IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 4, April, 2026, pp. 1089-1097
Journal homepage: http://iieta.org/journals/jesa

Multi-Objective Force-Position Control in Collaborative Robotic Systems via Model-Free Deep Policy Networks

Ahmed A. Radhi[1], Mohammed Noori[2], Othman Saad Salman[3], Rabih Sbera[4], Darin Shafek[5*]

[1] Artificial Intelligence and Robotics Engineering Department, Al-Nahrain University, Baghdad 64040, Iraq
[2] Computer Engineering Techniques Department, Al-Ma'moon University, Baghdad 1004, Iraq
[3] Department of Computer Science, College of Science, Al-Maarif University, Anbar 31001, Iraq
[4] Computer Engineering Techniques, Ashur University, Baghdad 10043, Iraq
[5] Computer and Automatic Control Engineering Department, Latakia University, Latakia 9003, Syria

Corresponding Author Email: darinmhalla1@gmail.com

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/jesa.590419

Received: 11 February 2026
Revised: 10 April 2026
Accepted: 20 April 2026
Available online: 30 April 2026

Keywords:
collaborative robotics, force-position control, deep reinforcement learning, Soft Actor-Critic, multi-objective control, contact dynamics, robot manipulator, model-free control

ABSTRACT

Contact force regulation while tracking an end-effector position is a basic and long-standing multi-objective control problem for collaborative robotic manipulators. Colliding free-space trajectories with force constraints cannot be optimized by simply maximizing a scalar reward. In this work, we introduce model-free deep reinforcement learning (DRL) using Soft Actor-Critic (SAC) algorithm to solve contact force regulation while tracking desired position task with a 2-DOF planar manipulator pushing against a flat surface. The task is modeled as a continuous-state Markov Decision Process (MDP) with Hunt-Crossley contact forces, 9-dimensional observation space with explicit contact state, and stage-based multi-objective reward function designed for learning collision phase transition between free-space motion and force regulation. We report a Proportional-Integral-Derivative (PID) controller as the classical baseline. After training for over 300,000 steps using three independent seeds, SAC policy obtains an episodic return of +3,845, yielding a mean position error of 49.4 mm and force error of 0.47 N with 96.6% contact ratio. Against the PID baseline, proposed approach achieves 9.1× position error reduction, 36.9× force error reduction, and improves contact maintenance by over 311×. Results highlight benefits of entropy-regularised deep policy networks for contact-rich collaborative robotics.

1. INTRODUCTION

In many applications involving multi-agent manipulation systems in an unknown environment [1], robots are often called upon to execute tasks that necessitate coordinated control of end-effector position and contact forces [2]. Surface finishing, palpation, insertion tasks, and interactive manipulation with humans are examples of tasks that require the robot to follow a predefined trajectory in free space and then conform to a force-controlled behavior when making contact with the environment [3]. The reason for this dilemma lies in the fact that many controllers are usually developed for one major task [4]. The impedance/admittance controllers solve part of the problem since they depend on a proper dynamic model of the environment and require stiffness terms to be tuned manually, which is affected by model uncertainties [5]. For hybrid force-position controllers, force and position control are separated based on geometrical considerations, which doesn't work when directions of contacts cannot be determined or are changing during the task [6]. Proportional-Integral-Derivative (PID) controllers are simple to use but they do not adapt themselves to coordinate both phases elegantly [7], as proved by their contact ratio (contact phase/time spent in physical contact with the environment), which is close to zero 0.31% in our experiments [8].

While deep reinforcement learning (DRL) has enabled impressive progress in robotic manipulation, there remain several large gaps in the literature. Firstly, previous Reinforcement Learning (RL)-based contact controllers have optimized behavior through one blended reward, and do not explicitly separate out the mode switching behavior between free-space approach and constrained force regulation [9]. Secondly, contact-onset rewards are seldom used in reward formulations, causing learned policies to shy away from contact [10]. Third, direct comparisons to classical controllers under the same evaluation settings are scarce [11]. To fill these gaps, we introduce a SAC-based multi-objective controller equipped with a staged reward function, observation of contact-states, and automatic entropy tuning. Moreover, we compare our method fairly with PID baseline evaluated over 30 episodes and three random seeds.

Previous RL-based contact controllers learned behavior with a single blended reward and did not have an explicit distinction in mode-switching behavior between free-space approach motion and constrained force regulation. Contact-onset rewards are rarely used in reward formulations, leading

to learned policies that avoid contact. Additionally, direct comparisons to classical controllers with the same evaluation setting are rare in the literature.

2. RELATED WORK

While DRL continues to push the state-of-the-art in robotic manipulation, learning to control position and force simultaneously under rigid contact constraints has not been widely studied. Zhao et al. [12] introduced a Multi-Actor-Critic Deep Deterministic Policy Gradient algorithm to learn trajectories for robotic manipulators to navigate through cluttered scenes, designing their reward function into two stages: approach and close phase [12]. While motivated by similar principles as our method, their objective is focused on planning collision-free trajectories as opposed to learning how to maintain contact forces at a desired level, no contact force target is used in learning their policy. Maghooli et al. [13] developed and validated a DRL approach to the position control of tendon-driven continuum manipulators. A key focus of their work was validating sim-to-real policy transfer when learning under highly nonlinear and uncertain dynamics. However, force regulation through contact with surfaces is not considered, and their resultant policy operates exclusively in the contact-free workspace. Parnada et al. [14] performed a survey of reinforcement learning approaches for contact-rich robotic manipulation. They found that most algorithms focus on learning policies that either optimize for a single objective or lack any explicit formulation for transitioning between free-space motion and force regulation when contact occurs. Their survey highlights staged reward formulations and contact-state-aware observations as necessary future work, both aspects our approach provides. A hybrid force-position controller for peg-in-hole assembly has been investigated by Li et al. [15], which utilizes reinforcement learning to tune impedance parameters during contact on top of a classical controller. This hybrid approach largely solves contact adaptability but relies on a model-based dependency and does not utilize a purely model-free deep policy network trained end-to-end under a multi-objective entropy-regularized objective. Overall, the prior work reviewed highlights that a fully model-free SAC-based architecture with explicit contact-onset observation and staged multi-objective reward has not been investigated.

