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

Deep Reinforcement Learning-Based Motion Planning and PDE Control for Flexible Manipulators

Amir Hossein Barjini, Seyed Adel Alizadeh Kolagar, Sadeq Yaqubi, and Jouni Mattila

***Abstract*—This article presents a motion planning and control framework for flexible robotic manipulators, integrating deep reinforcement learning (DRL) with a nonlinear partial differential equation (PDE) controller. Unlike conventional approaches that focus solely on control, we demonstrate that the desired trajectory significantly influences endpoint vibrations. To address this, a DRL motion planner, trained using the soft actor-critic (SAC) algorithm, generates optimized trajectories that inherently minimize vibrations. The PDE nonlinear controller then computes the required torques to track the planned trajectory while ensuring closed-loop stability using Lyapunov analysis. The proposed methodology is validated through both simulations and real-world experiments, demonstrating superior vibration suppression and tracking accuracy compared to traditional methods. The results underscore the potential of combining learning-based motion planning with model-based control for enhancing the precision and stability of flexible robotic manipulators.**

***Index Terms*—Flexible robotics, integrated planning and control, motion control, reinforcement learning.**

I. Introduction

MANIPULATORS are widely used in robotic operations and industrial automation. Low energy consumption and lightweight design are the primary reasons for the growing popularity of flexible robotic manipulators. Despite these advantages, the inherent flexibility of these robots complicates modeling and control [1], [2], [3]. In general, manipulator flexibility refers to either flexible links [4] or flexible joints [5].

A. Related Works

A review of the literature reveals various approaches for controlling flexible manipulators,including hardware-based methods such as actuation via cables [6], and software-based strategies like visual servoing [7]. For vibration suppression, some studies have modeled a fixed-free Euler-Bernoulli beam and aimed to control theendpoint vibrations using an input force [8], [9]. However, since it is important to simultaneously track the desired position and suppress vibrations in flexible manipulators, some studies have proposed boundary observer-based control with Lyapunov stability guarantees, applying a force for vibration suppression and a torque for tracking. Despite achieving good results in vibration suppression, these methods are limited to simulations, and actuation constraints remain a challenge for implementing input force in real-world experiments [10], [11]. An alternative, experimentally feasible approach is to use a single input torque applied to control the system's angular position [12]. However, controlling endpoint vibrations in such an under-actuated system poses greater challenges [13].

Various approaches have been adopted to tackle these challenges. Considering control strategies, some studies have utilized partial differential equations (PDEs) to design controllers [14], [15], while others have transformed the PDEs into ordinary differential equations (ODEs), using methods such as assumed mode method (AMM), and designed controllers based on the simplified equations [16], [17]. Although designing a controller based on a reduced-order ODE model results in simplified controller design, it can negatively affect the precision of the designed controller at higher frequencies [18].

From the motion planning perspective, extensive research has been conducted to optimize manipulator trajectories. However, most studies assume that manipulators have rigid-body links [19], [20]. Although the motion of flexible manipulators significantly influences system vibrations, there is still a lack of comprehensive research addressing this aspect. The advancement of artificial intelligence has led to the emergence of learning-based methods for manipulator motion planning in unknown environments. Among these, reinforcement learning (RL) has become a prominent approach [19], enabling agents to interact with their environment, encounter diverse scenarios, and learn optimal actions through reward-based mechanisms [21]. In recent years, with the progress of deep learning, traditional RL has evolved into deep reinforcement learning (DRL) [22]. By leveraging deep neural networks, DRL can handle high-dimensional, continuous state and action spaces, allowing robots to tackle more complex tasks such as navigation, adaptive motion planning, and task automation [23], [24]. Among the various DRL algorithms, the Soft Actor-Critic (SAC) algorithm stands out for its superior performance in high-dimensional motion planning problems [25]. By incorporating an entropy term into its objective function, SAC enhances exploration and improves learning efficiency, enabling it to outperform existing methods in complex planning tasks [24]. This could be especially useful for

Received 28 February 2025; accepted 27 June 2025. Date of publication 10 July 2025; date of current version 17 July 2025. This article was recommended for publication by Associate Editor S. Han and Editor C. D. Santina upon evaluation of the reviewers' comments. This work was supported by the Research Council of Finland through the Project "Nonlinear PDE-model-based control of flexible manipulators" under Grant 355664. *(Corresponding author: Sadeq Yaqubi.)*

The authors are with the Department of Engineering and Natural Sciences, Tampere University, 33100 Tampere, Finland (e-mail: amirhossein.barjini@tuni.fi; sadeq.yaqubi@tuni.fi).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3588057, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3588057

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

flexible-link manipulators, where nonlinearities and underactuation require broad exploration. SAC's stochastic policy exploits passive dynamics, and its entropy maximization ensures robust, stable learning in complex systems, making it ideal for dynamic, uncertain environments.

In recent years, DRL has been widely explored for motion planning in rigid-body manipulators, with approaches ranging from end-to-end frameworks that directly output joint torques [26] to methods that combine DRL with low-level controllers [27], [28]. However, to the best of our knowledge, the application of DRL to motion planning for flexible-link manipulators remains largely unexplored.

B. Contributions

The main contributions of this letter can be summarized as follows: 1) Development of a nonlinear PDE controller for flexible manipulators that ensures closed-loop stability via Lyapunov theory, enabling precise trajectory tracking using only a single input torque at the base. 2) Introduction of a DRL motion planner for flexible manipulators, leveraging the SAC algorithm to generate optimal trajectories that actively suppress endpoint vibrations. 3) Validation of the proposed synergistic model-based PDE and model-free DRL approach, effectively handling underactuation without endpoint actuation unlike traditional boundary control methods. 4) Validation of the proposed motion planning and control framework through simulations and real-world experiments on a hydraulically actuated flexible robotic manipulator, demonstrating superior performance compared to conventional control and motion planning methods.

