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

Modeling and Reinforcement Learning-Based Control of Simultaneous Positive and Negative Pressure Generation in Pneumatic Systems

Sang Hyeon Park
, Myeongyun Doh, Chanyong Park, Anh Tuan Luong, *Member, IEEE*, Hyouk Ryeol Choi, *Fellow, IEEE*, Ja Choon Koo, *Member, IEEE*, Hugo Rodrigue, *Senior Member, IEEE*, and Hyungpil Moon, *Member, IEEE*

***Abstract*—In soft robotics, actuators using both positive and negative pressures are notable for their high payload-to-weight ratios and wide operating ranges, but they require separate power sources. A single-pump system generating dual pressures presents a promising solution, though addressing pressure fluctuations due to coupled dynamics remains a challenge. In this work, we propose a reinforcement learning (RL)-based controller capable of tracking both pressures over a wide range. To facilitate RL training, we built a simulator that models not only airflow dynamics but also the pump's kinematics and the electromagnetic behavior of pneumatic components. Our controller employs Model-Predicted Observation (MPObs) to predict future input effects and mitigate nonlinearities, and uses a Conditioning for Action Policy Smoothness (CAPS)-based action smoothing to reduce abrupt input changes. Experimental results show that the proposed RL controller achieves root-mean-square errors (RMSEs) of 0.6935 kPa (positive) and 0.2646 kPa (negative), outperforming the Disturbance Observer (DOB)-based approach. Ablation studies confirm the synergistic effect of MPObs and CAPS, underscoring their importance in control. Furthermore, robustness tests with external loads from 0 to 20 kg demonstrate a maximum RMSE of 0.7906 kPa (positive) and 0.1186 kPa (negative), indicating strong robustness. This study verifies that our proposed RL-based controller overcomes the nonlinear challenges of pneumatic power sources and highlights its potential for future stand-alone systems in field applications.**

***Index Terms*—Reinforcement (RL) learning, modeling, control, and learning for soft robots, soft robot applications.**

I. Introduction

SOFT robots are being studied widely due to their inherent compliance and high payload-to-weight ratios, which make them attractive for diverse applications. In particular, pneumatic actuators—operated by air pressure—have been researched intensively. Among these, actuators that simultaneously utilize both positive and negative pressures can generate higher force [1], [2] and a wider operating range [3] compared to systems using only a single type of pressure, making them more effective. However, most current dual-pressure actuators require separate pressure sources for generating positive and negative pressures independently. In laboratory settings, wall-mounted power sources or large-scale compressors or vacuum pumps are commonly used, however, their substantial size and weight render them impractical for field applications. Conversely, systems where mobility is critical—such as mobile robots [4] or wearable robots [5]—often depend on cartridge-based power sources or pneumatic lines that draw air from wall-mounted sources, which can hinder performance.

To effectively deploy soft robots in field applications, our long-term objective is to develop a compact, stand-alone power source capable of simultaneously generating both positive and negative pressure. However, limited flow rates and coupled dynamics between the positive and negative channels present challenges for pressure generation and regulation, necessitating advanced control methods. Existing methods include rule-based/PD control for variable stiffness [6], sliding mode control with virtual elastic elements for PAM-based exoskeletons [7], and neural network–assisted feedback control [8]. Despite demonstrating effective control, these methods are constrained by reduced responsiveness to nonlinearity, manual tuning demands, and limited robustness, which ultimately hinder their applicability across a wide operating range. Additionally, PID [9] and on/off control [10] fail to provide effective pressure control due to the nonlinearities of the pneumatic system. Furthermore, in research where a single pump simultaneously generates both positive and negative pressures, the coupled dynamics are treated as disturbances and mitigated using a disturbance observer (DOB) based on a linearized model [11]. However, this approach only operates effectively at a specific linearization point, resulting in a limited control range.

To address the challenge of regulating both positive and negative pressure over a wide range in compact pneumatic systems, this study proposes a Reinforcement Learning (RL)-based controller. RL–based approach is well-suited for this task, as it can directly learn optimal control policies over a

Received 10 December 2024; accepted 30 April 2025. Date of publication 6 June 2025; date of current version 16 June 2025. This letter was recommended for publication by Associate Editor E. Falotico and Editor C. Laschi upon evaluation of the reviewers' comments. This work was supported by the National Research Foundation of Korea (NRF) through Korea government (MSIT) under Grant RS-2023-00207772. *(Sang Hyeon Park and Myeongyun Doh contributed equally to this work.) (Corresponding author: Hyungpil Moon.)*

Sang Hyeon Park, Myeongyun Doh, Chanyong Park, and Anh Tuan Luong are with the Faculty of Mechanical Engineering, Sungkyunkwan University, Suwon 2066, South Korea (e-mail: pshjohn98@g.skku.edu; ehaud-dbs@g.skku.edu; pcy23107@g.skku.edu; luongtuan@skku.edu).

Hyouk Ryeol Choi, Ja Choon Koo, Hugo Rodrigue, and Hyungpil Moon are with the Faculty of Mechanical Engineering, Sungkyunkwan University, Suwon 2066, South Korea, and also with the Faculty of Intelligent Robotics, Sungkyunkwan University, Suwon 2066, South Korea (e-mail: choihyoukryeol@gmail.com; jckoo@skku.edu; rodrigue@skku.edu; hyungpil@skku.edu).

Digital Object Identifier 10.1109/LRA.2025.3577422

wide range of pressures without requiring linearization, thereby offering robust performance under diverse operating conditions. Specifically, RL leverages experimental data to derive control strategies without needing a perfect model of the system's complex characteristics. These characteristics make RL a compelling choice for pressure control in pneumatic systems with strong nonlinearities [12], [13], [14], [15], [16].

To train the RL algorithm effectively and mitigate risks—such as signal divergence, excessive energy consumption, and hardware wear—that may arise when training directly on the real system [17], we propose a multiphysics-based pneumatic system simulator for RL training. Unlike simulators such as FluidSIM or Automation Studio, which focus solely on fluid dynamics and thermodynamics, our model also incorporates pump and valve kinematics along with their electromagnetic behavior. This allows the simulator to closely mirror key features—like pump suction/exhaust and simultaneous positive and negative pressure generation—while supporting rapid training and large-scale data collection. However, using a simulator inevitably introduces a domain gap, which can cause issues such as oscillations in control inputs [17]. To address this, we propose the Model-Predicted Observation (MPObs) technique, which forecasts future input effects to mitigate nonlinearities, including coupled dynamics, and an action-smoothing method based on Conditioning for Action Policy Smoothness (CAPS) [17] to suppress abrupt signal changes and ensure simulation to real (sim-to-real) transfer.