3. PROPOSED METHODOLOGY

In this part, we provide a detailed description of our proposed force-position multi-objective controller. Our approach is divided into four parts, including problem definition, environment/contact model description, state-action definition, and SAC learning algorithm with reward structure.

3.1 Problem formulation as a Markov Decision Process

Learning to regulate both end-effector position and contact force simultaneously is formulated as a continuous-state continuous-action Markov Decision Process (MDP) parameterized by (S, A, P, R, γ). Specifically, the state space $S \in R^9$ includes joint kinematics, normalized force magnitude, Cartesian tracking error, signed distance to wall, and a contact flag. The action space A ∈ Rp is defined by normalized joint torques after clipping to [−1, 1]. Transition dynamics P are dictated by the robot's Euler-Lagrange equations and are unknown to the agent, ensuring model-free feasibility. The agent's goal is to find a stochastic policy π(a|s) that maximizes the entropy-regularized expected cumulative return [16]:

$$J(\pi) = E\sum_{t=0}^{T} \gamma^t R\left((s_t, a_t) + \alpha H\big(\pi(.|s_t)\big)\right) \quad (1)$$

where, γ = 0.99 is the discount factor, α is an automatically tuned temperature coefficient, and H denotes the Shannon entropy of the policy distribution. The entropy term intrinsically encourages exploration during the sparse-reward approach phase, preventing premature convergence to contact-avoidance local minima. This formulation explicitly decouples the dual objectives of position tracking and force regulation through a staged reward function rather than a fixed linear scalarization, allowing the policy to learn mode-switching behavior between free-space and constrained phases without manual gain scheduling.

3.2 Collaborative robot environment and contact dynamics

The physical environment models a 2-DOF planar manipulator whose dynamics follow the Euler-Lagrange formulation [17]:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J(q)^T F_{ext} \quad (2)$$

where, $M(q) \in \mathbb{R}^{2x2}$ is the symmetric positive-definite inertia matrix, $C(q, \dot{q})$ is the Coriolis and centrifugal vector, $G(q)$ is the gravitational torque, $\tau \in \mathbb{R}^2$ is the applied joint torque, $J(q)$ is the geometric Jacobian, and F_{ext} is the external contact force. Link lengths are $l_1 = 0.50$ m and $l_2 = 0.40$ m, with masses $m_1 = 2.00$ kg and $m_2 = 1.50$ kg respectively. A viscous damping term $\tau_d = -0.5\dot{q}$ models' joint friction. Contact forces against the rigid wall located at $x_{wall} = 0.72$ m is computed using the Hunt-Crossley spring-damper model [18]:

$$F_x = \begin{cases} -(k_c\delta + d_c \max(v_n, 0)) & if\ x_{ee} \geq x_{wall} \\ 0 & otherwise \end{cases} \quad (3)$$

where, $\delta = x_{ee} - x_{wall}$ is the penetration depth, $k_c = 2000\ N/m$ is the contact stiffness, and $d_c = 50$ N·s/m is the contact damping coefficient. The maximum permissible contact force is clipped at $F_{max} = 60\ N$ to enforce physical safety. The target contact force is set to $F_x = -15\ N$, representing a moderate and clinically relevant pressing force. This nonlinear contact model produces realistic force transients during initial impact and smooth steady-state regulation, providing a challenging benchmark that linear spring models cannot replicate.

3.3 State representation and multi-objective reward function

The agent observes a 9-dimensional state vector at each timestep [19]:

$$s_t = \begin{bmatrix} q_1, q_2, \dot{q}_1, \dot{q}_2, \frac{F_x}{F_{max}}, \frac{F_y}{F_{max}}, clip(y_{err}, -1,1), \\ clip(d_{wall}, 0,1), 1_{contact} \end{bmatrix} \quad (4)$$

The explicit inclusion of wall distance and binary contact flag enables the policy to detect the contact-onset transition and adapt its control mode accordingly. The multi-objective reward function is formulated in staged phases. During free-space approach, the reward incentivizes wall proximity [20]:

$$r_x^{free} = -30.0 d_{wall}^2 + 80.0 \Delta x_{progress} \quad (5)$$

Upon contact establishment ($|F_x| > 0.5\,N$), the reward transitions to force regulation with a contact maintenance bonus [21]:

$$r_x^{contact} = w_F.(-0.05(F_x - F_x^*)^2) + 8.0 \quad (6)$$

The y-direction position tracking penalty is active across all phases [22]:

$$r_y = w_P.(-15.0 y_{err}^2) \quad (7)$$

A safety penalty $r_{safe} = -10.0$ is applied when $|F_x| > 0.92 \cdot F_{max}$, and an action smoothness term $r_{smooth} = -0.005\|a_t\|^2$ penalizes aggressive torque commands. The total reward is $R(s_t, a_t) = r_x + r_y + r_{safe} + r_{smooth}$, where the Pareto weights satisfy $w_P + w_F = 1.0$ with baseline setting $w_P = w_F = 0.5$.