This letter is organized as follows: Section II presents the mathematical modeling of the flexible manipulator, including the governing PDEs. In Section III, the proposed nonlinear PDE controller is introduced, followed by the development of the DRL motion planner. Section IV evaluates the proposed method through numerical simulations. Section V provides experimental validations on a real flexible robotic manipulator to verify the proposed method's effectiveness. Both the PDE controller and DRL motion planner are tested under real-world conditions. Finally, Section VI summarizes the key findings and explores future research directions.

II. Problem Formulation

In this chapter, we derive the PDE model for the flexible robotic manipulator. This model forms the foundation for both the model-based controller and the design of the motion planner for the system. Consequently, the accuracy of the model plays a crucial role in the reliability of the experimental results. The manipulator, a single-link system, operates in the vertical plane, where the system and payload's weight make gravity an influential factor. A schematic representation of the flexible manipulator is shown in Fig. 1.

The system is modeled as an infinite-dimensional system using Euler–Bernoulli beam theory, where XOY and xOy denote the inertial and rotating coordinate systems, respectively. The variables M, θ, ω, and τ represent the mass of the payload, the angular position of the flexible link, the elastic deviation along the link, and the input torque, respectively. Assuming small deformations, the arc position of each segment of the flexible link is expressed as $y(x,t) = x\theta(t) + \omega(x,t)$. Throughout this letter, a dot denotes a time derivative $\left([\dot{-}] = \frac{\partial[-]}{\partial t}\right)$, while a prime represents a spatial derivative $\left([-]' = \frac{\partial[-]}{\partial x}\right)$.

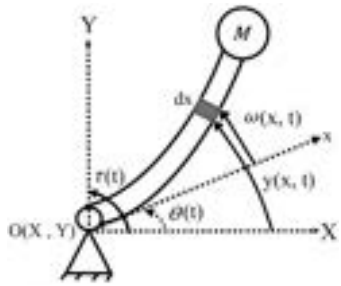

Fig. 1. Schematic View of the Flexible Manipulator.

To derive the equations of motion, the kinetic energy (T), potential energy (U), and virtual work done by external torque (W), can be expressed as:

$$T = \frac{1}{2}I_m\dot{\theta}^2(t) + \frac{1}{2}\rho A\int_0^L \dot{y}^2(x,t)dx + \frac{1}{2}M\dot{y}^2(L,t), \tag{1}$$

$$\begin{aligned} U &= \frac{1}{2}mgL\sin\theta(t) + Mg(L\sin\theta(t) + \omega(L,t)\cos\theta(t)) \\ &\quad + \frac{1}{2}EI\int_0^L [\omega''(x,t)]^2dx \\ &= \frac{1}{2}mgL\sin\theta(t) + MgL\sin\theta(t) \\ &\quad + Mg(y(L,t) - L\theta(t))\cos\theta(t) + \frac{1}{2}EI\int_0^L [y''(x,t)]^2dx, \end{aligned} \tag{2}$$

$$W = \tau(t)\theta(t), \tag{3}$$

where I_m, ρ, A, m, L, and EI denote the moment of inertia, density, cross-sectional area, mass, length, and bending stiffness of the flexible link, respectively.

Now, applying the extended Hamilton's principle, as $\int_{t_1}^{t_2}(\delta T - \delta U + \delta W)dt = 0$, the governing equations of motion and the corresponding boundary conditions are derived as follows:

$$\begin{aligned} &I_m\ddot{\theta}(t) - EI\omega''(0,t) + \frac{1}{2}mgL\cos\theta(t) \\ &\quad - Mg\omega(L,t)\sin\theta(t) = \tau(t), \end{aligned} \tag{4}$$

$$\rho Ax\ddot{\theta}(t) + \rho A\ddot{\omega}(x,t) + EI\omega''''(x,t) = 0, \tag{5}$$

$$ML\ddot{\theta}(t) + M\ddot{\omega}(L,t) - EI\omega'''(L,t) + Mg\cos\theta(t) = 0, \tag{6}$$

$$\omega(0,t) = \omega'(0,t) = \omega''(L,t) = 0. \tag{7}$$

Equations (4) and (5) represent the governing equations of motion, while (6) and (7) define the boundary conditions.

It can be observed that due to the gravity term $Mg\cos\theta$, the boundary condition (6) is non-homogeneous. Therefore, to obtain an accurate semi-analytical solution, a model transformation is required. To accomplish this, we first substitute (5) into (6) at $x = L$, resulting in the following modified boundary condition:

$$\omega''''(L,t) + p\omega'''(L,t) = f, \tag{8}$$

where $p = \frac{\rho A}{M}$ and $f = \frac{\rho A g}{EI}\cos\theta$. To homogenize the boundary conditions (7) and (8), we apply the following transformation, which maps $\omega(x,t)$ to $z(x,t)$, as follows [14]:

$$z(x,t) = \omega(x,t) + \nu(x,t). \tag{9}$$

$\nu(x,t)$ is chosen such that $z(x,t)$ satisfies the following homogeneous boundary conditions:

$$z(0,t) = z'(0,t) = z''(L,t) = 0, \tag{10}$$

$$z''''(L,t) + pz'''(L,t) = 0. \tag{11}$$

As a result, for $\nu(x,t)$ we should have:

$$\nu(0,t) = \nu'(0,t) = \nu''(L,t) = 0, \tag{12}$$

$$\nu''''(L,t) + p\nu'''(L,t) = -f. \tag{13}$$

Solving (12) and (13) yields the following expression for $\nu(x)$:

$$\nu(x,t) = \left(-\frac{\Gamma_1}{p^4}e^{-px} + \frac{1}{6p}x^3 + \Gamma_2 x^2 - \frac{\Gamma_1}{p^3}x + \frac{\Gamma_1}{p^4}\right) f \tag{14}$$

where Γ_1 and Γ_2 are defined as:

$$\Gamma_1 = \frac{2p^3L^3}{(3p^2L^2 - 6)e^{-pL} + 6 - 6pL}, \tag{15}$$

$$\Gamma_2 = \frac{\Gamma_1}{2p^2}e^{-pL} - \frac{1}{2p}. \tag{16}$$

Now, with $\nu(x,t)$ obtained from (14), the homogenized parameter $z(x,t)$ can be solved using the Assumed Mode Method (AMM).