To validate the accuracy of the proposed simulator and the performance of the RL controller, we conducted various experiments. Initially, we evaluated the simulator's accuracy and compared the performance of the RL controller with that of a conventional DOB-based controller. Ablation tests were then performed to quantitatively assess the contributions of the MPObs and CAPS techniques to the performance improvement of the RL-based controller. Finally, we evaluated the controller's robustness against external load variations without additional training, thereby demonstrating its stable operation under various disturbance conditions. The RL-based control method and simulator developed in this study serve as key foundational technologies for the practical implementation of stand-alone pneumatic power sources, and they enhance the potential for field applications in mobile and soft robots.

In this article, major contributions are:

- Construction of an interactive simulator for reinforcement learning (RL) that accurately models pneumatic systems capable of generating simultaneous positive and negative pressures by incorporating not only airflow dynamics but also the kinematic characteristics of the pump and the electromagnetic behavior of pneumatic components.
- Design an RL-based controller to generate target positive and negative pressures. It leverages Model-Predicted Observation (MPObs) to mitigate nonlinear effects by forecasting future system responses, and employs Conditioning for Action Policy Smoothness (CAPS) to smooth control inputs, thereby reducing oscillations caused by domain gaps and ensuring stable operation.

The remainder of this letter is organized as follows: Section II details the modeling of the pneumatic components and the construction of the simulator for RL training. Section III describes the design and training of the RL controller. Section IV presents experimental validations, and Section V concludes the letter.

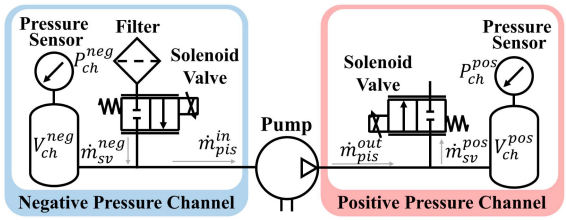

Fig. 1. Pneumatic system schematic.

II. Pneumatic System Simulator

The pneumatic system is depicted in Fig. 1. To simulate this pneumatic system, a custom simulator is developed because existing commercial pneumatic simulators cannot accurately model systems where a pump simultaneously performs both suction and exhaust processes. Our pneumatic system operates as follows: the central pneumatic pump draws air from the negative pressure channel (the blue region in Fig. 1) while expelling air into the positive pressure channel (the red region in Fig. 1). The air entering the pump is a combination of air drawn from the negative pressure chamber and the atmosphere through the negative pressure valve. Conversely, the air exiting the pump is divided between filling the positive pressure chamber and being released to the atmosphere through the positive pressure valve. The flow rates into and out of the chambers are controlled by adjusting the openings of proportional solenoid valves. These valves regulate the air exchange with the atmosphere, thereby modulating the internal pressures of the chambers. To model the unique dynamics of this system, a mathematical simulator encompassing not only air dynamics [18] but also kinematics [19] and electromagnetics [20] is implemented. This approach enables precise representation of simultaneous suction and exhaust processes, as well as their interaction with each pressure channel.

A. Pneumatic Pump

1) Description of Pneumatic Pump: The pneumatic piston pump is used to generate both positive and negative pressures, as illustrated in Fig. 2(a). The pump consists of two pistons connected to a motor rotor with a 180-degree phase difference. A slider-crank mechanism drives the pistons, causing their heads to move up and down. This reciprocating motion cyclically alters the internal volume of the pistons. When the internal volume decreases, the pressure rises, resulting in positive pressure as air is expelled through the outlet port. Conversely, when the internal volume increases, the internal pressure drops, allowing the pump to draw air through the inlet port, thereby generating negative pressure. This process is repeated continuously to maintain the desired pressure levels.

2) Modeling of Pneumatic Pump: The air within the system is modeled as an ideal gas, governed by the ideal gas law. The

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

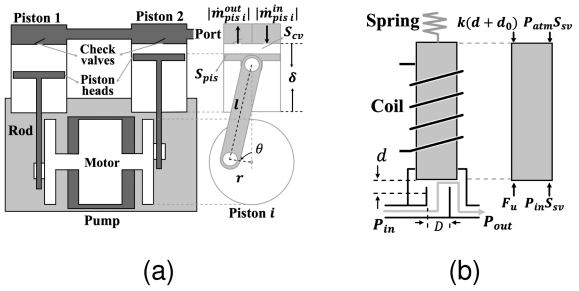

Fig. 2. Schematics of pneumatic elements. (a) Piston pump schematic. (b) Solenoid valve schematic.

pressure inside the piston P_{pis} and its time derivative $\dot{P}_{\text{pis}}$ can be expressed as (1).

$$P_{\text{pis}} = \frac{m_{\text{pis}} R T_{\text{pis}}}{V_{\text{pis}}}, \dot{P}_{\text{pis}} = -\frac{P_{\text{pis}} \dot{V}_{\text{pis}} - \dot{m}_{\text{pis}} R T_{\text{pis}}}{V_{\text{pis}}} \tag{1}$$

The piston volume V_{pis} is a function of the piston head position, which varies cyclically due to the slider-crank mechanism. Assuming a constant motor angular velocity ω, both the piston volume V_{pis} and its time derivative $\dot{V}_{\text{pis}}$ can be calculated as (2):

$$\begin{aligned} V_{\text{pis}} &= S_{\text{pis}} \left(\delta - r + l - r cos\theta - \sqrt{l^2 - r^2 \sin^2 \theta} \right) \\ \dot{V}_{\text{pis}} &= S_{\text{pis}} \omega \left(r \sin\theta + \frac{r^2 \sin\theta \cos\theta}{\sqrt{l^2 - r^2 \sin^2 \theta}} \right) \end{aligned} \tag{2}$$

Here, S_{pis} represents the equivalent area of the piston head, δ denotes the maximum piston volume variation, r is the crank length, θ is the crank angle, and ϕ is the piston head's rotational phase.