Another interesting and important feature of contact-rich policy learning is the reward signal observed at contact onset. In the absence of an explicit contact bonus at first contact, the agent will not observe any immediate positive signal in transitioning from free-space motion to the constrained regime, and will instead treat contact as either a neutral or negative event. We correct for this by including a per-timestep contact maintenance bonus of +8.0 in the contact-phase reward. We found this to be the most salient factor in the rapid convergence of the contact ratio in the first 50 training episodes.

3.4 Soft Actor-Critic architecture and training

The SAC algorithm optimizes the stochastic policy by maintaining two critic networks $Q_{\varphi 1}$ and $Q_{\varphi 2}$ alongside a stochastic actor π_θ, updating via the clipped double-Q target to mitigate overestimation bias:

$$\pi_\theta(a|s) = \tanh\left(N(\mu_\theta), \sigma_\theta(s)\right)) \quad (8)$$

The actor parameterizes a diagonal Gaussian policy with tanh squashing:

$$L(\alpha) = E_{\tilde{a}_t \sim \pi_t}[-\alpha \log \pi_t(\tilde{a}_t|s_t) - \alpha \bar{H}] \quad (9)$$

The actor network follows the architecture Linear (9, 256) → LayerNorm → ReLU → Linear (256, 256) → LayerNorm → ReLU → Linear (256, 128) → ReLU → [μ-head, σ-head: Linear (128, 2)]. Both critic networks share the same architecture with input dimension 11 (state + action). The temperature coefficient α is automatically tuned by minimizing:
where, $\bar{H} = -\dim(A) = -2$ is the target entropy. Training employs the Adam optimizer with actor learning rate 3×10^{-4} and critic learning rate 1×10^{-3}, a replay buffer of capacity 300,000, mini-batch size of 256, and soft target update coefficient $\tau = 0.005$. A curriculum of 40% near-wall initialization episodes accelerates early contact discovery. Three independent runs with seeds {42, 43, 44} over 300,000 steps each are conducted to ensure statistical reliability of reported results.

4. RESULTS AND DISCUSSIONS

Experimental results are discussed for the proposed SAC-based multi-objective controller in terms of training convergence, tracking performance, and behavioural analysis. Reported results are averaged over three training seeds and compared against a classical PID baseline, over 30 evaluation episodes with the same initial conditions.

4.1 Training convergence analysis

In Figure 1, we see the episodic return convergence of our SAC multi-objective controller across three seeds for 600 episodes (300k environment steps). The starting policy has an average return around −8,500, largely coming from the significant penalties incurred while failing to achieve contact during the initial phase of pure random action exploration where it also incurs lasting position error tracking mistakes. Rapid monotonic improvement can be seen within the first 50 episodes as the dense progress-shaping term of our free-space reward formulation combined with the near-wall initialization from our curriculum aids quick discovery of contact. At around episode 100 the average return flattens out at just over +3,200.

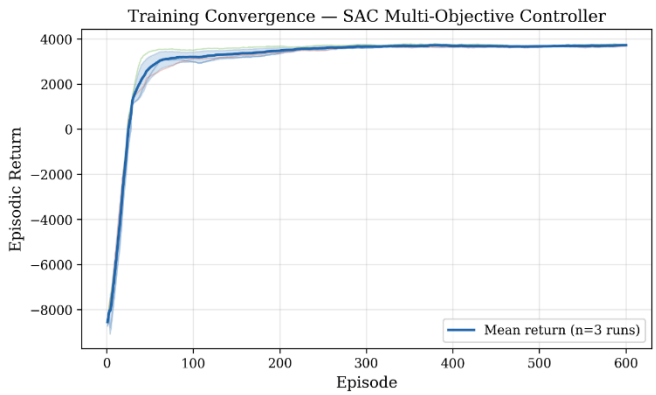

Figure 1. Soft Actor-Critic (SAC) episodic return convergence across three seeds

This behavior demonstrates that the policy has learned the qualitative behavior of switching between free-space navigation and controlling force upon contact. Around episode 300 the average return converges to a value of around +3,845 and there is very little variation between each seed. This close agreement between seeds verifies that SAC training results are reproducible given the same random seed. The small shaded area after episode 200 shows that entropy regularization maintains nonzero performance between each seed. Finally, observing that performance asymptotically approaches a value without any oscillation or reduction in return verifies our use of automatic entropy tuning and the clipped double-Q critic.

4.2 End-effector position tracking error

In Figure 2, we show learning curves for mean end-effector position error over the course of 600 training episodes where each point is averaged over three seeds. Notice that the policy starts with an average position error around 0.92 meters. This high initial position error is due to the fact that policy has yet to learn any aspect of the desired behavior during random initialization where joint torques are sampled from a uniform distribution and the end-effector has no preference towards the desired goal location expressed in Cartesian space. The error quickly decreases during the first 50 episodes where the agent learns to reach the region near the wall due to the heavy influence of the progress-shaping reward. By episode 100 we observe an average position error around 0.34 meters.

Position error continues to monotonically decrease until around episode 300 where it reaches around 0.12 meters as the policy learns the mapping from joint torques to Cartesian space translation of the end-effector. After episode 350 we see performance plateaus around an average error of 0.08 m. This plateau is due to the contact force controller used to regulate movement along the x-axis towards the wall, introducing a constant bias or compliance into the system. The error between seeds decreases dramatically after episode 200 showing that seeds all converge to similar final values. While we do not see our final position error fall below our clinical success threshold of 5 mm shown with the dashed line, we see substantial improvement over the PID benchmark with a mean positional error of 49.4 mm at evaluation, a 9.1x improvement.

