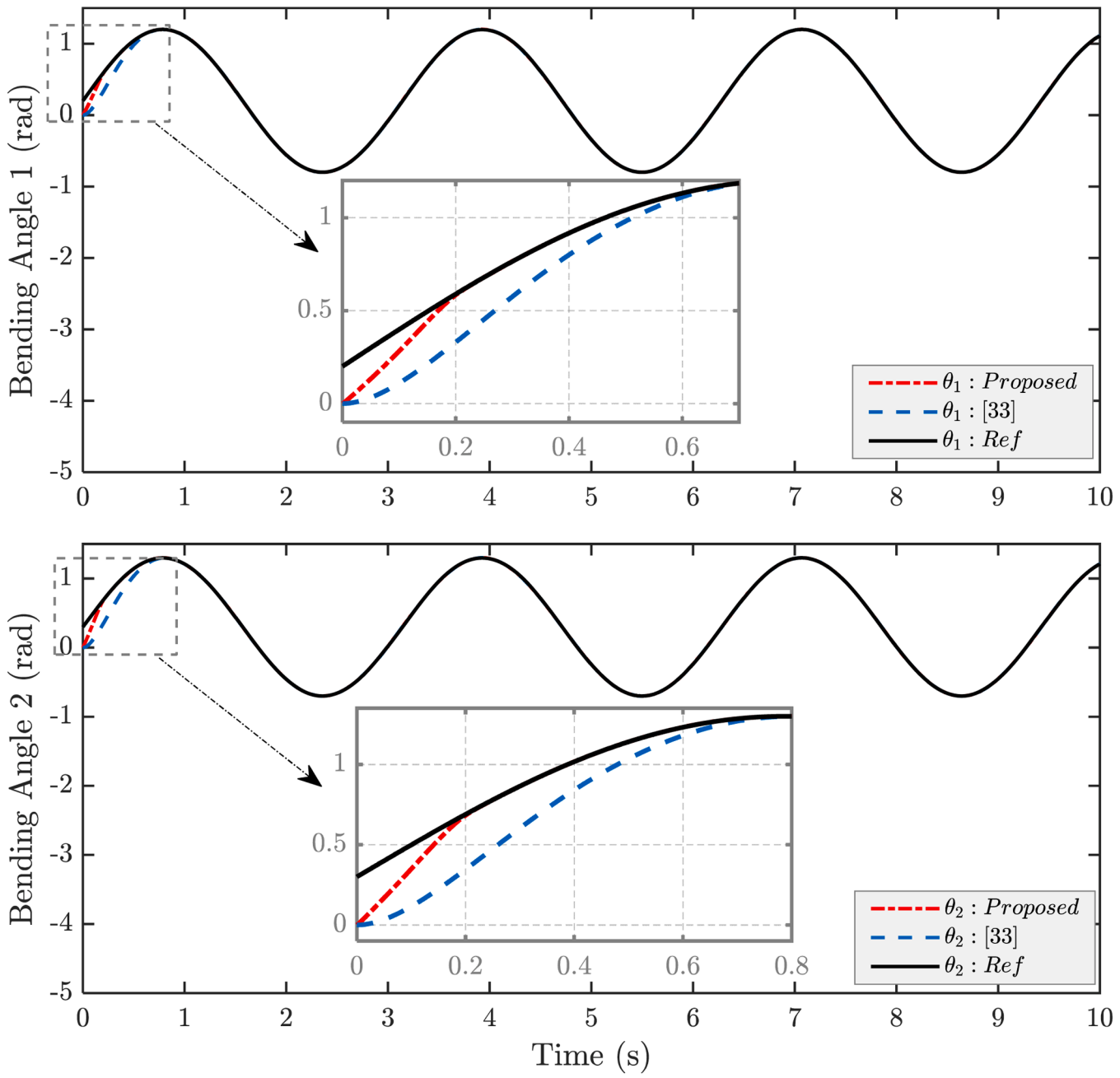


Fig. 4. Comparison of continuum robot bending angles tracking under case 1.

$$trace\left(-\mu\widetilde{W}_c^{\top}\widehat{W}_c\right) = -\frac{\mu}{2}trace\left(\widehat{W}_c^{\top}\widehat{W}_c\right) - \frac{\mu}{2}trace\left(\widetilde{W}_c^{\top}\widetilde{W}_c\right) - \frac{\mu}{2}trace\left(\widetilde{W}_c^{\top}W^*\right) + \frac{\mu}{2}trace\left(W^{*\top}\widehat{W}_c\right) \tag{64}$$

Using Young's inequality again leads:

$$-\frac{\mu}{2}trace\left(\widetilde{W}_c^{\top}W^*\right) = -\frac{\mu}{2}(Vec(\widetilde{W}_c))^{\top}Vec(W^*) \leqslant \frac{\mu\delta_c}{4}(Vec(\widetilde{W}_c))^{\top}Vec(\widetilde{W}_c) + \frac{\mu}{4\delta_c}(Vec(W^*))^{\top}Vec(W^*) = \frac{\mu\delta_c}{4}trace\left(\widetilde{W}_c^{\top}\widetilde{W}_c\right) + \frac{\mu}{4\delta_c}trace(W^{*\top}W^*) \tag{65}$$

$$\frac{\mu}{2}trace(W^{*\top}\widehat{W}_c) \leqslant \frac{\mu\sigma_c}{4}trace\left(\widehat{W}_c^{\top}\widehat{W}_c\right) + \frac{\mu}{4\sigma_c}trace(W^{*\top}W^*) \tag{66}$$

Thus we have:

$$trace\left(-\mu\widetilde{W}_c^{\top}\widehat{W}_c\right) \leqslant \left(-\frac{\mu}{2} + \frac{\mu\delta_c}{4}\right)trace\left(\widetilde{W}_c^{\top}\widetilde{W}_c\right) + \left(-\frac{\mu}{2} + \frac{\mu\sigma_c}{4}\right)trace\left(\widehat{W}_c^{\top}\widehat{W}_c\right) + \left(\frac{\mu}{4\delta_c} + \frac{\mu}{4\sigma_c}\right)trace(W^{*\top}W^*) \tag{67}$$

We introduce the actor NN adaptation law as follows:

$$\dot{\widehat{W}}_a = -\Gamma_a\mu(\widehat{W}_a - \widehat{W}_c) \tag{68}$$

Substituting (68) into $trace\left(\widetilde{W}_a^{\top}\Gamma_a^{-1}\widehat{W}_a\right)$ yields:

$$trace\left(\widetilde{W}_a^{\top}\Gamma_a^{-1}\dot{\widehat{W}}_a\right) = trace\left(-\mu\widetilde{W}_a^{\top}(\widehat{W}_a - \widehat{W}_c)\right) \tag{69}$$