The air mass flow rate entering and exiting the piston is influenced by the dynamics of the check valve, as well as the inlet port pressure and the internal piston pressure. The check valve behavior is modeled using the orifice flow equations [18], [21], as shown in (3).

$$|\dot{m}_{\text{pis}}(P_{\text{in}}, P_{\text{out}})| = C_{\text{cv}} S_{\text{cv}} \frac{P_{\text{in}}}{\sqrt{RT}} \Phi(P_{\text{in}}, P_{\text{out}})$$

$$\Phi(P_{\text{in}}, P_{\text{out}}) = \begin{cases} \sqrt{\kappa \left(\frac{2}{\kappa+1} \right)^{\frac{\kappa+1}{\kappa-1}}}, & \text{if } P_r \le P_{cr} \\ \sqrt{\frac{2\kappa}{\kappa-1}} \sqrt{(P_r)^{\frac{2}{\kappa}} - (P_r)^{\frac{\kappa+1}{\kappa}}}, & \text{if } P_{cr} < P_r \le 1 \\ 0, & \text{if } P_r > 1 \end{cases} \tag{3}$$

Here, κ represents the specific heat ratio of air, P_r denotes the pressure ratio, which is the ratio of the outlet pressure P_{out} to the inlet pressure P_{in}, and P_{cr} refers to the critical pressure ratio that defines the boundary between choked and unchoked flow. In conclusion, by utilizing (1) through (3), the time derivative of the pressure inside the piston can be calculated. Since the discharge coefficient C_{cv} and the orifice area of the check valve S_{cv} cannot be measured directly, their product, $\widehat{C}_{\text{cv}} = C_{\text{cv}} S_{\text{cv}}$, was determined using the Nelder-Mead method [22].

B. Solenoid Valve

1) Description of Solenoid Valve: A 2-way proportional solenoid valve is employed to regulate the flow rate and a schematic is Fig. 2(b). This valve operates based on a nozzle-flapper mechanism. When a direct current is applied, the solenoid within the valve generates a magnetic field, causing the flapper to move. The movement of the flapper opens the flow path within the valve, allowing air to flow. Simultaneously, the current passing through the solenoid is modulated via pulse width modulation (PWM), enabling precise control of the valve opening and, consequently, the air flow rate.

2) Modeling of Solenoid Valve: The mass flow rate through the nozzle-flapper mechanism can be expressed as (4) [21]:

$$|\dot{m}_{\text{sv}}(P_{\text{in}}, P_{\text{out}}, u)| = C_{\text{sv}} D \pi d \frac{P_{\text{in}}}{\sqrt{RT}} \Phi(P_{\text{in}}, P_{\text{out}}) \tag{4}$$

Here, C_{sv} represents the solenoid valve's flow coefficient, d denotes the flapper opening length, D is the valve nozzle diameter, and $\Phi(P_{\text{in}}, P_{\text{out}})$ corresponds to the parameter defined in (3). The flapper position is determined by the forces exerted by the solenoid, the return spring, and the pressure differential between the valve ports. The term $C_{\text{sv}} \pi D d$, which represents the effective flow coefficient of the solenoid valve, can be modeled using (5).

$$\begin{aligned} C_{\text{sv}} D \pi d &= C_{\text{sv}} D \pi \max \left\{ \frac{1}{k} \left(F_u + \frac{\pi D^2}{4} (P_{\text{in}} - P_{\text{atm}}) - k d_0 \right), 0 \right\} \\ &= \max \left\{ \widehat{C}_u u + \widehat{C}_P \frac{\pi D^2}{4} (P_{\text{in}} - P_{\text{atm}}) - \widehat{C}_k, 0 \right\} \end{aligned} \tag{5}$$

Here, F_u the solenoid force, is assumed to be a linear function of the control input $u \in [0, 1]$. Additionally, k is the spring constant of return spring, and d_0 represents the flapper's neutral position, where it rests without any current input. By combining (4) and (5), the mass flow rate through the valve can be calculate. The parameters related to the input, pressure, and spring coefficient, namely $\widehat{C}_u$, $\widehat{C}_P$, and $\widehat{C}_k$, can be estimated using the linear least squares approach.

C. Chamber

Air tanks serve as chambers in the system. The mass flow rate entering the positive pressure channel chamber is calculated as the difference between the mass flow rate supplied by the pump and the mass flow rate discharged through the valve. This net mass flow directly influences the internal pressure of the chamber, which is crucial for maintaining precise pressure regulation. Conversely, for the negative pressure chammel chamber, the net mass flow rate is obtained by subtracting the mass flow rate entering through the valve from the mass flow rate extracted by the pump. By applying the ideal gas law in conjunction with these mass flow rates, the internal pressures of both chambers can be accurately calculated. The pressure dynamics of the positive pressure chamber are described by (6), while those of the negative pressure chamber are captured in (7):

$$\dot{P}_{\text{ch}}^{\text{pos}} = \frac{R T_{\text{ch}}}{V_{\text{ch}}^{\text{pos}}} \left(\sum_{i \in \{1,2\}} |\dot{m}_{\text{pis}\,i}^{\text{out}}| - |\dot{m}_{\text{sv}}^{\text{pos}}| \right) \tag{6}$$

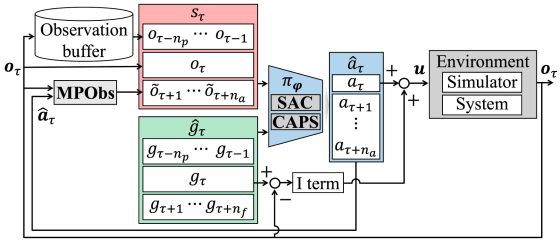

Fig. 3. Reinforcement learning based controller schematic.