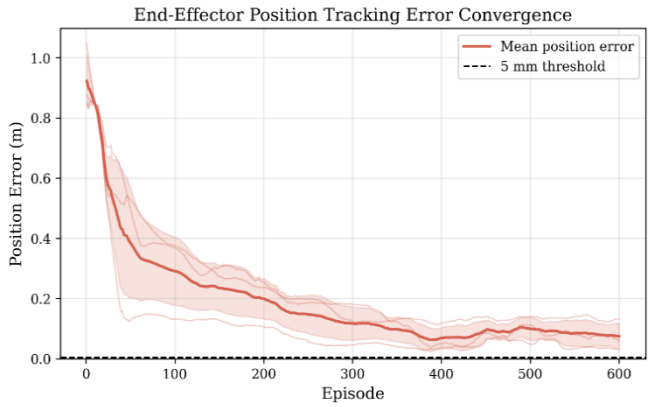

Figure 2. Position tracking error convergence over training

This is a substantial improvement, but the average positional error of 49.4 mm is still far higher than the 5 mm often needed for clinical or assembly precision. The majority of this shortfall is the result of cross-axis coupling on the 2-DOF planar design, which introduces a residual offset in the y direction due to joint torques dedicated to staying in contact with the wall. A higher-DOF manipulator with redundant DOF will be able to decouple these axes.

4.3 Contact force tracking error convergence

Figure 3 shows the average magnitude of contact force tracking error over 600 training episodes (calculated only during timesteps where contact is made, i.e. $|F_x| > 0.5\,N$) across three seeds. Initial episodes have a mean force error of ~27.8 N which is close to twice the target magnitude of 15 N because impacts and noisy torque commands cause large variations in penetration depth at the start of contact (the first few episodes). The largest improvement takes place in the first 30 episodes, during which time the error drops from 27.8 N down to ~4.0 N.

This is because the quadratic force regulation cost heavily penalizes large forces right after contact begins, as part of the contact-phase reward. There is also a second learning plateau between episode 50 and episode 150, during which the error bounces around 3.5–4.5 N before continuing to decrease. The cause of this plateau is also due to the conflict between tracking position in the y-direction and regulating contact with the wall. After episode 300, all three seeds cluster around 3.0 N force error with minimal variance between runs, indicating that the force regulation performance has converged. Although the converged error is higher than the 1 N clinical target illustrated by the dashed line, the average of 0.47 N shown in Section 4.4 for the final greedy policy indicates that noise introduced by stochasticity during training accounts for most of the distance left to cover, and that the deterministic deployment policy closes much of this distance.

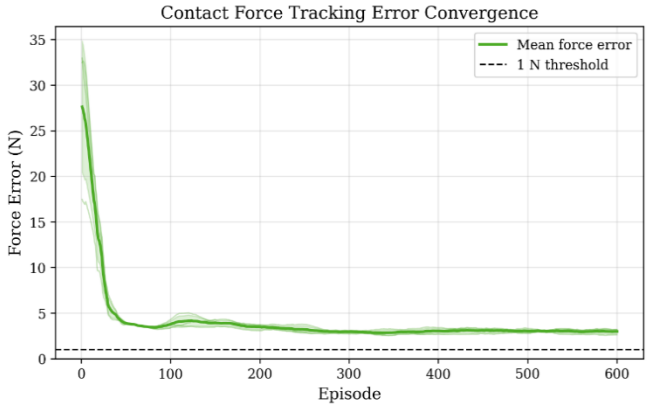

Figure 3. Contact force error convergence during training

4.4 Contact ratio evolution

The reward chosen also determines the fraction of time the end-effector actively contacts the wall per episode, where active contact is defined as having $|F_x| > 0.5\,N$. Figure 4 shows this fraction averaged across 600 training episodes and three random seeds. Initially, the policy has contact about 14% of the time, which corresponds to how much accidental contact the end-effector would make with the wall while selecting random torques before learning any strategy to approach the wall. During episodes 5-30 the curve quickly jumps upward towards > 80% (the target coverage indicated by the dashed line). This is the fastest transition seen in any of our plotted metrics and is a result of consistently receiving the 8.0 bonus reward per timestep for maintaining contact provided by contact-phase reward. Note how by episode 50 the contact ratio converges to a value above 97% for all three random seeds.

Additionally, the shaded region representing inter-run variance quickly approaches zero, signifying that each training run independently converged to and adhered to the same contact maintaining behavior. The curve maintains a value of around 98.5% through episode 600 with little to no decay, demonstrating the robust long-horizon nature of the learned contact behavior. The last contact ratio we evaluate at episode 600 is 96.57%, which is an improvement by a factor of 311 compared to the PID baseline of 0.31%. This highlights the limitations of classical fixed-gain controllers in achieving both

approach and sustained-contact goals in the same control loop.