Where

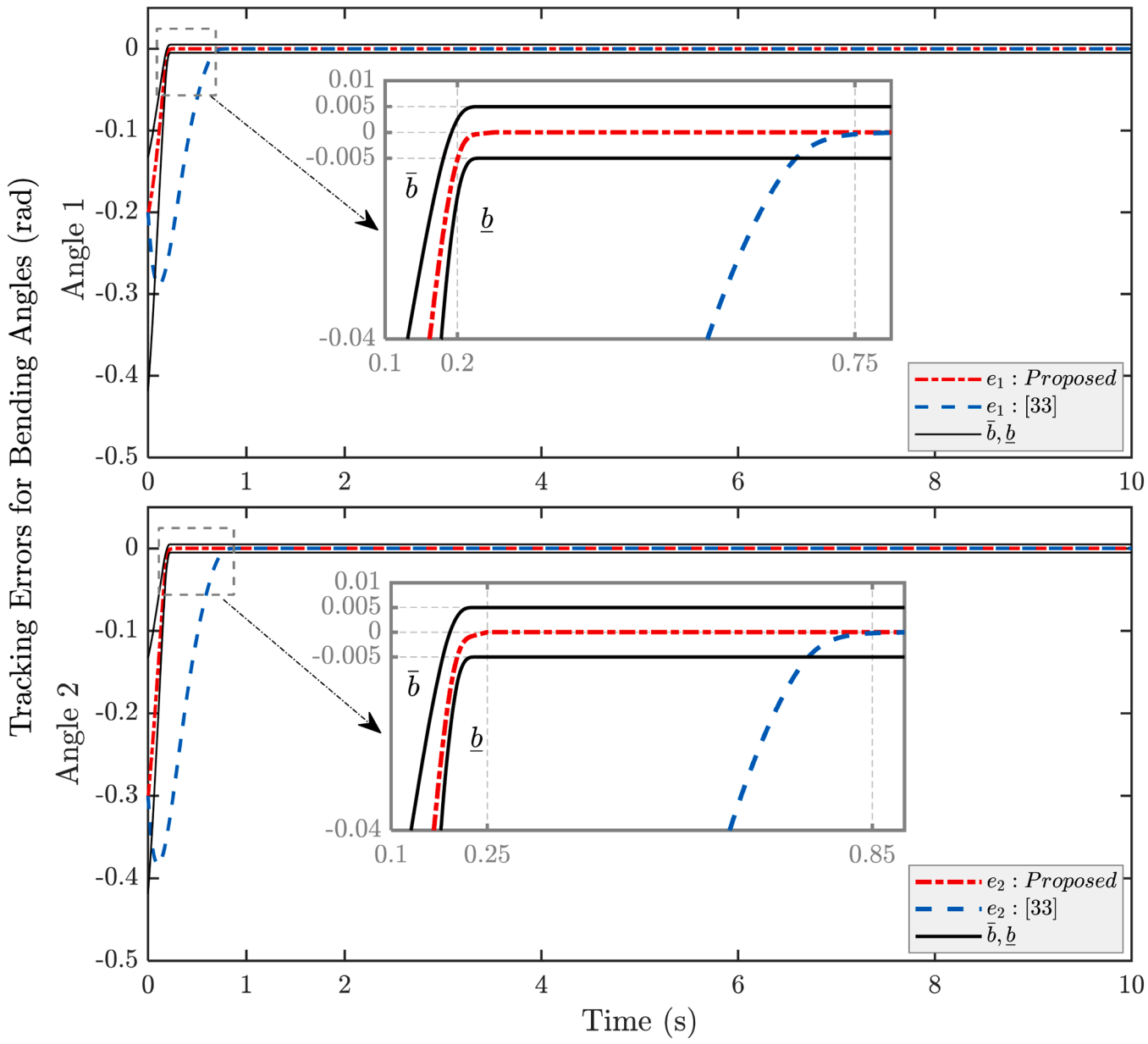


Fig. 5. Comparison of continuum robot bending angles tracking error under case 1.

$$-\mu\widetilde{W}_a^{\top}(\widehat{W}_a-\widehat{W}_c)=-\mu\widetilde{W}_a^{\top}(\widetilde{W}_a-\widetilde{W}_c)=-\mu\widetilde{W}_a^{\top}\widetilde{W}_a+\mu\widetilde{W}_a^{\top}\widetilde{W}_c \tag{70}$$

Therefore from Lemma 3 we have,

$$\begin{aligned} trace\big(-\mu\widetilde{W}_a^{\top}(\widehat{W}_a-\widehat{W}_c)\big)=-\mu trace\big(\widetilde{W}_a^{\top}\widetilde{W}_a\big)+\mu trace\big(\widetilde{W}_a\widetilde{W}_c\big)\leqslant\Big(\\ -\mu+\frac{\mu}{2\delta_a}\Big)trace\big(\widetilde{W}_a^{\top}\widetilde{W}_a\big)\\ +\frac{\mu\delta_a}{2}trace\ \big(\widetilde{W}_c^{\top}\widetilde{W}_c\big) \end{aligned} \tag{71}$$

At this time we can rewrite (62) in the following form:

$$\dot{V}\leqslant s^{\top}\Big(-\mu s+\Big(\frac{\gamma_c}{2}+\frac{\gamma_a}{2}\Big)s+r\Big(f(x)-\ddot{y}_r\Big)+\dot{r}\dot{e}+\dot{\omega}+\dot{\mu}z+\mu\dot{z}+\ r\,g(x)u_c\Big)$$

$$\begin{aligned} &+\Big(\frac{-\mu}{2}+\frac{\mu\delta_c}{4}+\frac{\mu\delta_a}{2}\Big)trace\big(\widetilde{W}_c^{\top}\widetilde{W}_c\big)+\Big(-\mu+\frac{\varrho}{2\delta_a}+\frac{\mu}{2\delta_a}\Big)trace\big(\widetilde{W}_a^{\top}\widetilde{W}_a\big)+\Big(\\ &-\frac{\mu}{2}+\frac{\mu\sigma_c}{4}+\frac{\varrho}{2\delta_c}\Big)trace\big(\widehat{W}_c^{\top}\widehat{W}_c\big)+\Big(\frac{\mu}{4\delta_c}+\frac{\mu}{4\sigma_c}\Big)trace\big(W^{*\top}W^{*}\big) \end{aligned} \tag{72}$$

Now we propose u_c as follows:

$$u_c=-(r\,g(x))^{-1}\Big(\Big(\frac{\gamma_c}{2}+\frac{\gamma_a}{2}\Big)s+r(f(x)-\ddot{y}r)+\dot{r}\dot{e}+\dot{\omega}+\dot{\mu}z+\mu\dot{z}\Big) \tag{73}$$

Where γ_c and γ_a are two positive constants. The proposed control scheme is demonstrated in Fig. 3. The stability analysis results of the proposed controller are encapsulated in the following theorem.

Theorem 1. The controllers defined in Eq. (50) and (73), in conjunction with the adaptation laws (61) and (68), ensure prescribed-time stability for the continuum robot described in Eq. (9), provided that the parameters δ_c, δ_a, γ_a, γ_c and σ_c satisfy the following conditions:

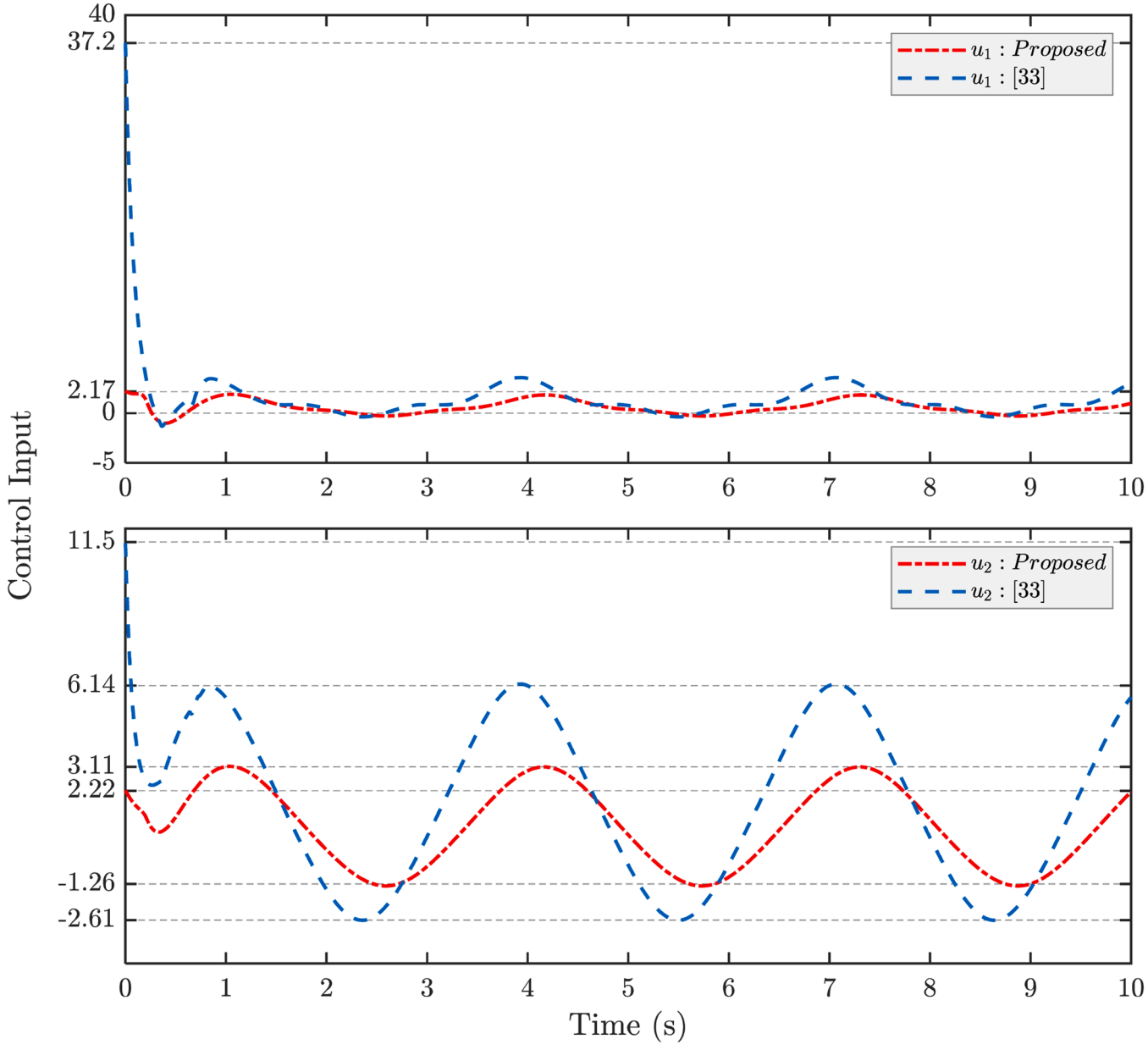


Fig. 6. Comparison of control inputs under case 1.