III. Proposed Methodology

In the previous chapter, it was shown that the flexible manipulator is controlled using a single input torque, which is responsible for adjusting the angular position (θ) of the system. Therefore, due to the flexibility of the robot, the system has infinite degrees of freedom and is considered under-actuated. This makes it challenging to control all degrees of freedom using only one control input. For example, while the controller can effectively control the angular position (θ), it may still result in undesired vibrations (ω) during sharp motions. In fact, both the controller and the motion planner play critical roles in this system. To effectively suppress vibrations and reach the target as efficiently as possible, it is essential to derive an optimal trajectory between each angle.

In this study, we propose a novel approach that combines a motion planner using DRL with a nonlinear PDE controller. In this high-level-low-level framework, the high-level motion planner first generates the desired angular position (θ_d) to reach the target angle (θ_T), in a manner that minimizes vibrations. The low-level controller, consisting of the PDE nonlinear controller, then ensures the tracking of this desired angular position by producing the required input torque, as shown in Fig. 2.

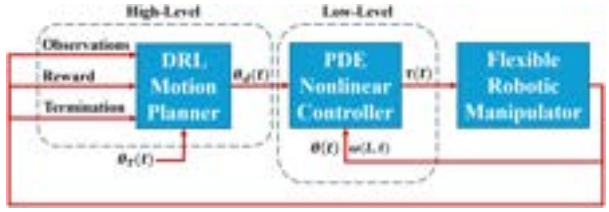

Fig. 2. Schematic View of the Proposed High-Level Motion Planner and the Low-Level Controller.

A. Nonlinear PDE Controller

As observed in the previous section, the input torque τ is applied to the system to control the angular position of the manipulator. In this section, a nonlinear PDE controller is proposed to control the flexible manipulator, which guarantees the stability of the system. The primary motivation for using a PDE controller lies in its model-based nature, which is particularly beneficial for systems with distributed parameters, such as flexible links. Since the dynamics of flexible structures are inherently described by partial differential equations, incorporating PDEs into the controller design allows for a more accurate representation of the system behavior, as follows:

$$\tau(t) = K_p e(t) + K_d\dot{e}(t) - H(t), \tag{17}$$

$$H(t) = EI\omega''(0,t) - \frac{1}{2}mgL\cos\theta(t) + Mg\omega(L,t)\sin\theta(t) - I_m\ddot{\theta}_d(t), \tag{18}$$

where $e(t) = \theta_d(t) - \theta(t)$ represents the tracking error, and K_p, K_d, and α are positive parameters. Consider the following Candidate Lyapunov Function (CLF), defined as:

$$V = \frac{1}{2}K_p e^2 + \frac{1}{2}I_m\dot{e}^2 + \alpha I_m e\dot{e}. \tag{19}$$

In this work, the Lyapunov function includes a coupling term ($\alpha I_m e\dot{e}$) to enable designing a controller with provable exponential stability. This term allows the derivative to satisfy $\dot{V} < -\lambda V$, ensuring exponential error decay.

Lemma 1: By selecting appropriate parameters for α, I_m, and K_p with the conditions $\alpha I_m < K_p$ and $\alpha < 1$, the proposed CLF in (19) is a positive function.

Proof: We begin by considering the following inequality:

$$(e \pm \dot{e})^2 = e^2 + \dot{e}^2 \pm 2e\dot{e} \geq 0, \tag{20}$$

thus, if we multiply both sides of this inequality by the positive value αI_m, we obtain:

$$|\alpha I_m e\dot{e}| \leq \frac{1}{2}\alpha I_m(e^2 + \dot{e}^2), \tag{21}$$

so, if we select α and K_p such that $\alpha I_m < K_p$ and $\alpha < 1$, we can conclude:

$$|\alpha I_m e\dot{e}| \leq \frac{1}{2}\alpha I_m(e^2 + \dot{e}^2) \leq \frac{1}{2}K_p e^2 + \frac{1}{2}I_m\dot{e}^2, \tag{22}$$

therefore, considering the proposed CLF (19) and the inequality in (22), we conclude that $V \geq 0$.

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

Lemma 2: By assuming a condition for α, I_m and K_d as $\alpha I_m < K_d$, the time derivative of the proposed CLF is upper bounded.

Proof:

$$\begin{aligned}\dot{V} &= K_p e\dot{e} + I_m \dot{e}\ddot{e} + \alpha I_m(\dot{e}^2 + e\ddot{e}) \\ &= (\dot{e} + \alpha e) I_m \ddot{e} + K_p e\dot{e} + \alpha I_m \dot{e}^2 \\ &= (\dot{e} + \alpha e)[EI\omega''(0,t) - \frac{1}{2}mgL\cos\theta + Mg\omega(L)\sin\theta \\ &\quad + \tau(t) - I_m\ddot{\theta}_d] + K_p e\dot{e} + \alpha I_m \dot{e}^2, \end{aligned} \tag{23}$$

now, by substituting the control law (17) into (23), we have:

$$\begin{aligned}\dot{V} &= -\alpha K_p e^2 - K_d \dot{e}^2 - K_d \alpha e\dot{e} + \alpha I_m \dot{e}^2 \\ &= -\alpha K_p e^2 - (K_d - \alpha I_m)\dot{e}^2 - K_d \alpha e\dot{e}. \end{aligned} \tag{24}$$