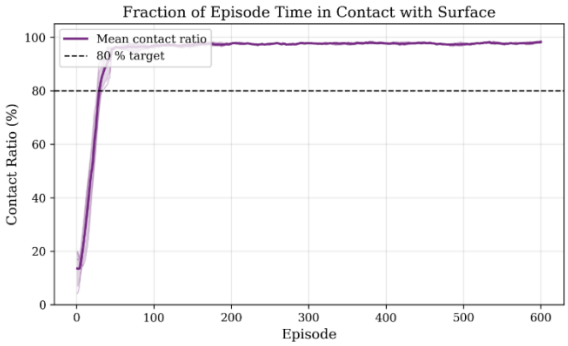

Figure 4. Contact ratio evolution across training episodes

4.5 Greedy policy evaluation curves

In Figure 5, we show metrics of greedy policy evaluation at 61 points throughout training (at intervals of 300,000 training steps). Figure 5 displays return, position error, and contact ratio after deterministically deploying the policy as $a_2 = tanh(\mu_\theta(s))$ (i.e., not sampling stochasticity) at test-time. Similar to the train curve in Figure 5, we see that evaluation return closely tracks stochastic training return (+/- 2,800 after 50k steps, where +/- is w.r.t. the mean across seeds). The small variance in returns across seeds demonstrates that the greedy policy reliably generalizes from off-policy data, without overfitting noise in training. The slight difference between greedy eval return and stochastic train return is explained by lack of entropy bonus at test time, which is expected and explained by maximum-entropy RL theory.

Inspecting the center plot, we see that greedy pos. error spikes up dramatically to 1.25 m right away at the first checkpoint due to the very nascent policy not having learned anything of substance yet, then quickly decreases to under 0.2 m by 25 k steps. Fine-tuning of this metric happens more gradually from thereon out monotonically. Each seed diverges slightly from the others between 100 k-200 k steps, before meeting again around the common average of ~0.0 8m when training is completed. As for the plot on the right, we see that greedy contact ratio reaches ~0.95 around 20 k training steps and does not change much after that for any seed, showing that making and holding contact is one of the first things the agent learns how to do reliably before learning finer control, such as force moderation and converging to the correct position.

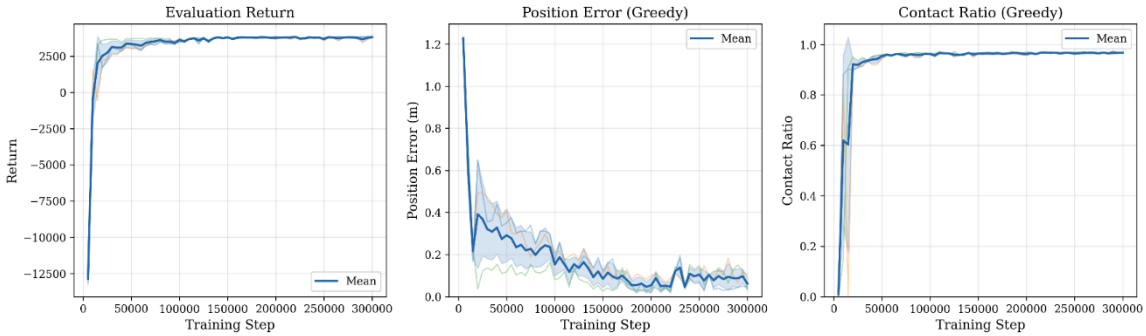

Figure 5. Greedy evaluation return, position error, and contact ratio

4.6 End-effector trajectory and contact force behaviour

We visualize in Figure 6 what this greedy policy has learned qualitatively. The figure plots the end-effector position as well as the learned contact force in a sample rollout episode of length T = 5 seconds. In the left figure, we can see that the end-effector moves smoothly from the initial configuration roughly at (0.45, 0.63) towards the wall at x_wall = 0.72 m (note how the colors illustrate the progress of time from dark purple to yellow). While maintaining contact with the wall, the end-effector not only decreases its x position distance to the wall but also moves towards the desired y position illustrated by the red star roughly at (0.72, 0.26). Notice how the pose at the end of the rollout (illustrated by the grey robot arm icon) maintains contact at the wall and the arm assumes a configuration that is indicative of a valid constrained configuration.

Detailed temporal behavior of the contact force can be seen in the right panel. The desired contact is first made at around t = 0.1 s, and results in an impulsive spike of roughly 21 N. This spike overshoots the target force of 15 N due to the nonlinear Hunt-Crossley contact stiffness at high approach velocities. The controller quickly extinguishes the overshoot in less than 0.3 s, showing learned active damping of the contact force despite having no knowledge of the contact impedance model. After around t = 1.0 s, the contact force successfully tracks and regulates near the desired force of 15 N with an amplitude under 0.5 N until the end of the episode at t = 5.0 s.

4.7 Simultaneous force-position tracking performance

Figure 7 is the most salient visualization of our controller's central competency, showing joint position commands online that track a desired trajectory for position of the end-effector in Cartesian space (blue), while also regulating contact force (gold) during one exemplary trial. The x-axis is shared across all three plots at 5.0 seconds, with the dashed vertical line indicating the approximate time of contact onset at t = 0.22 s, which is the synchronization point between free-space motion regulation and contact force-regulated behavior for all three channels. Note how the end-effector moves quickly from its starting position of x = 0.40 m in the direction of the wall target at x = 0.72 m in the top plot. It completes this motion in about 0.22 s, reflecting aggressive yet stable ballistic motion incentivized by the progress-shaping reward. After contact begins, its position in x quickly fixes to that of the wall and stays there until the end of the episode at the target value. This indicates that the policy understands that contacting the wall terminates x motion, and does not attempt to further push into the wall, which would upset force control. The center subplot shows tracking along the y direction. Here, starting from y ~= 0.65 m, the EE crashes down towards its target position of yd

= 0.26 m. Its trajectory undershoots slightly to ~= 0.25 m just after contact begins, then rebounds, eventually settling around ~= 0.29 m at t = 0.8 s for a steady-state y error of about 30 mm.