$$
\begin{aligned}
&-\frac{\mu}{2}+\frac{\delta c}{4}\mu+\frac{\delta_a}{2}\mu<0\\
&-\mu+\frac{\varrho}{2\gamma_a}+\frac{\mu}{2\delta_a}<0\\
&-\frac{\mu}{2}+\frac{\mu\sigma_c}{4}+\frac{\rho}{2\gamma_c}<0
\end{aligned}
$$

The sliding surface converges to the compact set $\Omega=\left\{s\in\mathbb{R}^n:\parallel s\parallel\leq\sqrt{\frac{2d}{k}}\right\}$ with $k=min\left\{2,\frac{2k_c}{\lambda_{max}\left(\Gamma_c^{-1}\right)},\frac{2k_a}{\lambda_{max}\left(\Gamma_a^{-1}\right)}\right\}$, $d=k_2 trace\left(W^{*T}W^{*}\right)$ at the prescribed-time T_p. Furthermore, the tracking error remains within a predefined monotone tube boundary for all time i.e. $\mathscr{E}:=\{x_1\in\mathbb{R}^n|\underline{b}<y-y_r<\bar{b},\ t\geq 0\}$. k_c, k_a and k_2 are two positive constants and will be defined later.

Proof. Suppose there are positive constants k_c, k_a, k_1 and k_2 such that:

$$
\begin{aligned}
&-\frac{\mu}{2}+\frac{\delta c}{4}\mu+\frac{\delta_a}{2}\mu\leq-k_c\mu\\
&-\mu+\frac{\varrho}{2\gamma_a}+\frac{\mu}{2\delta_a}\leq-k_a\mu\\
&-\frac{\mu}{2}+\frac{\mu\sigma_c}{4}+\frac{\rho}{2\gamma_c}\leq-k_1\\
&\frac{\mu}{4\delta_c}+\frac{\mu}{4\sigma_c}\leq k_2\mu
\end{aligned}
\tag{74}
$$

Based on (72) we have:

$$
\left(-\frac{\mu}{2}+\frac{\mu\sigma_c}{4}+\frac{\varrho}{2\delta_c}\right)trace\left(\widehat{W}_c^{\top}\widehat{W}_c\right)\leq-k_1 trace\left(\widehat{W}_c^{\top}\widehat{W}_c\right)\leq 0 \tag{75}
$$

Substituting (71) into (70) and then applying (73) yields:

$$
\dot{V}\leqslant-\mu s^{\top}s-k_c\mu trace\left(\widetilde{W}_c^{\top}\widetilde{W}_c\right)-k_a\mu trace\left(\widetilde{W}_a^{\top}\widetilde{W}_a\right)+k_2\mu\ trace\ \left(W^{*T}W^{*}\right) \tag{76}
$$

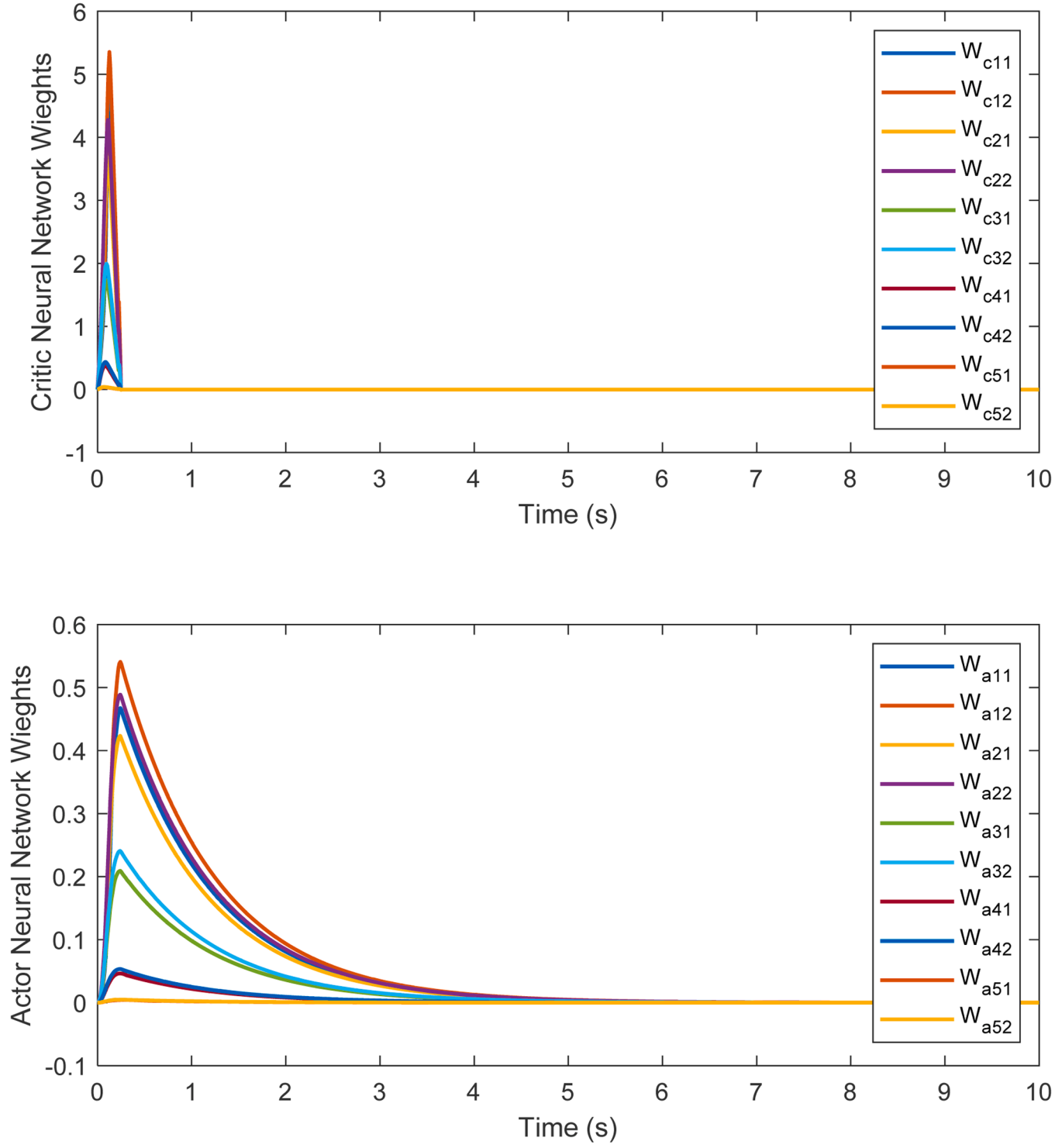


Fig. 7. Evolution of Actor and Critic NN weights in case 1.

From the definition of k and d we have:

$$\dot{V} \leqslant -k\mu V + \mu d \tag{77}$$

So based on Lemma 2, $V(t) \leqslant \frac{d}{k}$ for $t \geq T_p$ so $\| s \| \leq \sqrt{\frac{2d}{k}}$ and consequently can infer that z is bounded. Hence, based on Lemma 1 it be concluded that the tracking error remains in the predefined region $\mathscr{E} := \{x_1 \in \mathbb{R}^n | \underline{b} < y - y_r < \bar{b}, t \geq 0\}$. The proof is completed.