As $\alpha I_m < K_d$, we can define the following three positive variables λ_1, λ_2, and λ_3:

$$\alpha K_p = \lambda_1 \left(\frac{1}{2}K_p\right), \tag{25}$$

$$K_d - \alpha I_m = \lambda_2 \left(\frac{1}{2}I_m\right), \tag{26}$$

$$\alpha K_d = \lambda_3(\alpha I_m). \tag{27}$$

Consequently, we can write (24) as follows:

$$\dot{V} = -\lambda_1 \left(\frac{1}{2}K_p e^2\right) - \lambda_2 \left(\frac{1}{2}I_m \dot{e}^2\right) - \lambda_3 \left(\alpha I_m e\dot{e}\right). \tag{28}$$

Now, let's define λ as it satisfies the following condition:

$$\lambda < \min\{\lambda_1, \lambda_2, \lambda_3\}, \tag{29}$$

Thus, by selecting the parameter λ that satisfies the condition (29), we conclude that:

$$\dot{V} < -\lambda \left(\frac{1}{2}K_p e^2 + \frac{1}{2}I_m \dot{e}^2 + \alpha I_m e\dot{e}\right) = -\lambda V. \tag{30}$$

Theorem 1: The control law (17) ensures that the system tracks the desired angle (θ_d) and guarantees the exponential stability of the closed-loop system.

Proof: As shown in Lemma 1 and Lemma 2, $V \geq 0$ and $\dot{V} < -\lambda V$, therefore, from (30), we have:

$$\frac{dV}{V} < -\lambda dt, \tag{31}$$

By integrating this inequality over $[0, t]$, we can obtain:

$$V(t) < e^{-\lambda t} V(0), \tag{32}$$

Therefore, according to the Lyapunov stability theorem, the closed-loop system will be exponentially stable, and as $t \to \infty$, the system will track the desired angular position, leading to $e \to 0$.

B. Deep Reinforcement Learning for Motion Planning

As explained previously, both the controller and the motion planner play a crucial role in vibration suppression of flexible manipulators. We propose a DRL agent that generates motion plans for a flexible robotic manipulator to minimize vibrations while achieving the target angle. The SAC algorithm, a model-free DRL approach, has demonstrated high effectiveness in motion planning tasks [24]. Unlike standard RL, which aims to maximize the expected cumulative reward $\sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t,a_t)]$, the SAC algorithm optimizes a more general objective that incorporates entropy regularization. This objective encourages exploration by promoting stochastic policies, formulated as:

$$J(\pi) = \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t,a_t) + \beta H(\pi(\cdot|s_t))], \tag{33}$$

where $H(\pi(\cdot|s_t))$ represents the entropy of the policy, and β is a temperature parameter that controls the trade-off between reward maximization and policy stochasticity. In this algorithm, we use a soft state value function $V_\psi(s_t)$, two soft Q-functions $Q_{\theta_i}(s_t,a_t)$, and a policy $\pi_\phi(a_t|s_t)$, all implemented as neural networks with parameters ψ, θ_i, and ϕ, respectively [25].

Building on established research, the state and action spaces, as well as the reward function, were carefully adapted to capture the unique characteristics of our problem, enabling stable learning and consistent convergence.

In this context, the state, s_t, is defined as follows:

$$s_t = \left\langle e_T(t), \theta(t), \dot{\theta}(t), \tau(t), \omega(L,t), \dot{\omega}(L,t) \right\rangle, \tag{34}$$

where $e_T(t) = \theta_T(t) - \theta(t)$ represents the difference between the target joint angle θ_T and the current joint angle θ, and τ is the input torque. $\dot{\theta}$ and $\dot{\omega}$ denote the time derivatives of θ and ω, respectively, in time step t.

Joint velocities were used as the action space, a_t, as they are well-suited for motion planning tasks and have demonstrated superior reliability in sim-to-real transfer scenarios [28]. The continuous action vector is defined as:

$$a_t \sim \pi_\phi(a_t|s_t), \tag{35}$$

where $a_t \in [-1, 1]$ represents normalized joint velocity commands. These actions are then scaled by the maximum allowable joint velocity, $\dot{\theta}_{\max}$, to produce the final desired joint velocities:

$$\dot{\theta}_d = \dot{\theta}_{\max} \cdot a_t, \tag{36}$$

resulting in a final velocity range of $[-\dot{\theta}_{\max}, \dot{\theta}_{\max}]$. The value of $\dot{\theta}_{\max}$ is manually specified according to the physical characteristics and limitations of the manipulator. The desired angle $\theta_d(t)$ is then obtained by integrating the desired angular velocity:

$$\theta_d = \int_0^t \dot{\theta}_d dt. \tag{37}$$

The reward function is defined in (38) and consists of five terms, each designed to address specific challenges posed by the flexible-link manipulator.

$$R_t = W_e|e_T(t)| + W_{\dot{\theta}}|\dot{\theta}(t)| + W_{\dot{\omega}}|\dot{\omega}(L,t)| + R_{reach} + R_{failure}. \tag{38}$$

While error deviation alone is typically sufficient for standard reaching tasks, the additional terms were introduced to handle the unique dynamics of flexible manipulators. The first term, $W_e|e_T(t)|$, penalizes the deviation between the target and the current joint angle, which is common in reaching tasks. The second term, $W_{\dot{\theta}}|\dot{\theta}(t)|$, penalizes excessive joint velocity, a

Algorithm 1: Motion Planning and Control for Flexible Manipulators.