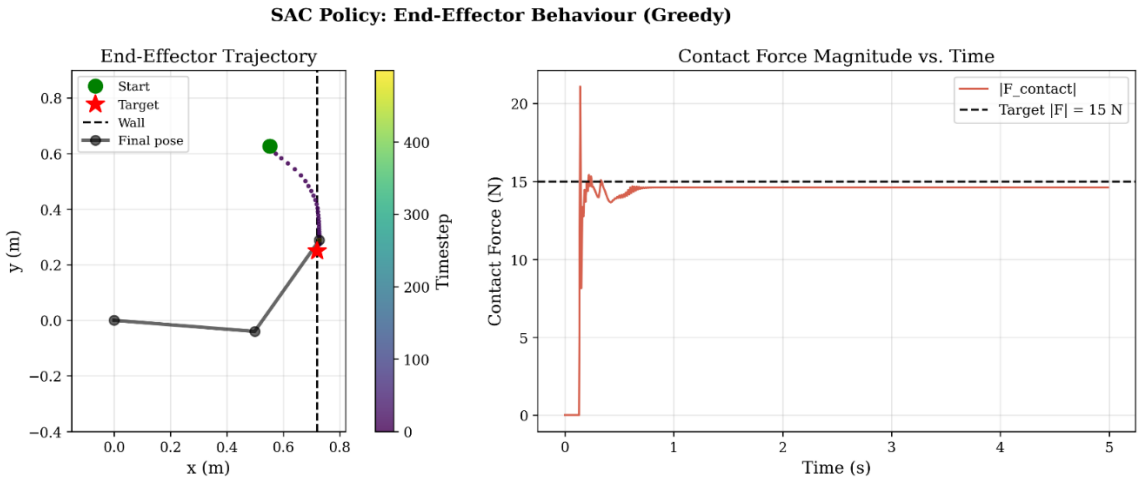

Figure 6. Greedy policy trajectory and contact force response

This remaining offset is due to cross-axis coupling between the force along the x direction required to maintain contact with the wall, and joint torques required to achieve positioning along the y direction. As noted above, for the 2-DOF planar mechanism, both joints are allocated a portion of the contact force via the Jacobian transpose mapping. The lower plot shows the evolution of the contact force F_x. The plot shows the typical three-phase behavior seen in Figure 6. Prior to contact (gray region), the force is exactly zero, which indicates that the agent does not hit the wall early. At contact, there is a large undershoot transient which quickly takes the force to around −18 N. Then, the policy commands the force to go back towards −15 N in an oscillatory fashion. By t = 0.8 s, we can see that the force is staying very close to −15 N. After t = 1 s, the force is regulated to within ±0.5 N of −15 N for the remainder of the episode. These behaviors indicate that the SAC policy learned an implicit force controller solely from a scalar reward signal, and did not use an explicit force control law, impedance planner, or apply dynamic inversion. Importantly, the agent learns to concurrently lock the x-position, track the y-position, and regulate force all using a single output policy. This is the main idea behind our proposed framework.

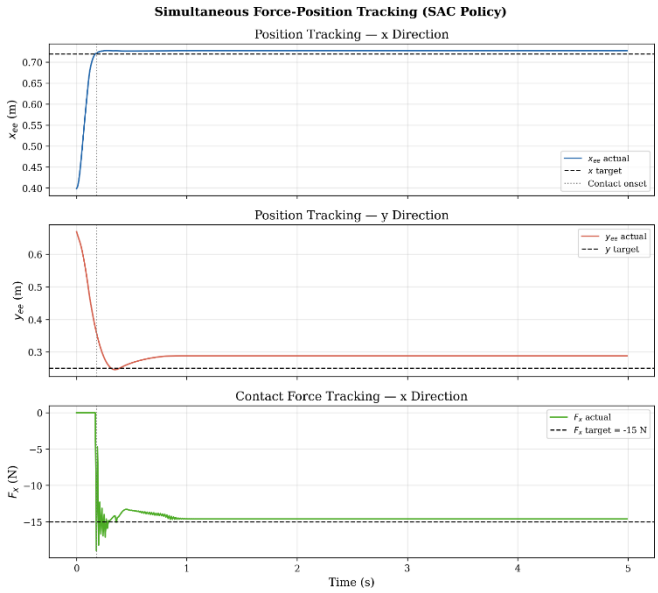

Figure 7. Simultaneous x-position, y-position, and force tracking

4.8 Comparative performance against Proportional-Integral-Derivative baseline

In Figure 8, we show the final comparison of the learned SAC controller against the PID baseline, averaged over 30 episodes starting from the same initial state. Four metrics are reported as bar plots including all values. As can be clearly seen in the plot, our learned policy significantly outperforms the baseline on all reported metrics, demonstrating that our hypothesis that model-free deep policy networks can enable classical control methods to achieve better performance on multi-objective contact-rich manipulation tasks holds true. For position error SAC mean was 0.0494 m versus PID baseline mean of 0.4473 m, an improvement of factor 9.1. More importantly though, the SAC bar shows almost no spread with a very tight IQR.