4. Simulation and Results

In this section, the performance of the proposed method through numerical simulations is evaluated, highlighting its unique features. The simulations utilize the mathematical model of a two-segment continuum robot as described in [33], with system parameters sourced from [34]. For a comprehensive analysis, the proposed method is compared with an alternative approach presented in [33], examining three cases with varying conditions and optimal trajectories.

Case 1: Sinusoidal Trajectory

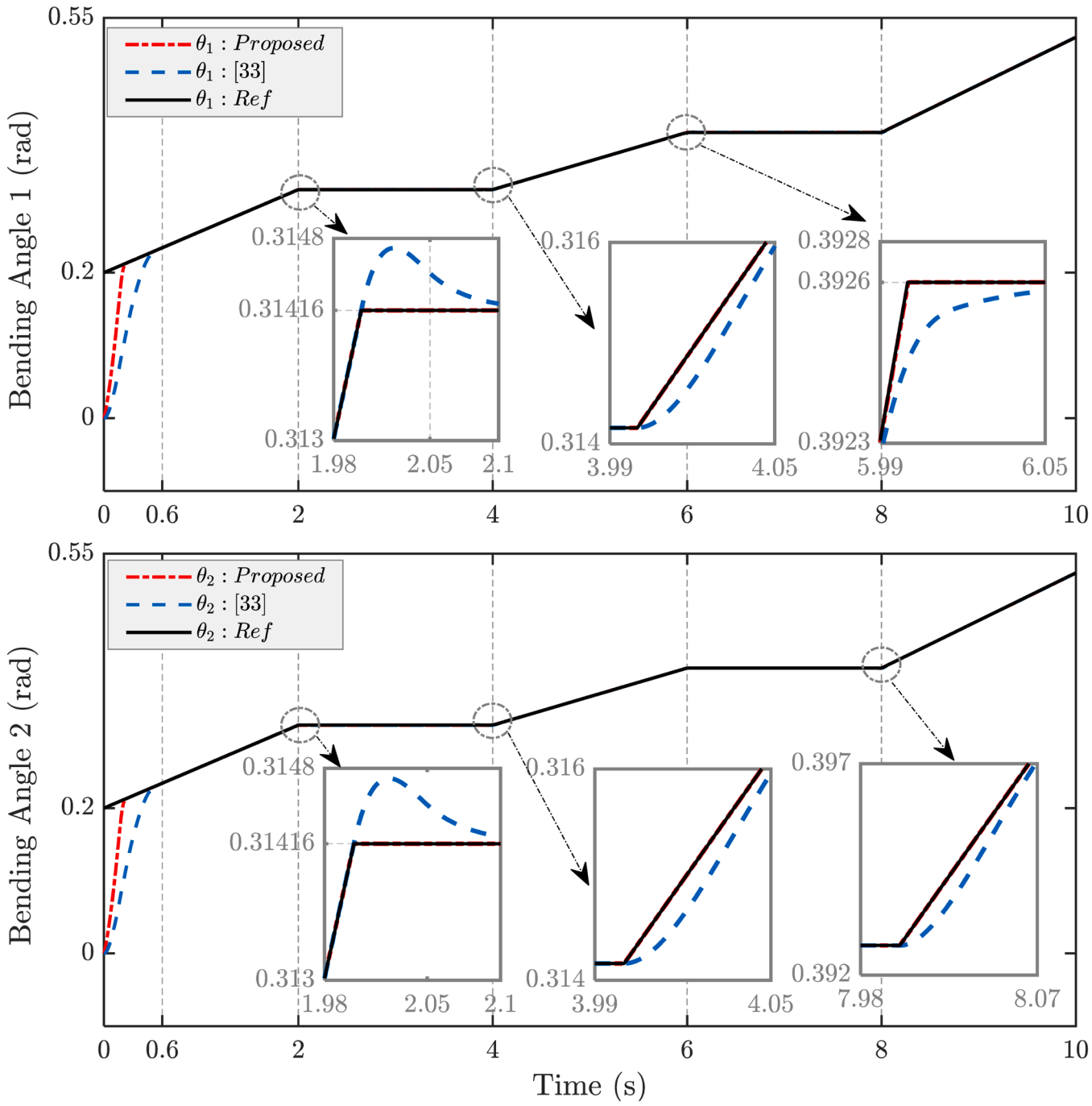


Fig. 8. Comparison of continuum robot bending angles tracking under case 2.

In this case, the continuum surgical robot is required to track a winding trajectory to perform its tasks. This trajectory is defined as the desired trajectory for the controller, represented by $y_r = [0.2 + \sin(2t), 0.3 + \sin(2t)]$. The parameters of the controller, adaptation rules and Prescribed-time are selected as $\gamma_a = \gamma_c = 1000, \Gamma_a = \Gamma_c = 1$ and $\sigma = 1$, $T_P = 0.25$ respectively. Also, the monotone tube parameters are $q_1 = q_2 = 2, q_{\infty 1} = q_{\infty 2} = 0.01, \alpha_1 = \alpha_2 = 1, T_{c1} = T_{c2} = 0.25$ and $\underline{\lambda} = \overline{\lambda} = 0.5$. For both actor and critic networks 5 neurons are chosen, the width of the Gaussian function ϱ_i are set to 1and the centers of the receptive fields m_i are evenly spaced in [-1,1]. First, we compare the tracking speed of the proposed method with that of the method presented in [33]. As illustrated in Fig. 4, the convergence speed of the proposed method is approximately 3.5 times faster than that of the method in [33]. Additionally, the zoomed-in section of Fig. 4 shows that both methods track the sinusoidal trajectory without any overshoot or undershoot.

Fig. 5 illustrates the tracking error for the bending angles of the continuum two-segment robot. As shown, the proposed method significantly outperforms the method in [33] in terms of both speed and safety. Specifically, the proposed method achieves a convergence speed nearly four times faster than the method in [33], with a settling time of approximately 0.2 seconds compared to 0.75 seconds for the method in [33]. From a safety perspective, the proposed method consistently operates within predefined safety bands ($\overline{b}$, $\underline{b}$), whereas the method in [33] fails to meet these constraints, potentially risking damage to body tissues.

Fig. 6 presents the control signals of the two methods for tracking the sinusoidal trajectory. As depicted, for both angles 1 and 2, the control effort required by the method in [33] is nearly double that of the