Input: Target trajectory $\theta_T(t)$, initial states
Output: Input torque $\tau(t)$ applied to the flexible manipulator

Initialize: SAC policy $\pi(s_t)$, PDE controller, initial conditions
for each time step t **do**
  Motion Planning (High-Level):
  DRL State: $s_t \leftarrow \langle e_T(t), \theta(t), \dot{\theta}(t), \tau(t), \omega(L,t), \dot{\omega}(L,t) \rangle$
  Generate desired velocity: $\dot{\theta}_d(t) \leftarrow \pi(s_t)$
  Calculate the desired position: $\theta_d(t) \leftarrow \int_0^t \dot{\theta}_d dt$
  Control (Low-Level):
  Compute input torque (from PDE controller in equation (17)): $\tau(t) \leftarrow f_{\text{PDE}}(\theta_d(t), \theta(t), \omega(L,t))$
  Apply torque $\tau(t)$ to the manipulator
end for

TABLE I
PARAMETERS OF THE FLEXIBLE MANIPULATOR AND THEIR VALUES

Symbol	Definition	Value (Unit)
L	Length of the Flexible Link	4.5 (m)
ρ	Density of the Flexible Link	7850 (Kg/m^3)
A	Cross Area of the Flexible Link	6.84×10^{-4} (m^2)
E	Young's Modulus of the Flexible Link	200 (GPa)
I	Moment of Inertia of the Flexible Link	3.71×10^{-7} (m^4)
M	Payload Mass	20 (Kg)

critical factor for flexible manipulators as high velocities can lead to instability. The third term, $W_{\dot{\omega}}|\dot{\omega}(L,t)|$, addresses tip vibration by penalizing excessive tip velocity, helping to reduce oscillations inherent in flexible manipulators. Additionally, a positive reward, R_{reach}, is given when the agent successfully reaches the target under desired conditions, while $R_{failure}$ imposes a penalty if the manipulator exceeds its operational limits. These terms were manually tuned to ensure convergence and a balanced exploration-exploitation trade-off, with their combination tailored to the specific demands of our flexible manipulator task.

As shown in Algorithm 1, the proposed method combines high-level model-free motion planning using SAC with low-level model-based controller to achieve accurate motion tracking for flexible manipulators while suppressing the vibration.

IV. NUMERICAL SIMULATION

Now, to evaluate the performance of the proposed high-level motion planner and low-level controller, a comparative analysis is conducted between traditional methods and the proposed method. The flexible manipulator is simulated in MATLAB Simulink, considering three modes. The simulation results for the conventional PID controller, the proposed PDE controller, and the combined PDE controller with the DRL motion planner are presented. The parameters used in these simulations are based on the real robot and are detailed in Table I.

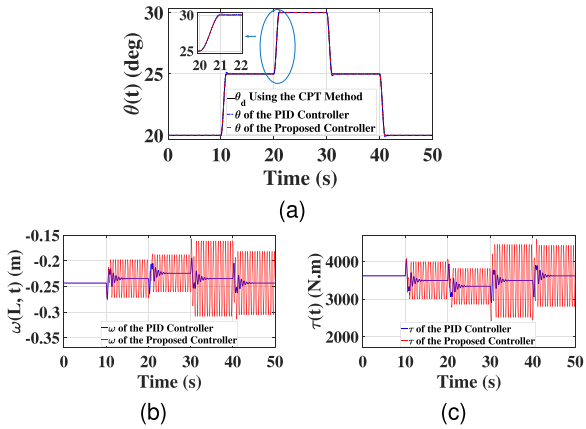

Fig. 3. Comparison of the PID Controller and the Proposed PDE Controller Using the CPT Method. (a) Angular Position. (b) Elastic Deviation of the Payload. (c) Control Input Torque.

A. Simulation Results Using the Conventional PID Controller

To track the desired angular position (θ_d), based on the cubic polynomial trajectory (CPT) method, we consider the conventional PID controller defined as:

$$\tau(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \dot{e}(t), \tag{39}$$

where, $e(t)$ is the tracking error, defined as $e(t) = \theta_d(t) - \theta(t)$, and K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively. By implementing this PID controller with the selection of $K_p = 150000$, $K_i = 100000$, and $K_d = 20000$, the simulation results are presented in blue in Fig. 3.

As observed in Fig. 3(a), although the result of the PID controller for tracking is acceptable, the vibrations at the payload, shown in Fig. 3(b), are significant, especially when the flexible manipulator starts to move between angles. Additionally, large jumps in the required torque are observed in in Fig. 3(c), which could potentially harm the actuator in a real robot.

B. Simulation Results Using the Proposed PDE Controller

Now, by applying the proposed nonlinear PDE controller, as presented in (17), with $K_p = 12000$ and $K_d = 15000$, the system is simulated as in the previous section. These simulation results are also depicted in red in Fig. 3, where the system is expected to track the same desired angular position (θ_d) as the previous section, generated using the CPT method.

As observed in Fig. 3(a), the tracking performance was significantly improved compared to the PID controller. Furthermore, it is important to note that while the conventional PID controller cannot guarantee the stability of the closed-loop system, the proposed controller ensures stability.

However, as shown in Fig. 3(b), the proposed controller did not effectively suppress the vibrations. This can be attributed to two main reasons: First, the PDE controller focuses solely on accurately controlling the angular position (θ) without addressing the suppression of payload vibrations. Second, the system is under-actuated, with only one control input; since there is no control input applied to the end effector, it cannot control the payload vibrations ($\omega(L,t)$). Thus, in this under-actuated

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

TABLE II
PARAMETERS OF THE DRL AGENT AND THEIR VALUES

Parameters	Value	Parameters	Value
Batch size	128	Initial random steps	500
Experience buffer length	10^6	Training episodes	5000
Discount factor	0.99	Max time steps/episode	300
Time step	0.1	W_e	-5×10^{-3}
Learning rate	10^{-4}	$W_{\dot\theta}$	-10^{-3}
Target smoothing factor	0.001	$W_{\dot\omega}$	-3×10^{-1}

system, utilizing a motion planner for vibration suppression becomes essential.

Furthermore, the proposed nonlinear PDE controller demonstrates better performance in tracking the desired angular position (θ_d) compared to the conventional PID controller. However, both the conventional PID controller and the proposed nonlinear PDE controller exhibited poor performance in vibration suppression. As a result, this letter proposes a DRL motion planner designed to generate the optimal path with minimal vibrations. The results using the DRL motion planner are presented in the following section.