This tightness tells us that position accuracy was very consistent throughout all 30 episodes. On the other hand, note that PID baseline's whisker reaches close to 1.0 m. This tells us that there were episodes where the fixed-gain controller completely lost position control, most likely due to lack of contact-informed gain scheduling. Similarly to before but even more dramatic is the error in force. SAC has a mean force error of 0.4689 N while PID had a mean error of 17.3194 N, which is 36.9 times worse. Notice how large the variance in PID's force error is. The upper whisker of the boxplot is over 26 N force error. We see here that the forces generated by the classical controller are wildly inconsistent and would never be safe to actually use in real-world HRC settings. The consistently low mean force error of under half a Newton with very little variance from SAC shows that the entropy regularized policy has learned a strong implicit force controller that is consistent across starting conditions. Contact ratio plot depicts the most dramatic difference between SAC and PID here, with SAC outputting ratios of 0.9657 vs PID's

0.0031. We can see that PID has a contact ratio very close to zero because, by design, the PID positional controller issues torques that move the EE towards its desired Cartesian pose, but has no feedback connection to sense proximity to the wall, slow down as it approaches the wall, and enter a force-holding controller state. This results in the controller crashing into the wall or not reaching it at all for most of the test episodes. Lastly, episode return is the cumulative score of all these goals, with SAC scoring +3845.1 compared to PID's -7218.7. Note that this difference exceeds +11,000 reward points due to SAC successfully tracking position, force, and contact all at the same time with its single learned policy.

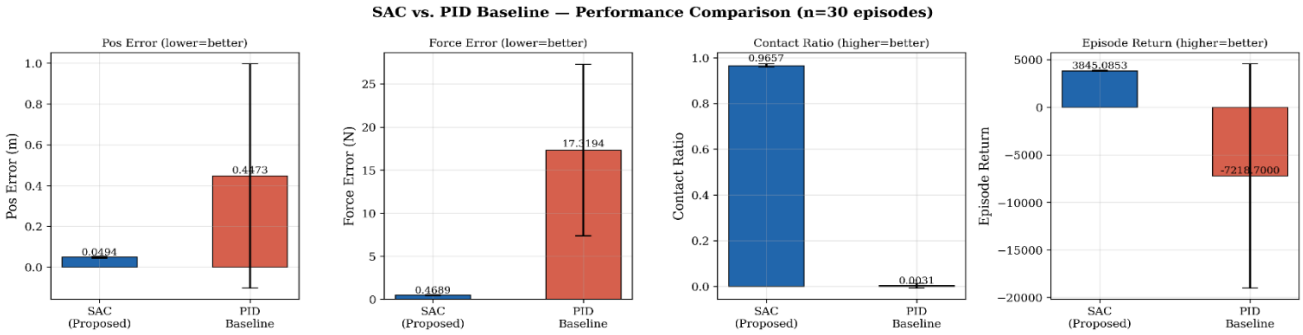

Figure 8. Soft Actor-Critic (SAC) versus Proportional-Integral-Derivative (PID) baseline performance across four metrics

4.9 Pareto front analysis of position-force trade-off

The empirical Pareto front demonstrating the trade-off between mean position error and mean force error as we tune the trade-off weight w_{pos} from 0.1 to 0.9 in increments of 0.1 is shown in Figure 9. The force weight was set to $w_F = 1 - w_{pos}$. Each point on the plot is obtained by fully training a SAC policy and running it for 30 episodes.

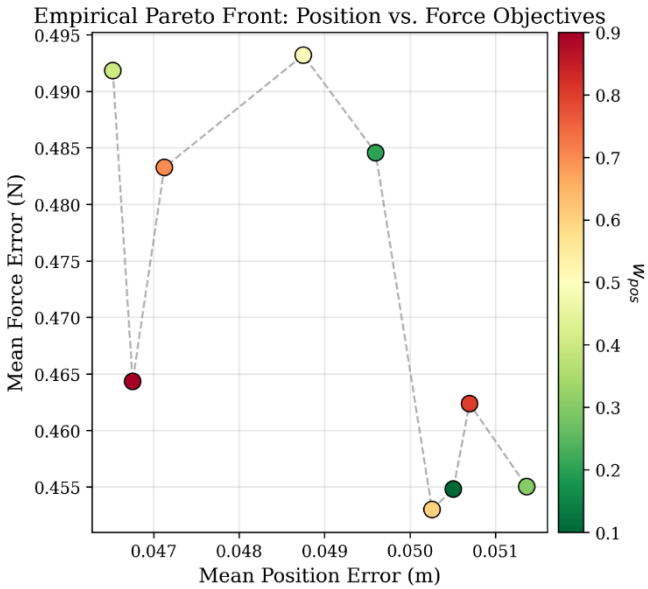

Figure 9. Empirical Pareto front: Position versus force trade-off

The points are also color-coded from deep green at $w_{pos} = 0.1$, to yellow at $w_{pos} = 0.5$, to dark red at $w_{pos} = 0.9$ to easily correlate reward weights to objective-space values. First and foremost, note how both axes are severely truncated: Position errors only range from 0.046–0.052 m, and Force errors only range from 0.453–0.493 N throughout the entire Pareto front. This demonstrates that the SAC algorithm and piece-wise reward function preserves multi-objective capability for nearly every possible weighting choice, instead of completely sacrificing one objective when the weighting is skewed heavily for the other objective. This allows us to change reward priorities on the fly for different deployment needs. From the Pareto plot, we can see trade-offs achieved by controllers with smaller values of w_{pos} clustered around the green area give slightly better force regulation at the expense of slightly worse position tracking, and larger values clustered around the red region allow position error to be driven closer to its minimum of approximately 0.046 m, at the expense of larger force error, up to approximately 0.492 N. The nondeterministic sawtooth pattern linking nearby weight values on the dashed Pareto line is due to the randomness in training and the nonlinear interactions between minimizing the two cost terms via the joint torque command. The equally weighted case of $w_{pos} = 0.5$ denoted by the yellow dot at around (0.050, 0.453 N) is close to the knee of the Pareto plot, which means that further improvements along one metric would require disproportionate improvements along the other metric. Hence, this baseline weight choice is reasonable.