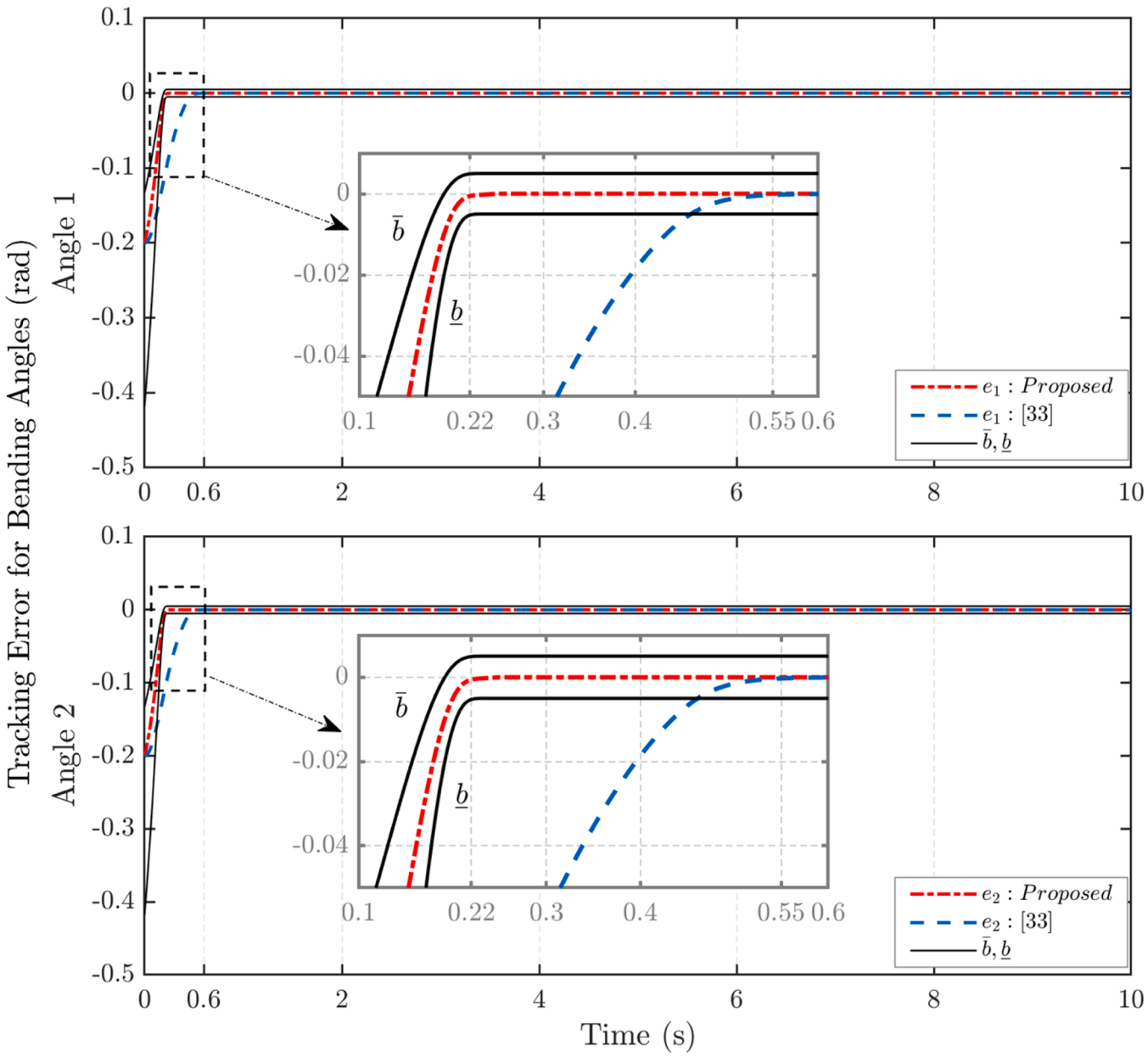


Fig. 9. Comparison of continuum robot bending angles tracking error under case 2.

proposed method, indicating higher energy consumption. Additionally, it is evident that at the initial moment, the method in [33] demands significantly more control effort compared to the proposed method. This excessive control effort at startup could potentially lead to irreparable damage during highly sensitive surgical procedures. The weights of the Actor and Critic Neural Network are depicted in Fig. 7, illustrating their evolution over time. As shown, the weights exhibit a stable adaptation process, confirming the convergence of the reinforcement learning framework. Unlike traditional offline learning approaches that rely on a predefined number of iterations, our method employs an online learning strategy, where the neural network weights are continuously updated using Lyapunov-based adaptive laws. This ensures real-time adaptation to dynamic environments without requiring a fixed dataset for training. The smooth convergence behavior in Fig. 7 validates the effectiveness of the proposed learning mechanism in achieving stable and optimal control performance.

Case 2: Step-ramp Trajectory

In the second case, the objective is to evaluate the performance of the proposed method under conditions where the desired trajectory exhibits abrupt changes. This trajectory serves as a rigorous test of the controller's capability to maintain the surgeon robot on the desired path, thereby preventing any deviation that could potentially damage body tissues. In this simulation, the controller parameters, adaptive rules, prescribed-time, and monotone tube parameters are consistent with previous settings to effectively demonstrate the controller's robustness. Despite the increased complexity of the reference trajectory, the controller maintains consistent performance without the need for retuning, demonstrating low sensitivity to parameter changes and ensuring reliable tracking across different conditions. Fig. 8 illustrates the tracking performance of the surgical robot using the proposed method compared to the method in [33]. As clearly shown, at the onset of the trajectory, the proposed method achieves tracking at approximately twice the speed of the method in [33]. Throughout the trajectory, particularly during abrupt changes at seconds 2, 4, 6, and 8, the proposed method, enhanced with reinforcement learning capabilities, maintains adherence to the desired path without any deviation. In

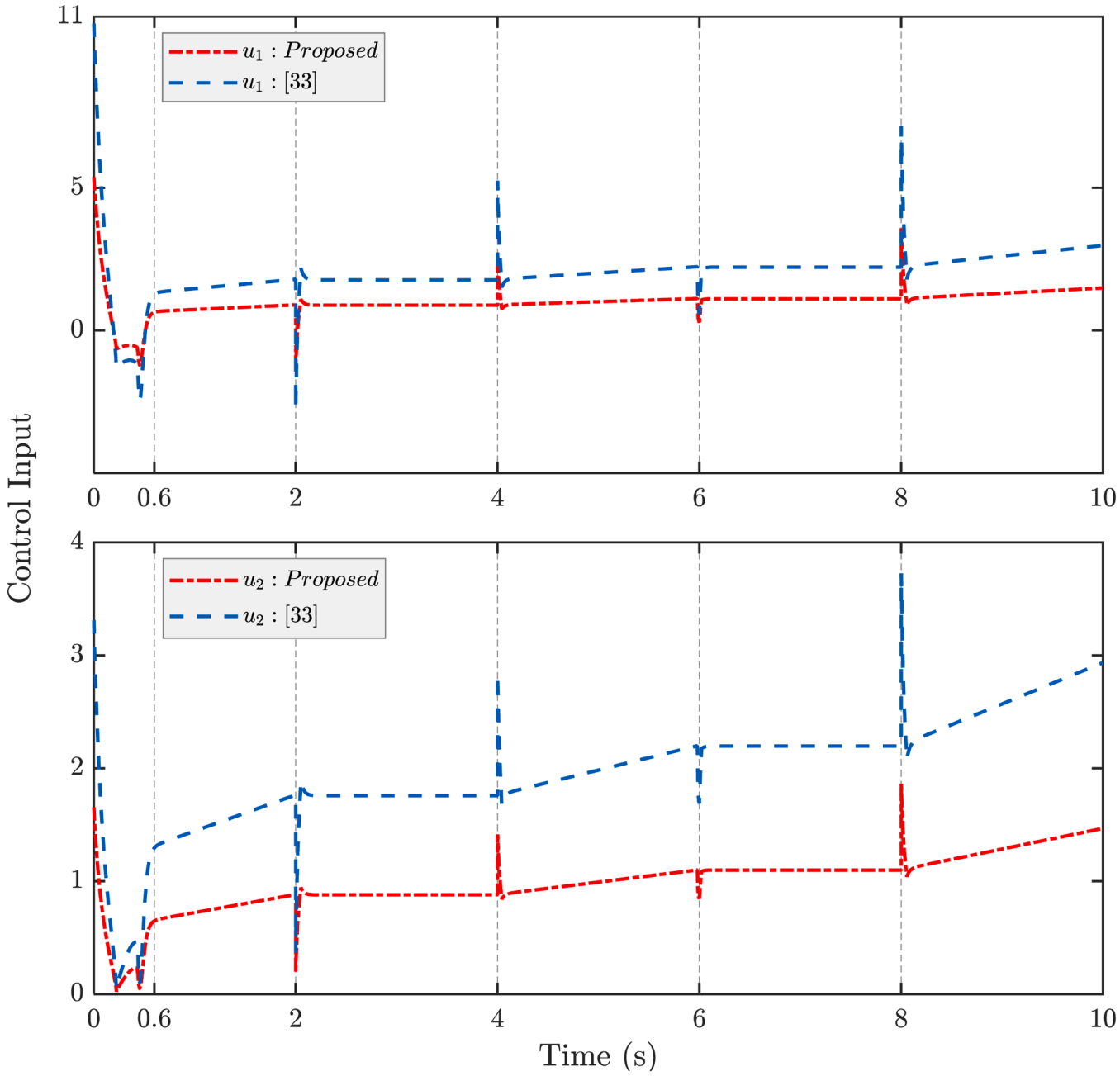


Fig. 10. Comparison of control inputs under case 2.

contrast, the method in [33] exhibits momentary deviations from the desired trajectory.

Given the paramount importance of safety in surgical procedures, a detailed comparison of the two methods is essential. A closer examination reveals that the proposed method, utilizing the monotone tube feature, consistently remains within the target range. Conversely, the method described in [33] often exceeds the safe range, which could potentially harm body tissues and jeopardize patient safety (Fig. 9). Furthermore, the proposed method, with its prescribed-time feature, reduces the tracking error to near zero at a rate approximately three times faster than the method in [33].