C. Simulation Results Using the Proposed PDE Controller Combined With the DRL Motion Planner

As observed in the previous subsection, both control and motion planning are crucial for vibration suppression in flexible manipulators. As explained in the previous chapter, the SAC algorithm is proposed for motion planning, generating the desired angular position (θ_d).

The DRL agent, as described in Table II, operates at a frequency of 10 Hz, processing model states every 0.1 seconds and generating corresponding actions in the form of desired velocities for the low-level controller. These actions are initially constrained within the range $[-1, 1]$, and then scaled to fit within $[-\dot\theta_{\max}, \dot\theta_{\max}]$, where $\dot\theta_{\max} = 5$ deg/s. The low-level controller, on the other hand, operates at a significantly higher frequency of 1 kHz. Additionally, the reaching reward and the failure penalty, presented in (38), are defined as follows:

$$R_{reach} = +200, \quad \text{if} \quad \begin{cases} |e_T(t)| < 0.1, \\ |\dot\theta(t)| < 0.1, \\ |\dot\omega(L,t)| < 0.1 \end{cases}, \tag{40}$$

$$R_{failure} = -200, \quad \text{if} \quad \begin{cases} \theta(t) < 0, \\ \text{or} \\ \theta(t) > 90 \end{cases}. \tag{41}$$

To promote generalization in the DRL-based motion planner, a random target angle $\theta_T(t)$ and initial angle $\theta(0)$ are selected within the operational range of $[0^\circ, 90^\circ]$ at the start of each episode. Furthermore, to enhance robustness against system variations and bridge the sim-to-real gap, we adopt a domain randomization approach. In this method, key physical parameters of the environment—namely $[L, m, M, I_m, \rho A, EI]$—are perturbed by a random variation of $\pm10\%$ around their nominal values listed in Table I. This exposes the agent to a diverse set of scenarios during training, enabling it to learn policies resilient to uncertainty. Fig. 4 illustrates the training performance in terms of episode rewards, based on an implementation in MATLAB and Simulink. As shown, the agent's reward converges over time, indicating acceptable learning performance and stable policy development.

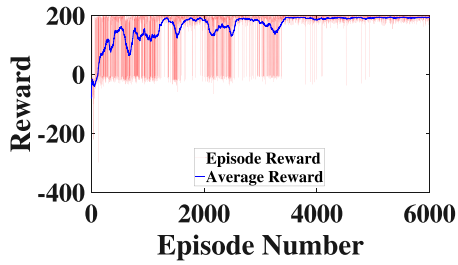

Fig. 4. Rewards in the Training Process.

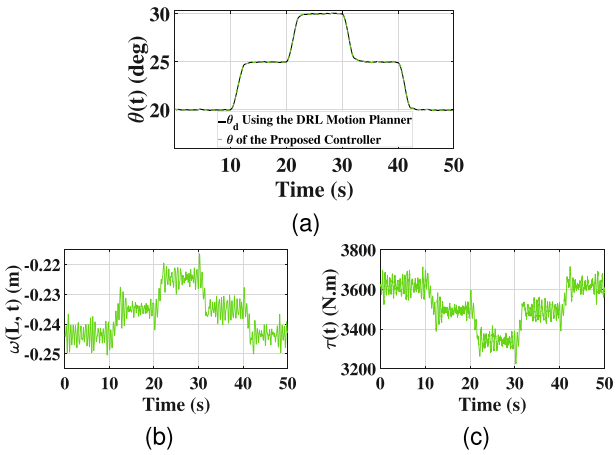

Fig. 5. Simulation Results of the Proposed PDE Controller Using the Proposed DRL Motion Planner. (a) Angular Position. (b) Elastic Deviation of the Payload. (c) Control Input Torque.

The simulation results using the proposed PDE controller and the DRL motion planner are illustrated in Fig. 5. As seen in Fig. 5(a), the proposed controller successfully tracks the desired angular position generated by the proposed motion planner. Furthermore, as shown in Fig. 5(b), the proposed DRL motion planner effectively suppresses the vibrations.

To evaluate vibration suppression at the flexible link tip, we compared the RMSE of $\dot\omega(L,t)$. Our PDE controller with DRL motion planner achieved the lowest RMSE of 1.2906×10^{-4} m/s, compared to 1.6×10^{-3} m/s for PDE controller with CPT method and 2.0807×10^{-4} m/s for PID controller with CPT method, confirming its superior vibration mitigation.

V. EXPERIMENTAL RESULTS

In the previous chapter, it was observed in simulation that the proposed PDE controller combined with the proposed DRL motion planner could effectively track the desired angle while suppressing the endpoint vibrations.

Now, in this chapter, the performance of the proposed controller and motion planner is evaluated on the real flexible robotic manipulator. Therefore, the performance of the proposed PDE

Fig. 6. Experimental Platform of the Flexible Manipulator.

controller without and with the proposed DRL motion planner are evaluated in the following sections.

A. Experimental Platform

The experimental setup of the flexible robotic manipulator is shown in Fig. 6.

In general, the robot consists of several components, each serving a specific role, as outlined below:

1) Flexible Link: The flexible link is constructed from OPTIM 700 MH Plus, a high-strength structural steel. With a length of 4.5 m and a cross-sectional area of 60 mm $\times$ 60 mm (thickness of 3mm), the beam's mass is approximately 22.5 kg, while it carries an external load at the end of its length. The steel has a yield strength of 700 MPa, with a tensile strength ranging from 750 to 950 MPa.

2) Hydraulic Valve and Cylinder: For controlling the link's angular position, a hydraulic-actuated system consisting of a $\oslash$ 35/25–300 mm single-acting hydraulic cylinder and a Bosch Rexroth 4WRPEH servo valve is used.