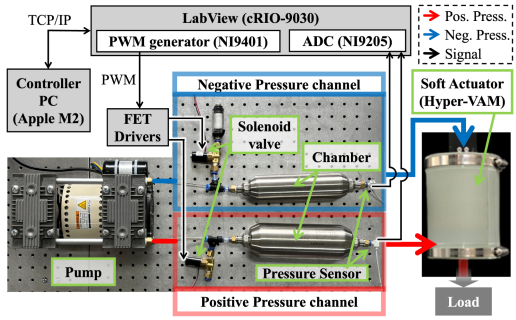

Fig. 4. System implementation.

$$\dot{P}_{\text{ch}}^{\text{neg}} = \frac{RT_{\text{ch}}}{V_{\text{ch}}^{\text{pos}}} \left(- \sum_{i \in \{1,2\}} |\dot{m}_{\text{pis}\,i}^{\text{in}}| + |\dot{m}_{\text{sv}}^{\text{neg}}| \right) \tag{7}$$

D. Simulator

In the preceding sections, the pump, solenoid valve, and chamber were individually modeled, and by integrating these models, a continuous nonlinear state-space model was derived. The system state of the pneumatic system, denoted by $\mathbf{x}(t) = [P_{\text{pis}\,1}, P_{\text{pis}\,2}, P_{\text{ch}}^{\text{pos}}, P_{\text{ch}}^{\text{neg}}]^T \in \mathbb{R}^4$, includes the pressures inside the piston and the positive and negative pressures in each chamber. The control input, $\mathbf{u}(t) = [u^{\text{pos}}, u^{\text{neg}}]^T$, consists of the signals applied to the solenoid valves of the positive and negative channels, where $u^{\text{pos}}, u^{\text{neg}} \in [0, 1]$. The integrated model is iteratively simulated using the 4th-order Runge-Kutta numerical integrator with timestep, t_s to replicate the behavior of the pneumatic system shown in Fig. 1. At each timestep, the control inputs modulate the valve states, thereby altering the airflow through the valves and pump, which subsequently updates the pressures within the piston and chambers.

This simulator enables realistic simulations by incorporating the kinematic characteristics of the pump and the electromagnetic behavior of the solenoid valves. By accurately modeling the coupled dynamics between positive and negative pressures through the kinematic motion of the pump's internal piston and valve actuation, it moves beyond traditional models that focus solely on fluid dynamics. Additionally, it uses state-space and quasi-static models, improving computational efficiency and enabling faster simulations. Its low computational load and interactivity enable diverse training scenarios, providing an ideal platform for training.

III. Reinforcement Learning Controller

In this section, we aim to design a policy network $\pi_\varphi(\cdot)$, parameterized by φ, that generates optimized control inputs through the training of a RL controller. This framework leverages two key methods: Model-Predicted Observation (MPObs) and action smoothing, which are uniquely integrated to address the challenges of pneumatic systems and enhance control performance.

The policy is optimized in a simulation environment developed in Section II and further fine-tuned through a sim-to-real transfer process to reduce the domain gap. Finally, the trained policy is deployed on the actual pneumatic system. The entire process is illustrated in the control schematic shown in Fig. 3 .

A. Control Architecture

1) Time Discretization and Key Definitions: The controller periodically updates the system state and generates the corresponding control input according to the control frequency f. Time t is converted into discrete time steps τ based on the frequency f, starting from an initial time t_0, with each step's time defined as $t_\tau = t_0 + \tau/f$. Additionally, we denote the history over a specific step interval as $[\tau_i : \tau_f]$, representing the set of data from t_{τ_i} to t_{τ_f}.

2) State Utilizing Model-Predicted Observation (MPObs): The state (s) is defined as $\mathbf{s}_\tau = \{\mathbf{o}_{[\tau-n_p:\tau]}, \widetilde{\mathbf{o}}_{[\tau+1:\tau+n_a]}\} \in \mathbb{R}^{2(n_p+n_a+1)}$, consisting of the observation $\mathbf{o} \in \mathbb{R}^2$ spanning n_p steps prior to and including the current step, and the MPObs $\widetilde{\mathbf{o}} \in \mathbb{R}^2$, which predicts the system behavior for the next n_a steps. Specifically, MPObs $\widetilde{\mathbf{o}}$ represents the future observation predicted using the pneumatic system model. It is obtained from the current observation $\mathbf{o}_\tau = \{P_{\text{ch}}^{\text{pos}}(t_\tau), P_{\text{ch}}^{\text{neg}}(t_\tau)\} \in \mathbb{R}^2$ and is expressed as $\widetilde{\mathbf{o}}_{\tau+p} = \{\widetilde{P}_{\text{ch}}^{\text{pos}}(t_{\tau+p}), \widetilde{P}_{\text{ch}}^{\text{neg}}(t_{\tau+p})\} \in \mathbb{R}^2$ for p steps forward. These predictions are derived from the initial state $\mathbf{x}(t_\tau)$, which is constructed from the actual system measurement $\mathbf{o}_\tau$. The future state $\widetilde{\mathbf{x}}(t_\tau + p/f)$ is then estimated using the pneumatic system model constructed in Section II, from which $\widetilde{\mathbf{o}}_{\tau+p}$ is determined—emulating Model Predictive Control (MPC)'s forward prediction [23]. MPObs enhances the RL-based controller's stability and performance by providing insights into the future effects of control inputs generated by the policy π_φ on the pneumatic system. By utilizing these predictions, the controller can effectively manage challenges such as the coupled dynamics of positive and negative pressures — even in the sim-to-real transfer procedure.

3) Goal and Action Trajectory: The goal $\mathbf{g}_\tau = \{P_{\text{ch}}^{\text{pos, ref}}(t_\tau), P_{\text{ch}}^{\text{neg, ref}}(t_\tau)\} \in \mathbb{R}^2$ represents the target pressures in the positive and negative pressure channels of the chambers at time step τ. The goal trajectory $\widehat{\mathbf{g}} = \mathbf{g}_{[\tau-n_p:\tau+n_f]} \in \mathbb{R}^{2(n_p+n_f+1)}$ is the set of goals spanning from n_p steps in the past up to n_f steps into the future (where $n_f \geq n_a$). On the other hand, the action $\mathbf{a}_\tau = \{u^{\text{pos}}(t_\tau), u^{\text{neg}}(t_\tau)\}$ represents the control input $u^{\text{pos}}, u^{\text{neg}} \in [0, 1]$ applied to the solenoid valves of the positive and negative pressure channels in the actual system at the discretized time t_τ. The policy π_φ generates the action trajectory $\widehat{\mathbf{a}}_\tau = \mathbf{a}_{[\tau:\tau+n_a]} \in \mathbb{R}^{2(n_a+1)}$, which includes the current action $\mathbf{a}_\tau$ as well as a sequence of n_a future actions necessary to compute the MPObs incorporated in the state.

4) Reward Function: The reward function $r(\mathbf{s}_\tau, \widehat{\mathbf{g}}_\tau) \in \mathbb{R}$ is defined as the weighted sum of the L1 norms between $\mathbf{g}_{\tau+j}$ and $\mathbf{o}_{\tau+j}$ for $j = 0, 1, \ldots, n_a$, where the weights $w_{r,j} < 0$ determine the relative importance of each time step.