Energy consumption of continuum robots is a critical area of research, with many efforts focused on its optimization. The proposed method effectively addresses this issue. As illustrated in Fig. 10, the control signal magnitude of the proposed method is approximately half that of the method in [33]. Consequently, the energy consumption of the robot under the proposed method is significantly reduced compared to the method in [33]. Additionally, during instances of trajectory changes, the control effort required by the method in [33] is substantially higher, which can increase the risk of undesirable outcomes, such as potential damage to the robot's actuators or harm to delicate tissues in surgical applications. While it is challenging to define a precise energy consumption threshold that would directly lead to damage, the increased control effort, particularly in response to trajectory changes, can elevate the risk of these negative effects. This comparison highlights the relative efficiency and safety of our proposed method, which requires less control effort and is therefore less likely to cause undue stress on the system or harm in sensitive applications. The intention of this statement is to emphasize the potential risks associated with higher control demands, rather than to imply a specific energy threshold.

Case 3: Monotone Tube and Prescribed-time Control Parameter Effect

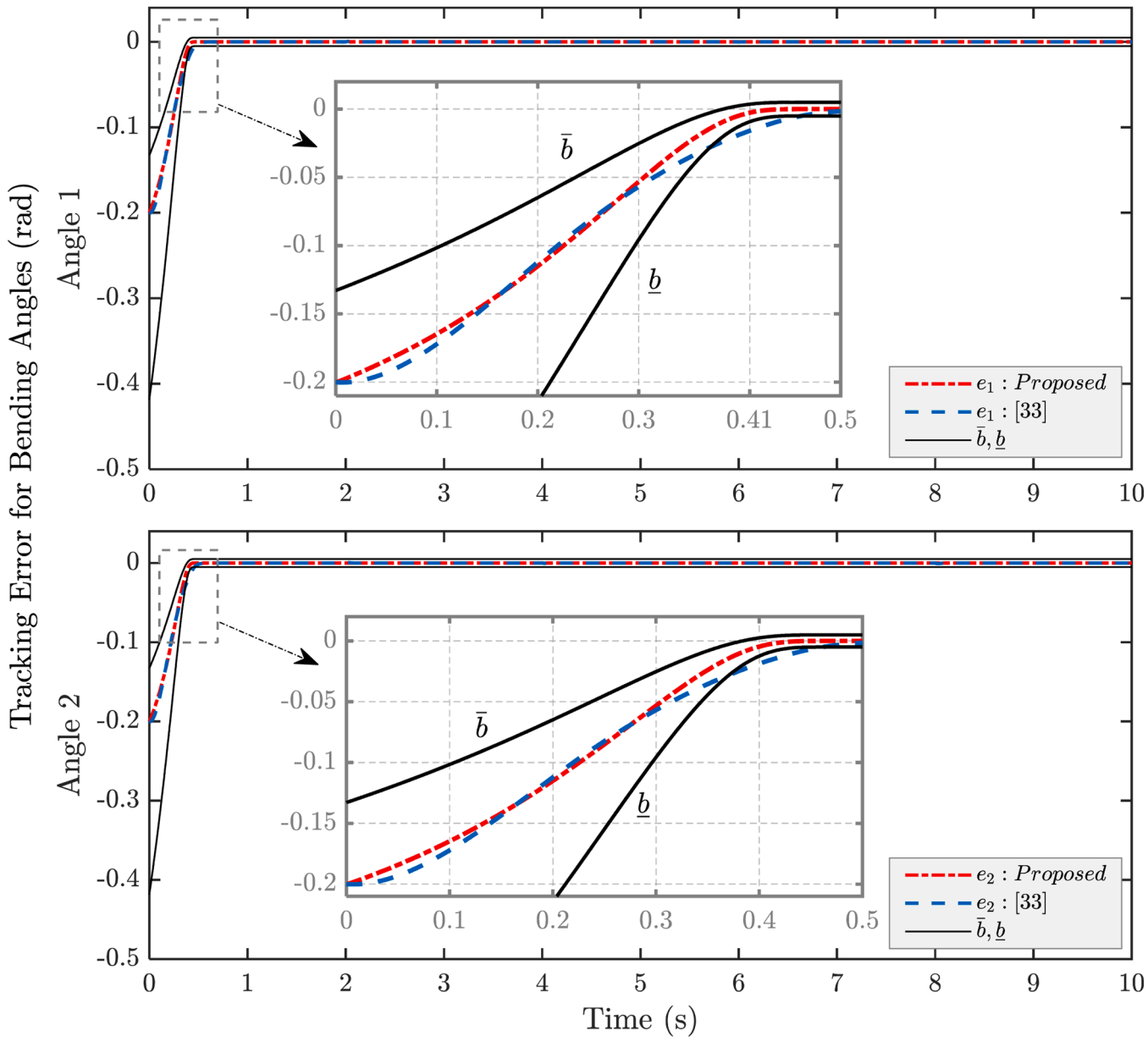


Fig. 11. Comparison of continuum robot bending angles tracking error under case 3.

The monotone tube and prescribed-time control method incorporated in the proposed controller facilitate both the adjustment of convergence speed and the maintenance of safety within the robotic system during surgical procedures. In certain surgical scenarios, rapid response from the robot is crucial. This section aims to demonstrate the capability of the proposed controller to modulate the speed of the surgeon robot. To achieve this, the monotone tube and prescribed-time control parameters are recalibrated to set the settling time to 0.5 seconds ($T_{c1} = T_{c2} = T_p = 0.5$). The remaining parameters are selected as he previous cases. Fig. 11 illustrates the convergence of the reference trajectory using the proposed methods and the method in [33]. A comparison between Fig. 8 and Fig. 11 reveals that the convergence speed has been adjusted, with the settling time modified from approximately 0.2 seconds to 0.4 seconds. The capability to set the settling time to a desired value within a specified range is a distinctive feature of the proposed method, as clearly demonstrated in Fig. 11. This section aims to evaluate whether altering the monotone tube and prescribed-time control parameters and reducing the convergence speed impacts the system's performance. Additionally, a comparison between Figs. 8 and 12 demonstrates that modifying the controller parameters to adjust the settling time does not impact the system's performance in the steady-state.

5. Conclusion

This paper has introduced a novel prescribed-time control approach for continuum robots, addressing critical aspects such as system safety, transient and steady-state tracking, and optimal control through reinforcement learning. The proposed method effectively transforms system state constraints on tracking error into an unconstrained problem using a monotone tube boundary, thereby circumventing the complexities and limitations associated with traditional techniques like MPC, CBF, and BLF. This approach allows for the pre-assignment of transient characteristics and settling time, independent of initial conditions, thereby enhancing precision and convergence rates essential for surgical applications. The optimized prescribed-time control strategy, leveraging an actor-critic neural network, ensures controller optimality by significantly reducing control effort, power consumption, and heat generation