3) Measurement of the Actual Force: For measuring the amplitude of the force applied by the hydraulic cylinder (F), an HBM U9B load cell is utilized. As the hydraulic cylinder is attached to the basement, 29cm from the revolute joint ($r = 29$ cm), the torque of $T = r \times F$ is applied to the flexible link that controls the angular position of the robot.

4) Measurement of the Angular Position: To measure the angular position of the flexible link (θ), a SICK DS460 Encoder is used.

5) Measurement of the Position of the Payload: For deriving the position of the payload ($y(L,t)$), this project uses the Leica Absolute Laser Tracker AT960-LR, a cutting-edge portable measurement device designed for high-precision tracking of objects in three-dimensional space.

By knowing the position of the endpoint, using the Laser Tracker, in conjunction with the angular position, using the Encoder, the elastic deviation of the endpoint ($\omega(L,t) = y(L,t) - L\theta(t)$) is calculated.

B. Experimental Results Using the Proposed PDE Controller

In this section, the proposed PDE controller, given by (17), is implemented on the flexible robotic manipulator shown in Fig. 6. It should be noted that, in this section, the DRL motion planner is not used, and the desired angle (θ_d) is generated using the CPT method. The results are illustrated in Fig. 7, represented in red.

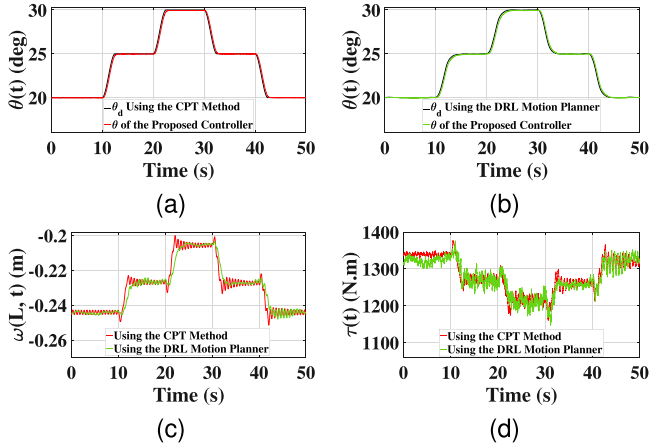

Fig. 7. Experimental Results. (a) Angular Position Using the CPT Method. (b) Angular Position Using the DRL Motion Planner. (c) Comparison of the Elastic Deviation of the Payload. (d) Comparison of the Control Input Torque.

TABLE III
RSME OF THE ANGULAR POSITION, USING THE PROPOSED CONTROLLER, WITH DIFFERENT UNCERTAINTIES

Uncertainties	Root Mean Squared Error (Unit)
-40% Uncertainty	$0.2934(deg)$
-20% Uncertainty	$0.2257(deg)$
Nominal Parameters	$0.1875(deg)$
$+20\%$ Uncertainty	$0.2041(deg)$
$+40\%$ Uncertainty	$0.2712(deg)$

As seen in Fig. 7(a), the proposed controller effectively tracks the desired angle. However, the endpoint vibrations, as shown in Fig. 7(c), are not acceptable, particularly during lowering.

To evaluate the robustness of the proposed controller, parametric uncertainty should be considered in this research. The main parameters that are used in the proposed controller (17) are [E, I, m, L, M]. Therefore, the Root Mean Squared Error (RMSE) of the angular position with uncertainties ranging from -40% to $+40\%$, compared to the nominal parameters presented in Table I, are presented in Table III.

It can be seen that although the controller with nominal parameters has the best performance in tracking the desired angle, with an RMSE of $0.1875(deg)$, the controllers under uncertainties still exhibit acceptable performance, with a maximum RMSE of $0.2934(deg)$.

C. Experimental Results Using the Proposed PDE Controller Combined With the Proposed DRL Motion Planner

In this section, the proposed PDE controller is evaluated when the desired angle (θ_d) is generated using the proposed DRL motion planner. The proposed DRL motion planner is designed to suppress the vibrations and generate the optimal path between each pair of angles. The results are depicted in Fig. 7, using green coloring.

As shown in Fig. 7(b), the DRL motion planner produces smooth transitions between angles. Compared to CPT,

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

it achieves superior vibration suppression (Fig. 7(c)), keeping amplitudes below 3 mm when raising and 6 mm when lowering. Notably, unlike model-based methods such as model predictive control which are sensitive to modeling uncertainties, the model-free DRL approach adapts through interaction, enabling reliable trajectory generation despite uncertainties. Moreover, DRL offers superior real-time performance over offline algorithms like CPT, making it well-suited for flexible systems.

VI. Conclusion

This letter presented a novel approach for flexible robotic manipulators, integrating a nonlinear PDE controller with a DRL motion planner. The primary objective was to achieve precise trajectory tracking while suppressing endpoint vibrations, which is an inherent challenge in flexible manipulators.

Numerical simulations demonstrated that the proposed PDE controller outperforms the PID controller in terms of tracking accuracy and stability; however, it was insufficient in mitigating vibrations due to the system's underactuated nature. To address this limitation, a DRL motion planner was introduced, leveraging the SAC algorithm with domain randomization, to generate an optimized trajectory that minimizes vibrations.

Experimental validations confirmed the effectiveness of the proposed approach, that it successfully reduced vibration amplitudes to within 3 mm (raising) and 6 mm (lowering)—a substantial improvement for a heavy duty flexible manipulator.

While the proposed method significantly enhanced performance, future research may explore:

- Enhancing the DRL agent to adapt to varying payloads and dynamic uncertainties.
- Extending the DRL framework to optimize not only vibration suppression but also energy efficiency and speed.
- Implementing the approach on multi-link flexible manipulators to validate its scalability in more complex systems.
- Investigating fine-tuning of the DRL motion planner on the real system, following initial training in simulation, to further improve real-world adaptability.

References

[1] E. A. Alandoli and T. S. Lee, "A critical review of control techniques for flexible and rigid link manipulators," *Robotica*, vol. 38, no. 12, pp. 2239–2265, 2020.