B. Policy Optimization

1) Objective Function With Regularization for Action Smoothing: The objective function for policy optimization includes not only terms for RL but also terms for action smoothing based on a method called Conditioning for Action Policy Smoothness (CAPS). The objective function is expressed as (8):

$$J_{\pi_\varphi} = J^{\mathrm{RL}}_{\pi_\varphi} - \lambda_T J^T_{\pi_\varphi} - \lambda_S J^S_{\pi_\varphi} \tag{8}$$

The RL objective $J^{\mathrm{RL}}_{\pi_\varphi}$ serves as the primary objective function for policy optimization, guiding the learning process by evaluating and improving the policy throughout the entire training process. Various RL algorithms can define different policy objectives, and in this study, we employ the Soft Actor-Critic (SAC) algorithm [24]. SAC is employed due to its strong performance in continuous action spaces, stable learning, high data efficiency, and its capability to facilitate effective sim-to-real transfer [25], [26]. Compared to other algorithms, such as Proximal Policy Optimization (PPO) [27], Deep Deterministic Policy Gradient (DDPG) [28], and Twin Delayed Deep Deterministic Policy Gradient (TD3) [29], SAC offers greater reliability for robust policy learning.

For action smoothing, the objective function incorporates temporal and spatial smoothness regularization terms, $J^T_{\pi_\varphi}$ and $J^S_{\pi_\varphi}$, based on CAPS, as shown in (9) and (10):

$$J^T_{\pi_\varphi} = \|\pi_\varphi(\mathbf{s}_t, \mathbf{g}_t) - \pi_\varphi(\mathbf{s}_{t+1}, \mathbf{g}_t)\|_2 \tag{9}$$

$$J^S_{\pi_\varphi} = \|\pi_\varphi(\mathbf{s}_t, \mathbf{g}_t) - \pi_\varphi(\bar{\mathbf{s}}_t, \mathbf{g}_t)\|_2 \quad \text{where} \quad \bar{\mathbf{s}}_t \sim \mathcal{N}(\mathbf{s}_t, \sigma^2) \tag{10}$$

$J^T_{\pi_\varphi}$ encourages consistency between actions, while $J^S_{\pi_\varphi}$ improves the controller's robustness. The effects of $J^T_{\pi_\varphi}$ and $J^S_{\pi_\varphi}$ can be adjusted using their respective weights, λ_T and λ_S. The entire optimization process is carried out using the Adam optimizer [30].

2) Simulation to Real Transfer: The sim-to-real transfer process is essential for reducing the domain gap between the pneumatic model used to create both the simulator and the MPObs, and the actual pneumatic system. This process specifically aims to mitigate action fluctuations caused by the domain gap [17]. To achieve sim-to-real transfer, we fine-tune the policy through online adaptation using the objective function defined in (8). A key aspect of the sim-to-real transfer is adjusting the balance between exploration and action smoothing. In the initial training stages, active exploration is crucial for quickly narrowing the domain gap and identifying an optimal policy, while action smoothing becomes more important in later stages to enhance control performance. To balance exploration and action smoothing, we gradually adjust the influence of the temporal regularization term $J^T_{\pi_\varphi}$ in (8) and (9) over time. Since the contribution of $J^T_{\pi_\varphi}$ is controlled by the temporal regularization weight λ_T in (8), adjusting λ_T allows us to modulate the effect of temporal regularization dynamically throughout training. Specifically, we linearly increase λ_T in (8) according to the update rule $\lambda_T \leftarrow \lambda_T + \Delta_{\lambda_T}$ where $\lambda_T^{\min} \le \lambda_T \le \lambda_T^{\max}$, incrementing it by a fixed amount $\Delta_{\lambda_T} \ge 0$ at each training step. Finally, the control input $\mathbf{u}(t_\tau)$ is derived from the first value, $\mathbf{a}_\tau$, of the action trajectory $\widehat{\mathbf{a}}_\tau \sim \pi_\varphi(\cdot)$, incorporating an integral term and an anti-windup mechanism.

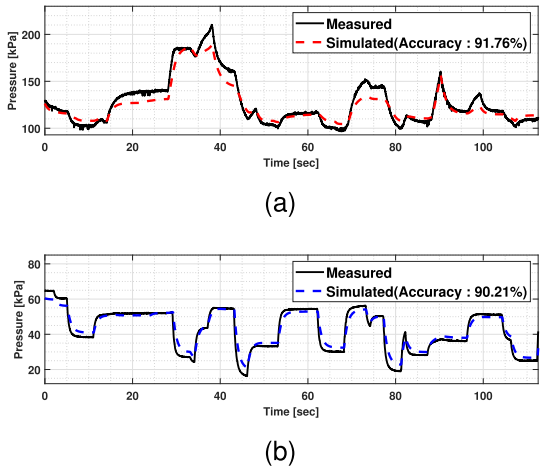

Fig. 5. Validation of simulator accuracy. (a) Positive pressure. (b) Negative pressure. The simulator accuracy is 91.76% for positive pressure and 90.21% for negative pressure.

IV. Results

A. Experiment Setup

1) Pneumatic System Setup: To evaluate the controllability of the proposed RL-based controller, an experimental setup of the pneumatic hardware system was configured as illustrated in Fig. 4. The system employed a double-head piston-type pneumatic pump, with a maximum flow rate of 42 LPM. PVQ31-5G-16 (SMC) was utilized as a proportional solenoid valve, offering an operational pressure range of 0–700 kPa. These valves were actuated via field-effect transistor (FET) drivers, which received pulse-width modulation (PWM) control signals. Pressure measurements were conducted using PSE540A-R06 (SMC) for positive pressure and PSE541A-R06 (SMC) for negative pressure. The positive pressure sensor covered a range of 0 to 1 MPa, while the negative pressure sensor spanned from 0 to -101 kPa, with both sensors providing analog voltage outputs between 1 and 5 V. Air chambers (CRVZS, FESTO) were employed, featuring a 0.75 L volume for the positive pressure chamber and a 0.4 L volume for the negative pressure chamber.

2) Actuation and Sensing Integration Setup: For sensor data acquisition and PWM signal generation, a cRIO-9030 (National Instruments) was utilized. The system employed an NI9205 module as an analog-to-digital converter for sensor data, and an NI9401 module for generating PWM signals. Communication between the RL controller on the PC and the cRIO was established via the TCP/IP protocol. The cRIO transmitted sensor data to the PC, where the control signal was computed and subsequently sent back to the cRIO for actuation.