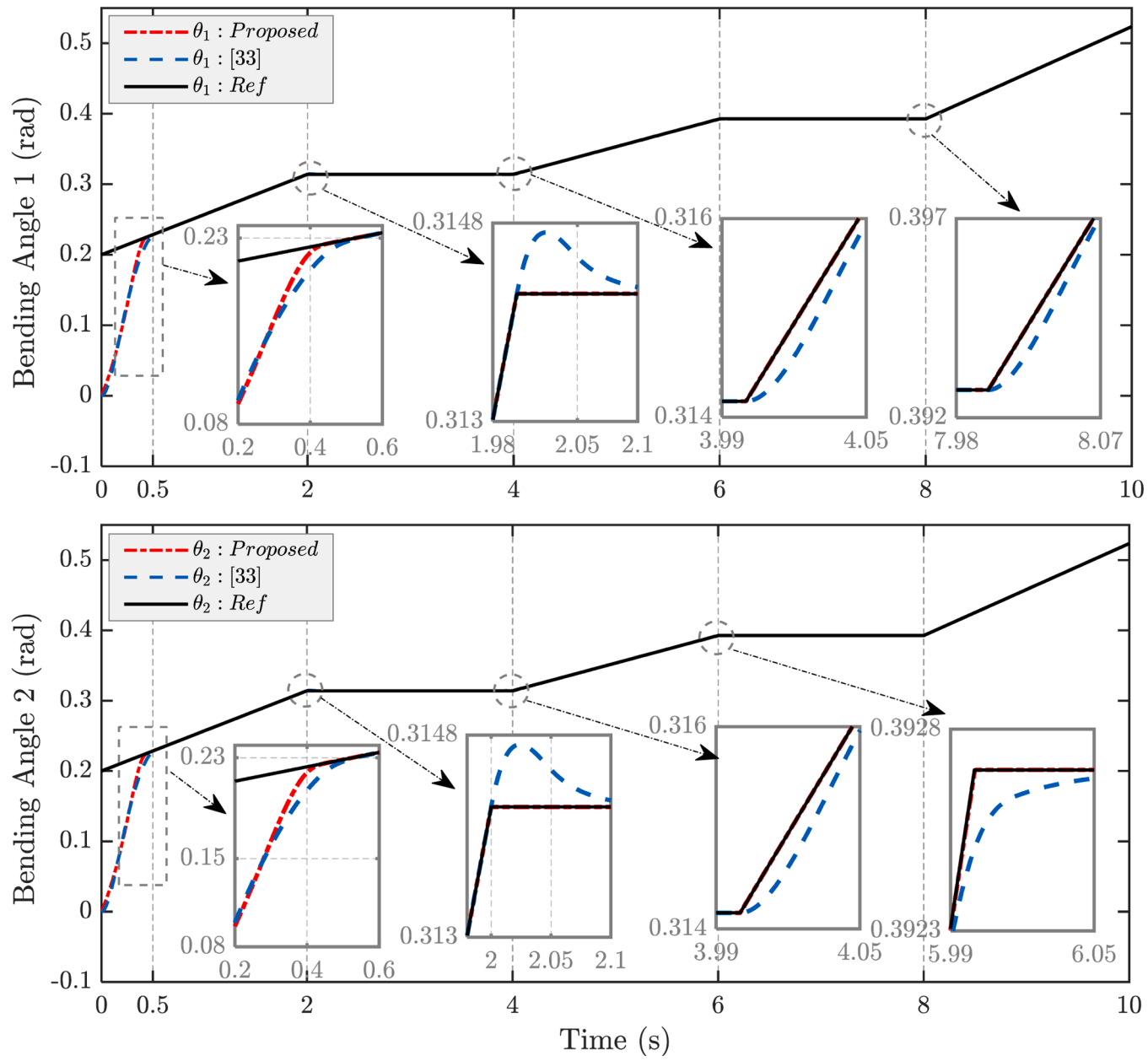


Fig. 12. Comparison of continuum robot bending angles tracking under case 3.

in the robot's actuators. This is particularly crucial when the robot is in contact with a patient's body. Furthermore, the method's adaptability to unfamiliar and dynamic environments ensures robust performance across various surgical scenarios. Simulation results clearly demonstrate the powerful features of the proposed method, including the ability to set the settling time to the desired value, eliminate overshoot and undershoot, maintain the tracking signal within the desired path, and keep the tracking error within a limited bound. The proposed method also enables high-speed surgical operations in unknown environments, leveraging reinforcement learning to maintain all constraints even in the presence of uncertainty and disturbances. This advancement in control methodology holds significant promise for enhancing the safety, accuracy, and efficiency of continuum robots in surgical procedures, ultimately contributing to improved patient outcomes. Future work could explore several promising directions. Challenges such as actuator or sensor fault detection and compensation, output feedback control, obstacle avoidance, optimal path generation as a desired trajectory, and considering full state constraints are attractive problems that could be addressed by modifying the proposed method. Additionally, investigating the integration of advanced machine learning techniques for real-time adaptation and the development of robust control strategies for multi-robot systems could further enhance the capabilities of continuum robots in complex surgical environments.

CRediT authorship contribution statement

Mohammad Jabari: Writing – original draft, Visualization, Software, Methodology, Investigation, Conceptualization. **Andrea Botta:** Writing – review & editing, Methodology, Investigation, Data curation. **Luigi Tagliavini:** Writing – review & editing, Methodology, Investigation, Data curation. **Carmen Visconte:** Writing – review & editing, Validation, Supervision, Resources, Data curation, Conceptualization. **Giuseppe Quaglia:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Mohammad Jabari reports financial support was provided by PNRR-NGEU. This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 351/2022. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] P. Rezaei, S.Y. Lee, K. Cho, J.O. Hahn, Robust Control of Exo-Abs, a Wearable Platform for Ubiquitous Respiratory Assistance, Journal of Dynamic Systems, Measurement, and Control 147 (2) (2025).

[2] D. Bhandarkar, G. Mittal, R. Shah, A. Katara, T.E. Udwadia, Single-incision laparoscopic cholecystectomy: How I do it? Journal of minimal access surgery 7 (1) (2011) 17–23.

[3] E.A. Arkenbout, P.W. Henselmans, F. Jelínek, P. Breedveld, A state of the art review and categorization of multi-branched instruments for NOTES and SILS, Surgical endoscopy 29 (2015) 1281–1296.

[4] J. Burgner-Kahrs, D.C. Rucker, H. Choset, Continuum robots for medical applications: A survey, IEEE transactions on robotics 31 (6) (2015) 1261–1280.

[5] Z. Wang, T. Wang, B. Zhao, Y. He, Y. Hu, B. Li, P. Zhang, M.Q.H. Meng, Hybrid adaptive control strategy for continuum surgical robot under external load, IEEE Robotics and Automation Letters 6 (2) (2021) 1407–1414.

[6] J. Lai, K. Huang, B. Lu, Q. Zhao, H.K. Chu, Verticalized-tip trajectory tracking of a 3D-printable soft continuum robot: Enabling surgical blood suction automation, IEEE/ASME Transactions on Mechatronics 27 (3) (2021) 1545–1556.

[7] L. Zhang, J. Ma, L. Zang, F. Dong, A. Lu, B. Feng, Z. He, H. Hong, M. Zheng, Prevention and management of hemorrhage during a laparoscopic colorectal surgery, Annals of Laparoscopic and Endoscopic Surgery 1 (7) (2016).

[8] N. Simaan, R.M. Yasin, L. Wang, Medical technologies and challenges of robot-assisted minimally invasive intervention and diagnostics, Annual Review of Control, Robotics, and Autonomous Systems 1 (1) (2018) 465–490.

[9] F. Qi, B. Chen, S. Gao, S. She, Dynamic model and control for a cable-driven continuum manipulator used for minimally invasive surgery, The International Journal of Medical Robotics and Computer Assisted Surgery 17 (3) (2021) e2234.

[10] A. Ghoul, K. Kara, M. Benrabah, B. Nasri, Control of Continuum Robot Using Two Optimized PID Controller, in: Multi Conference on Electrical Engineering CEE, 2021.

[11] A. Yazdanpanah, F. Piltan, A. Roshanzamir, M. Mirshekari, N.G. Mozafari, Design PID baseline fuzzy tuning proportional-derivative coefficient nonlinear controller with application to continuum robot, International Journal of Intelligent Systems and Applications 6 (5) (2014) 90.

[12] Y. Wang, Y. Qian, W. Liu, A Novel Model Predictive Control Strategy for Continuum Robot: Optimization and Application, in: The International Conference on Applied Nonlinear Dynamics, Vibration and Control, Singapore: Springer Nature Singapore, 2023, pp. 484–497.

[13] B. Ouyang, H. Mo, H. Chen, Y. Liu, D. Sun, Robust model-predictive deformation control of a soft object by using a flexible continuum robot, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 613–618.

[14] S.M. Ebrahimi, F. Norouzi, H. Dastres, R. Faieghi, M. Naderi, M. Malekzadeh, Sensor fault detection and compensation with performance prescription for robotic manipulators, Journal of the Franklin Institute 361 (7) (2024) 106742.

[15] S.S. Amini, A.K. Khalaji, Trajectory Tracking Control of a Novel Planner Continuum Robot, International Journal of Advanced Design & Manufacturing Technology 15 (4) (2022).

[16] M.S. Sofla, M.J. Sadigh, S.H. Sadati, C. Bergeles, M. Zareinejad, Sliding-surface dynamic control of a continuum manipulator with large workspace, Control Engineering Practice 141 (2023) 105680.

[17] A. Abu Alqumsan, S. Khoo, M. Norton, Multi-surface sliding mode control of continuum robots with mismatched uncertainties, Meccanica 54 (14) (2019) 2307–2316.