[2] J. Chen et al., "A variable length, variable stiffness flexible instrument for transoral robotic surgery," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 3835–3842, Apr. 2022.

[3] Y. Hu, W. Li, L. Zhang, and G.-Z. Yang, "Designing, prototyping, and testing a flexible suturing robot for transanal endoscopic microsurgery," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1669–1675, Apr. 2019.

[4] M. Sayahkarajy, Z. Mohamed, and A. A. Mohd Faudzi, "Review of modelling and control of flexible-link manipulators," *Proc. Inst. Mech. Engineers, Part I: J. Syst. Control Eng.*, vol. 230, no. 8, pp. 861–873, 2016.

[5] M. Iskandar, C. van Ommeren, X. Wu, A. Albu-Schäffer, and A. Dietrich, "Model predictive control applied to different time-scale dynamics of flexible joint robots," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 672–679, Feb. 2023.

[6] L. Tang, M. Gouttefarde, H. Sun, L. Yin, and C. Zhou, "Dynamic modelling and vibration suppression of a single-link flexible manipulator with two cables," *Mechanism Mach. Theory*, vol. 162, 2021, Art. no. 104347.

[7] K. Li, H. Wang, X. Liang, and Y. Miao, "Visual servoing of flexible-link manipulators by considering vibration suppression without deformation measurements," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12454–12463, Nov. 2022.

[8] Z. Jing, Y. Ma, X. Wu, X. He, and Y. Sun, "Backstepping control for vibration suppression of 2-D Euler–Bernoulli beam based on nonlinear saturation compensato," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 53, no. 5, pp. 2562–2571, May 2023.

[9] W. He and S. S. Ge, "Vibration control of a flexible beam with output constraint," *IEEE Trans. Ind. Electron.*, vol. 62, no. 8, pp. 5023–5030, Aug. 2015.

[10] Z. Zhao, X. He, and C. K. Ahn, "Boundary disturbance observer-based control of a vibrating single-link flexible manipulator," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 51, no. 4, pp. 2382–2390, Apr. 2021.

[11] T. Jiang, J. Liu, and W. He, "Boundary control for a flexible manipulator based on infinite dimensional disturbance observer," *J. Sound Vib.*, vol. 348, pp. 1–14, 2015.

[12] A. H. Barjini, S. Yaqubi, S. M. Tahamipour-Z, and J. Mattila, "Deep learning-based deflection correction and end-point control of heavy-duty vertical single-link flexible manipulators," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2024, pp. 905–912.

[13] J. K. Viswanadhapalli, V. K. Elumalai, S. Shivram, S. Shah, and D. Mahajan, "Deep reinforcement learning with reward shaping for tracking control and vibration suppression of flexible link manipulator," *Appl. Soft Comput.*, vol. 152, 2024, Art. no. 110756.

[14] S. Yaqubi, S. M. Tahamipour-Z, and J. Mattila, "Semi-analytical design of PDE endpoint controller for flexible manipulator with non-homogenous boundary conditions," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 4, pp. 7257–7274, Oct. 2024.

[15] H. Sayyaadi and M. Hejrati, "Boundary force control of two one-link flexible manipulator to accomplish safe grasping task," in *Proc. 29th Annu Int. Conf. Iran. Assoc. Mech. Eng.; Proc. 8th Int. Conf. Therm. Power Plants Ind.*, 2021. [Online]. Available: https://civilica.com/doc/1238334

[16] H. Gao, W. He, C. Zhou, and C. Sun, "Neural network control of a two-link flexible robotic manipulator using assumed mode method," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 755–765, Feb. 2019.

[17] W. He, H. Gao, C. Zhou, C. Yang, and Z. Li, "Reinforcement learning control of a flexible two-link manipulator: An experimental investigation," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 51, no. 12, pp. 7326–7336, Dec. 2021.

[18] A. H. Barjini, M. Khoshnazar, and H. Moradi, "Design of a sliding mode controller for suppressing coupled axial & torsional vibrations in horizontal drill strings using extended Kalman filter," *J. Sound Vib.*, vol. 586, 2024, Art. no. 118477.

[19] M. G. Tamizi, M. Yaghoubi, and H. Najjaran, "A review of recent trend in motion planning of industrial robots," *Int. J. Intell. Robot. Appl.*, vol. 7, no. 2, pp. 253–274, 2023.

[20] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *J. Mach. Learn. Res.*, vol. 22, no. 30, pp. 1–82, 2021.

[21] R. S. Sutton et al., *Reinforcement Learning: An Introduction*, vol. 1. Cambridge, MA, USA: MIT Press, 1998.

[22] A. del Real et al., "A review of deep reinforcement learning approaches for smart manufacturing in industry 4.0 and 5.0 framework," *Appl. Sci.*, vol. 12, no. 23, 2022, Art. no. 12377.

[23] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 3389–3396.

[24] E. Prianto, J.-H. Park, J.-H. Bae, and J.-S. Kim, "Deep reinforcement learning-based path planning for multi-arm manipulators with periodically moving obstacles," *Appl. Sci.*, vol. 11, no. 6, 2021, Art. no. 2587.

[25] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn.*, Stockholm, Sweden, Jul. 2018, pp. 1861–1870.

[26] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 39, pp. 1–40, 2016.

[27] M. H. Shahna, S. A. A. Kolagar, and J. Mattila, "Integrating deeprl with robust low-level control in robotic manipulators for non-repetitive reaching tasks," in *Proc. IEEE Int. Conf. Mechatron. Automat.*, 2024, pp. 329–336.

[28] E. Aljalbout, F. Frank, M. Karl, and P. van der Smagt, "On the role of the action space in robot manipulation learning and sim-to-real transfer," *IEEE Robot. Autom. Lett.*, vol. 9, no. 6, pp. 5895–5902, Jun. 2024.