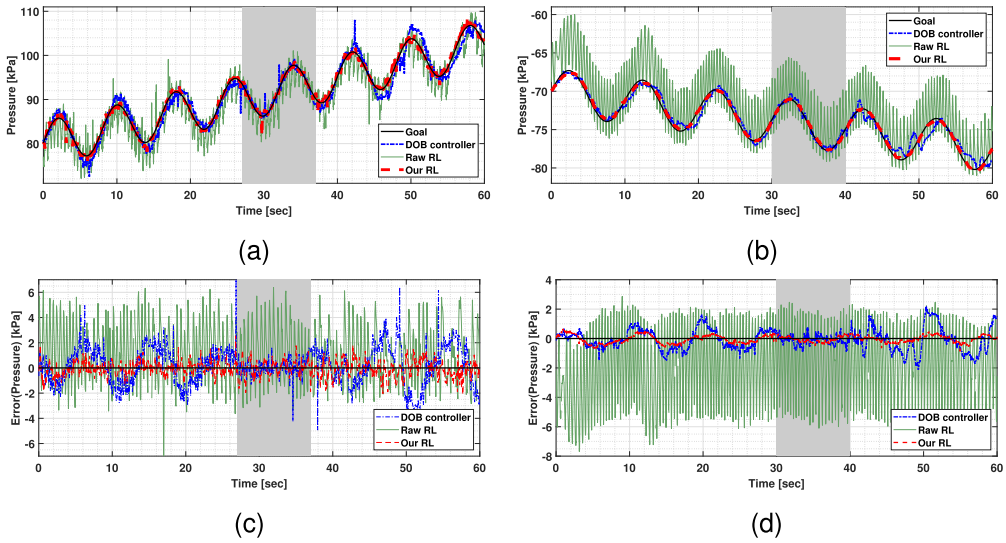

Fig. 6. Comparative analysis with sinusoidal reference on real system: (a) Sinusoidal input of the positive pressure. (b) Sinusoidal input of the negative pressure. (c) Absolute error of the positive pressure, (d) Absolute error of the negative pressure. The gray box highlights the effective tracking range of the DOB controller (27–37 s for positive pressure, 30–40 s for negative pressure), with poor performance outside these ranges, while the proposed RL controller shows superior performance across all ranges with the lowest RMSE (0.6935 kPa and 0.2646 kPa).

3) Reinforcement Learning Training and Application: All training procedures and real-time control operations were executed on an Apple M2 CPU without a GPU. The simulation training consisted of 512,000 steps (0.02 s per step in simulation time), corresponding to 3 hours of simulated time, and was completed in approximately 1.5 hours in real-world time. Following this, the sim-to-real transfer was conducted over 102,400 steps on the pneumatic system depicted in Fig. 4, taking roughly 34 minutes to complete. Additionally, real-time control was maintained at a 50 Hz frequency, ensuring stable and effective system operation.

4) Experiment Scenario: Four experiments were conducted: a validation test for simulator accuracy, a comparison test of controllers, an ablation test of RL controllers, and a robustness test for RL controller. To validate the simulator's accuracy, random control signals were applied, demonstrating the similarity between the simulator predictions and test data over a wide range. For the comparison and ablation tests, a sine wave with a varying range—expressed as the sum of a sine function and a linear function—was applied to evaluate the controller's high control capabilities and broad control range. This range exceeds the control range of the DOB-based linear controller, revealing the limitations of linear controllers. In contrast, the RL controller demonstrated controllability over a much wider range. To confirm the RL controller's robustness, experiments with varying loads showed it tracks a sine wave under load without retraining, validating its effectiveness. Those tests were conducted on real system explained in Section IV-A1.

B. Validation of Simulator Accuracy

The comparison of measured and modeled pressures, as illustrated in Fig. 5, demonstrates a close correspondence between the two. The accuracy rates of the pressures were found to be

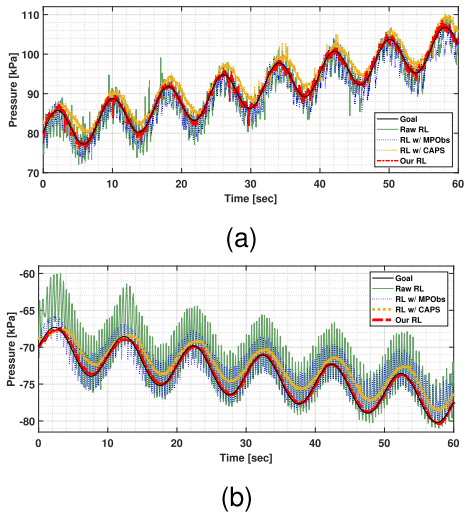

Fig. 7. Ablation analysis with sinusoidal reference on real system: (a) Sinusoidal input of the positive pressure. (b) Sinusoidal input of the negative pressure.

91.76% and 90.21%, respectively, calculated by substracting mean absolute precentage error from 1. These indicate a high level of accuracy for the developed simulator. These results validate the reliability of the simulator in capturing the dynamic behavior of the pneumatic system, thereby supporting its use in predictive simulations and real-time control applications.

C. Comparative Analysis of Controller Performance

To evaluate the performance of our RL-based controller on the pneumatic system, we conducted comparative experiments with a conventional DOB-based linear controller and a baseline RL controller lacking solutions without mechanisms to address

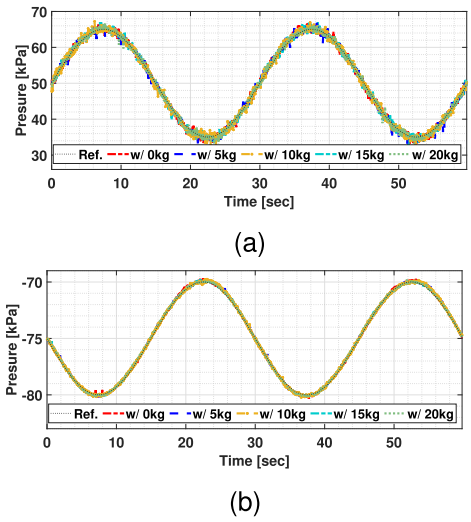

Fig. 8. Robustness analysis of the proposed RL-based controller. (a) Positive pressure tracking test of RL-based controller with 0 kg, 5 kg, 10 kg, 15 kg, and 20 kg external loads. (b) Negative pressure tracking test of RL-based controller with 0 kg, 5 kg, 10 kg, 15 kg, and 20 kg external loads. The RL-based controller tracks reference pressure in all experiments with maximum RMSE of those experiments is 0.7906 kPa for positive pressure and 0.1186 kPa for negative pressure.