[18] H. Dastres, S.M. Ebrahimi, M. Malekzadeh, F. Gordillo, Robust adaptive parameter estimator design for a multi-sinusoidal signal with fixed-time stability and guaranteed prescribed performance boundary of estimation error, Journal of the Franklin Institute 360 (1) (2023) 223–250.

[19] M.C. Yip, D.B. Camarillo, Model-less hybrid position/force control: a minimalist approach for continuum manipulators in unknown, constrained environments, IEEE Robotics and Automation Letters 1 (2) (2016) 844–851.

[20] H. Dastres, A. Mohammadi, B. Rezaie, Adaptive robust control design to maximize the harvested power in a wind turbine with input constraint, Journal of Renewable Energy and Environment 7 (4) (2020) 30–43.

[21] V. Teymoori, H. Dastres, M.J. Kamper, R.J. Wang, N. Arish, Enhanced Fast Terminal Sliding Mode Observer for Wide-Speed Sensorless Control of PM Vernier Ship Propulsion Machine Drives, in: 2023 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2023 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), IEEE, 2023, pp. 1–7.

[22] X. Liang, G. He, T. Su, W. Wang, C. Huang, Q. Zhao, Z.G. Hou, Finite-time observer-based variable impedance control of cable-driven continuum manipulators, IEEE Transactions on Human-Machine Systems 52 (1) (2021) 26–40.

[23] M. Golestani, R. Chhabra, M. Esmaeilzadeh, Finite-time Nonlinear H∞ Control of Robot Manipulators with Prescribed Performance, IEEE Control Systems Letters 7 (2023) 1363–1368.

[24] M. Russo, S.M.H. Sadati, X. Dong, A. Mohammad, I.D. Walker, C. Bergeles, K. Xu, D.A. Axinte, Continuum robots: An overview, Advanced Intelligent Systems 5 (5) (2023) 2200367.

[25] H. Dastres, A. Mohammadi, M. Shamekhi, A neural network based adaptive sliding mode controller for pitch angle control of a wind turbine, in: 2020 11th Power Electronics, Drive Systems, and Technologies Conference (PEDSTC), IEEE, 2020, pp. 1–6.

[26] Y. Su, C. Zheng, P. Mercorelli, Robust approximate fixed-time tracking control for uncertain robot manipulators, Mechanical Systems and Signal Processing 135 (2020) 106379.

[27] M. Wang, A. Yang, Dynamic learning from adaptive neural control of robot manipulators with prescribed performance, IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (8) (2017) 2244–2255.

[28] H. Ma, Q. Zhou, H. Li, R. Lu, Adaptive prescribed performance control of a flexible-joint robotic manipulator with dynamic uncertainties, IEEE Transactions on Cybernetics 52 (12) (2021) 12905–12915.

[29] Y. Shi, B. Yi, W. Xie, W. Zhang, Enhancing prescribed performance of tracking control using monotone tube boundaries, Automatica 159 (2024) 111304.

[30] Z. Liu, C. Lin, Y. Shang, Prescribed-time adaptive neural feedback control for a class of nonlinear systems, Neurocomputing 511 (2022) 155–162.

[31] G. Wen, C.P. Chen, Optimized backstepping consensus control using reinforcement learning for a class of nonlinear strict-feedback-dynamic multi-agent systems, IEEE Transactions on Neural Networks and Learning Systems 34 (3) (2021) 1524–1536.

[32] M.S. Qureshi, P. Swarnkar, S. Gupta, A supervisory on-line tuned fuzzy logic based sliding mode control for robotics: An application to surgical robots, Robotics and Autonomous Systems 109 (2018) 68–85.

[33] M. Ding, H. Wu, X. Zheng, Y. Guo, Adaptive predefined-time attitude stabilization control of space continuum robot, IEEE Transactions on Circuits and Systems II: Express Briefs 71 (2) (2022) 647–651.

[34] A. Ghoul, K. Kara, M. Benrabah, M.L. Hadjili, Optimized nonlinear sliding mode control of a continuum robot manipulator, Journal of Control, Automation and Electrical Systems 33 (5) (2022) 1355–1363.

[35] A. Amouri, A. Zaatri, C. Mahfoudi, Dynamic modeling of a class of continuum manipulators in fixed orientation, Journal of Intelligent & Robotic Systems 91 (3) (2018) 413–424.

[36] Y. Shi, B. Yi, W. Xie, W. Zhang, Enhancing prescribed performance of tracking control using monotone tube boundaries, Automatica 159 (2024 Jan 1) 111304.

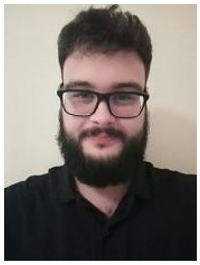
[37] Z. Liu, C. Lin, Y. Shang, Prescribed-time adaptive neural feedback control for a class of nonlinear systems, Neurocomputing 511 (2022 Oct 28) 155–162.

[38] K.G. Vamvoudakis, FL. Lewis, Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica 46 (5) (2010 May 1) 878–888.

[39] S.A. Rizvi, Z. Lin, Output Feedback Reinforcement Learning Control for Linear Systems, Birkhäuser, 2023.

[40] J. Wang, C. Wang, M. Xin, Z. Ding, J. Shan, Cooperative Control of Multi-Agent Systems: An Optimal and Robust Perspective, Academic Press, 2020.

Mohammad Jabari is a Ph.D. student at the University of Genova, Department of Informatics, Bioengineering, Robotics and Systems Engineering (DIBRIS), enrolled in the National Ph. D. Program in Robotics and Intelligent Machines. He conducts his research at the Department of Mechanical and Aerospace Engineering (DIMEAS), Politecnico di Torino, Italy. He received his B.Sc. in Electrical Engineering — Electrical Power from Malayer University in 2014, and his M.Sc. in Mechatronics Engineering from the University of Tabriz in 2019. His research interests include robotics and automation, control and optimization of continuum robots, and the development of innovative robotic systems for hand rehabilitation.



Andrea Botta is a research fellow at Politecnico di Torino, Italy. He obtained his Ph.D. in Mechanical Engineering in 2022 from Politecnico di Torino. He received his B.Sc. in Mechanical Engineering in 2014 and completed his M.Sc. in Mechatronic Engineering in 2017, both at Politecnico di Torino. His research activities focus on sustainable development and the achievement of the Sustainable Development Goals. In particular, his research interests include the design and development of innovative mobile service robots, as well as assistive and rehabilitative devices for the elderly, individuals with disabilities, and those recovering from injuries

Luigi Tagliavini is a postdoctoral research assistant in the Department of Mechanical and Aerospace Engineering (DIMEAS) at Politecnico di Torino, Italy. He received his B.Sc. in Mechanical Engineering in 2018 from Università degli Studi di Ferrara, his M.Sc. in Mechanical Engineering in 2020 from Politecnico di Torino, and his Ph.D. in Mechanical Engineering in 2024, also from Politecnico di Torino. His research interests lie in the design and development of innovative mobile service robots and assistive technologies.

Carmen Visconte received her Ph.D. in Applied Mechanics from Politecnico di Torino in 2004. She is currently an Associate Professor of Applied Mechanics in the Department of Mechanical and Aerospace Engineering at Politecnico di Torino. Since 2018, she has been a member of the PoliTO Interdepartmental Centre for Service Robotics (PIC4SeR), where she has expanded her research to include service robotics, with a specific focus on soft devices.

Giuseppe Quaglia is a Full Professor of Applied Mechanics at Politecnico di Torino, Italy. His research activities focus on energy saving, renewable energy, sustainability, vehicle dynamics and systems, robotics and mechatronics, automation, actuators, and mechanisms. He is also actively involved in the development of devices for people with disabilities and biomedical applications, as well as technologies and systems that support sustainable human development. He serves as Chair of the Technical Committee for Sustainable Energy Systems of IFToMM, a member of the SIRI Council (Italian Association of Robotics and Automation), and a member of the Management Board of the PoliTO Interdepartmental Centre for Service Robotics (PIC4SeR). He also chairs IFToMM's Cross-Disciplinary Group 1: Securing Our Future Environment — Air, Water, Energy. In addition, he is the coordinator of the Inclusion and Equal Opportunities Green Team at Politecnico di Torino and the director of the Politecnico di Torino Japan Hub.