oscillations and action smoothing. Sinusoidal inputs were applied, varying between 80 kPa and 100 kPa for positive, and -70 kPa and -75 kPa for negative pressure, with 8 seconds and 10 seconds periods, and 6 kPa and 3 kPa magnitudes. As illustrated in Fig. 6, the DOB controller performed well within 27 to 37 seconds range for positive pressure and 30 to 40 seconds range for negative pressure but showed poor tracking beyond this range. Similarly, the baseline RL controller struggled with tracking RMSE of 2.5439 kPa and 3.0254 kPa due to unresolved oscillations and action smoothing. In contrast, our RL controller excelled across all ranges, addressing oscillations and action smoothing issues effectively, highlighting its robustness for pneumatic systems with the smallest RMSE of 0.6935 kPa and 0.2646 kPa.

D. Ablation Test of RL-Based Controller

To evaluate the impact of MPObs and CAPS-based action smoothing in addressing oscillations, four controllers were compared: the raw RL controller, RL with MPObs, RL with CAPS, and our fully enhanced RL with both components. Using the same sinusoidal pressure profiles, the results Fig. 7 show that the baseline RL controller struggled with oscillations and large tracking errors. Adding MPObs reduced oscillations, improving tracking, while CAPS boosted robustness. The full RL controller, combining both MPObs and CAPS, outperformed all others, effectively addressing both issues. This validates the importance of MPObs and CAPS in enhancing control performance.

E. Robustness Test of the Proposed RL Controller

To evaluate the robustness of our RL-based controller, experiments with external loads were conducted. The hyper-VAM actuator [1], employing both positive and negative pressures concurrently, was selected for the experiments as it is best for validation of our controller's robustness. Testing involved a trained controller, with no further training, incrementally increasing the external load from 0 kg to 20 kg in 5 kg steps. We selected 20 kg as the maximum load because the actuator could not move beyond this load under the given reference pressures. The experimental results are shown in Fig. 8. Under loaded conditions, the proposed RL-based controller consistently tracks the target pressure without requiring additional training or tuning. As presented in Table II, it was confirmed that the RMSE remains within 0.7906 kPa for positive pressure and 0.1186 kPa for negative pressure across all load conditions, thereby demonstrating the robustness of the proposed approach to variations in external loads[1].

TABLE I
ROOT-MEAN-SQUARE ERRORS OF ABLATION TEST

	Positive [kPa]	Negative [kPa]
Raw RL	2.5439	3.0254
RL w/ MPObs	2.3965	1.7714
RL w/ CAPS	2.2935	1.4204
Our RL	**0.6935**	**0.2646**

TABLE II
ROOT-MEAN-SQUARE ERRORS OF ROBUSTNESS TEST

	Positive [kPa]	Negative [kPa]
w/ 0 kg	0.7451	0.1180
w/ 5 kg	0.7906	0.1181
w/ 10 kg	0.7508	0.1186
w/ 15 kg	0.7593	0.1095
w/ 20 kg	0.7012	0.1097

V. CONCLUSION

In this work, we developed an Reinforcement Learning(RL)-based controller that tracks both pressures over a wide range. By integrating a multiphysics pneumatic model—capturing the pump's kinematics and electromagnetic behavior—with a custom simulator, we accurately represented the coupled dynamics essential for efficient training. Our controller employs Model-Predicted Observation (MPObs) and Conditioning for Action Policy Smoothness (CAPS)-based smoothing. MPObs predicts future behavior using the pneumatic model to mitigate nonlinearities, while CAPS reduces abrupt input changes and minimizes oscillations from domain gaps. Experimental validation shows that our controller consistently outperforms conventional linear controllers (e.g., Disturbance Observer(DOB)-based methods) and maintains robust performance across varying loads, as evidenced by low RMSEs for both pressure directions. Ablation studies further demonstrate that combining MPObs and CAPS synergistically reduces control input oscillations. These results provide an effective strategy for addressing nonlinearities in pneumatic power sources, paving the way for more robust and versatile control solutions. Future work will extend this approach to additional control channels and focus on hardware miniaturization for portability, while optimizing the simulator to

[1]The experimental results are available in video at https://youtu.be/9vdT-9wRKYo

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

maintain high accuracy and low computational load as it scales for stand-alone field applications.

APPENDIX

TABLE III
SIMULATOR PARAMETERS

Name	Value
Ideal gas constant (R)	0.287 kJ/(kg · K)
Temperatures (T_{pis}, T_{ch})	323.15 K, 293.15 K
Max. effective length of the piston (δ)	4.1 cm
Piston slider-crank (r, l)	2 cm, 7 cm
Pump Motor angular velocity (ω)	3000 rpm
Piston phase (ϕ_1, ϕ_2)	0 °, 180 °
Piston head cross-sectional area (S_{pis})	38.485 cm^2
Specific heat ratio (κ)	1.4
Atmospheric pressuer (P_{atm})	101.325 kPa
Chamber volumes ($V_{\text{ch}}^{\text{pos}}$, $V_{\text{ch}}^{\text{neg}}$)	0.75 L, 0.4 L
Piston outlet check valve coeff. ($\widehat{C}_{\text{CV}}^{\text{out}}$)	1.46 mm^2
Piston inlet check valve coeff. ($\widehat{C}_{\text{CV}}^{\text{in}}$)	33.47 mm^2
Pos. channel valve input coeff. ($\widehat{C}_u^{\text{pos}}$)	4.876 mm^2
Pos. channel valve press. coeff. ($\widehat{C}_P^{\text{pos}}$)	3.066 mm^2/N
Pos. channel valve spring coeff. ($\widehat{C}_k^{\text{pos}}$)	4.474 mm^2
Neg. channel valve input coeff. ($\widehat{C}_u^{\text{neg}}$)	7.992 mm^2
Neg. channel valve press. coeff. ($\widehat{C}_P^{\text{neg}}$)	7.590 mm^2/N
Neg. channel valve spring coeff. ($\widehat{C}_k^{\text{neg}}$)	7.339 mm^2
Time step of simulator (t_s)	0.0001 sec

TABLE IV
REINFORCEMENT LEARNING HYPER-PARAMETERS

Name	Value
Control frequency (f)	50 Hz
Number of previous observations (n_p)	10
Number of predicted observations (n_f)	15
Number of action trajectory (n_a)	5
Reward weight, $w_{r,0}$	-0.3
Reward weights, $w_{r,1}, \ldots, w_{r,n_a-1}$	-0.01
Reward weight, w_{r,n_a}	-0.25
Number of simulation training steps	512000
Number of sim-to-real transfer steps	102400
Learning rate	3×10^{-4}
Discount factor (γ)	0.9
Replay buffer size	5×10^5
Number of hidden layers	2
Number of hidden units per layer	256
Number of samples per minibatch	128
Nonlinearity	ReLU
Temperature parameter (α)	Auto. ent. adj. [25]
Entropy target	$-\dim(\lvert\widehat{\mathbf{a}}\rvert)$
Target smoothing coefficient (τ)	0.005
Optimizer	Adam [30]
Sim. temporal regularization weight (λ_T)	1.0
Sim-to-real temp. reg. weights ($\lambda_T^{\min}$, $\lambda_T^{\max}$)	1.0, 3.0
Sim-to-real temp. reg. hardening rate (Δ_{λ_T})	3.906×10^{-5}
Spatial regularization weight (λ_S)	0.4
Spatial regularization noise (σ)	1.5
Integral term gains	0.01
Anti-windup gains	10

REFERENCES

[1] A. Coutinho, J. H. Park, B. Jamil, H. R. Choi, and H. Rodrigue, "Hyperbaric vacuum-based artificial muscles for high-performance actuation," *Adv. Intell. Syst.*, vol. 5, no. 1, 2023, Art. no. 2200090.

[2] J. H. Park et al., "Cooperative antagonistic mechanism driven by bidirectional pneumatic artificial muscles for soft robotic joints," *Mechatronics*, vol. 97, 2024, Art. no. 103099.

[3] D. Hu, J. Zhang, Y. Yang, Q. Li, D. Li, and J. Hong, "A novel soft robotic glove with positive-negative pneumatic actuator for hand rehabilitation," in *Proc. 2020 IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, 2020, pp. 1840–1847.

[4] D. Drotman, S. Jadhav, D. Sharp, C. Chan, and M. T Tolley, "Electronics-free pneumatic circuits for controlling soft-legged robots," *Sci. Robot.*, vol. 6, no. 51, 2021, Art. no. eaay2627.

[5] A. Talhan, Y. Yoo, and J. R. Cooperstock, "Soft pneumatic haptic wearable to create the illusion of human touch," *IEEE Trans. Haptics*, vol. 17, no. 2, pp. 177–190, Apr.–Jun. 2024.

[6] L. Mišković, M. Dežman, and T. Petrič, "Pneumatic exoskeleton joint with a self-supporting air tank and stiffness modulation: Design, modeling, and experimental evaluation," *IEEE/ASME Trans. Mechatron.*, vol. 29, no. 5, pp. 3415–3426, Oct. 2024.

[7] Y. Cao, M. Zhang, J. Huang, and S. Mohammed, "Prescribed performance control of a link-type exoskeleton powered by pneumatic muscles with virtual elasticity," *Nonlinear Dyn.*, vol. 112, no. 12, pp. 10043–10060, 2024.

[8] G. Liu, S. Diao, T. Yang, X. Zhang, Y. Fang, and N. Sun, "Supervised learning control for compliant pneumatic artificial muscle robots with preassigned-time performance," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 54, no. 9, pp. 5352–5364, Sep. 2024.

[9] P. Chen, Q. Ding, Y. Liu, Z. Deng, and J. Huang, "Programmable pressure control in pneumatic soft robots with 2-way 2-state Solenoid valves," *IEEE Robot. Automat. Lett.*, vol. 9, no. 7, pp. 6448–6455, Jul. 2024.

[10] M. S. Xavier, A. J. Fleming, and Y. K. Yong, "Design and control of pneumatic systems for soft robotics: A simulation approach," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5800–5807, Jul. 2021.

[11] C. Park et al., "Simultaneous positive and negative pressure control using disturbance observer compensating coupled disturbance dynamics," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 5763–5770, Apr. 2022.

[12] OpenAI: Marcin Andrychowicz et al., "Learning dexterous in-hand manipulation," *Int. J. Robot. Res.*, vol. 39, no. 1, pp. 3–20, 2020.

[13] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," in *Proc. Robot., Sci. Syst.*, 2020, pp. 12–16.

[14] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[15] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. 4th Int. Conf. Learn. Representations*, 2016, pp. 1–14.

[16] A. R. Mahmood, D. Korenkevych, G. Vasan, W. Ma, and J. Bergstra, "Benchmarking reinforcement learning algorithms on real-world robots," in *Proc. 2nd Conf. Robot Learn.*, 2018, pp. 561–591.

[17] S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko, "Regularizing action policies for smooth control with reinforcement learning," in *Proc. 2021 IEEE Int. Conf. Robot. Automat.*, 2021, pp. 1810–1816.

[18] J. F. Blackburn, G. Reethof, and J. L. Shearer, *Fluid Power Control*. New York and London: Technology Press of MIT and Wiley, 1960.

[19] J. J. Uicker, J. J. Uicker Jr, G. R Pennock, and J. E Shigley, *Theory of Machines and Mechanisms*. Cambridge, MA, USA: Cambridge Univ. Press, 2023.

[20] W.H. Jr. Hayt and J.A. Buck, *Engineering electromagnetics*, 8th ed., New York, NY: McGraw-Hill, 2010.

[21] D. Ben-Dov and S. E. Salcudean, "A force-controlled pneumatic actuator," *IEEE Trans. Robot. Autom.*, vol. 11, no. 6, pp. 906–911, Dec. 1995.

[22] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.

[23] E. F. Camacho and C. B. Alba, *Model Predictive Control*, 2nd ed. Berlin, Germany: Springer, 2013.

[24] 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. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

[25] T. Haarnoja et al., "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.

[26] H. Nguyen and H. La, "Review of deep reinforcement learning for robot manipulation," in *Proc. 3rd IEEE Int. Conf. Robot. Comput.*, 2019, pp. 590–595.

[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[28] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[29] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1587–1596.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.