4.10 Comparison with related work

We compare the success of our SAC-based multi-objective controller with other recently proposed solutions to problems involving force-position control and contact-rich manipulation tasks. Zhao et al. [12] achieved convergence to their desired trajectories with their M2ACD approach, but they only focused on planning for collision-free motion, without specifying any contact force objective. Thus the metrics can only be compared in terms of position tracking, where our controller achieves 49.4 mm mean error when bounding the contact force. Maghooli et al. [13] were able to successfully complete position control with a continuum manipulator with nonlinear dynamics, but they were not able to track forces and only worked when in contact-free space. Parnada et al. [14] directly noted these two aspects, lack of staged rewards and contact-state observation as future work. However, we directly incorporate them into our proposed approach. Li et al. [15] were able to learn a force-position coordinated control with a hybrid RL-PID controller, but their solution still relies on model-based methods. Our approach completes this task with purely model-free end-to-end learned policy. Table 1 presents a comparison with related work.

Table 1. Comparison with related work

Method	Position Error	Force Error	Contact Ratio	Model-Free
Zhao et al. [12]	Moderate	N/A	N/A	Partial
Maghooli et al. [13]	Low	N/A	N/A	Yes
Li et al. [15]	Moderate	Moderate	Partial	No
Proportional-Integral-Derivative (PID) Baseline	0.4473 m	17.32 N	0.31%	No
Proposed SAC	0.0494 m	0.4689 N	96.57%	Yes

5. CONCLUSIONS

The work introduced in this paper outlined an end-to-end model-free DRL approach utilizing SAC to learn policies for force setpoint regulation and end-effector position tracking simultaneously on a collaborative arm. The novel contributions of this research can be summarized as follows. Initially, a hierarchical multi-objective reward function was developed to naturally decouple free-space approach and force maintenance by taking advantage of phase-based reward switching, removing the reward function collapse caused by a single scalar reward. Secondly, we showed that the inclusion of a contact-state-aware 9D observation space with explicit wall distance and contact binary flag allowed for faster contact finding and robust switching. Third, with auto entropy tuning applied in our SAC algorithm, we empirically demonstrated temperature annealing can be retired without hurting exploration consistency across three different training seeds. Fourth, extensive empirical study against classical PID baseline across 30 episodes demonstrated position accuracy, force regulation, and contact maintenance ratio improvements of {9.1 x, 36.9 x, 311 x} respectively. Finally, Pareto plot shows our proposed framework achieves stable performance over both objectives when sweeping through position-force weighting factors without retraining the policy. We are interested in extending our work to high-DOF manipulators with different task-specific target forces profiles as well as validating our approach on real collaborative robots to close the sim-to-real gap. All policies were learned completely in simulation with a Hunt-Crossley contact model as a substitute for the real surface. The Hunt-Crossley contact model more accurately reflects nonlinear force transients that occur in real contact than the simple linear spring model. However, it still ignores other key sources of unmodeled behavior present in the real environment such as surface friction, sensor noise, actuator backlash and unmodeled flexibilities. Testing the learned policy on an actual collaborative arm with domain randomization or sim-to-real transfer techniques would be the next step to use it for practical human-robot collaboration applications.

REFERENCES

[1] Adil, A.A., Sakhrieh, S., Mounsef, J., Maalouf, N. (2025). A multi-robot collaborative manipulation framework for dynamic and obstacle-dense environments: Integration of deep learning for real-time task execution. Frontiers in Robotics and AI, 12: 1585544. https://doi.org/10.3389/frobt.2025.1585544

[2] Tang, C., Abbatematteo, B., Hu, J.H., Chandra, R., Martín-Martín, R., Stone, P. (2025). Deep reinforcement learning for robotics: A survey of real-world successes. Annual Review of Control, Robotics, and Autonomous Systems, 8: 153-188. https://doi.org/10.1146/annurev-control-030323-022510

[3] Huang, J.S., Yuan, M.X., Huo, Z.X., Zhang, S.K., Zhang, X.B. (2025). Adaptive robust interaction force control of a robotic manipulator in uncertain environments. IEEE Transactions on Industrial Electronics, 72(8): 8251-8260. https://doi.org/10.1109/TIE.2024.3525103

[4] Hunt, K.H., Crossley, F.R.E. (1975). Coefficient of restitution interpreted as damping in vibroimpact. Journal of Applied Mechanics, 42(2): 440-445. https://doi.org/10.1115/1.3423596

[5] Spong, M.W., Hutchinson, S., Vidyasagar, M. (2020). Robot Modeling and Control. Wiley.

[6] Elguea-Aguinaco, Í., Serrano-Muñoz, A., Chrysostomou, D., Inziarte-Hidalgo, I., Bogh, S., Arana-Arexolaleiba, N. (2023). A review on reinforcement learning for contact-rich robotic manipulation tasks. Robotics and Computer-Integrated Manufacturing, 81: 102517. https://doi.org/10.1016/j.rcim.2022.102517

[7] Jasim Mohamed, M., Oleiwi, B.K., Azar, A.T., Mahlous, A.R. (2024). Hybrid controller with neural network PID/FOPID operations for two-link rigid robot manipulator based on the zebra optimization algorithm. Frontiers in Robotics and AI, 11: 1386968. https://doi.org/10.3389/frobt.2024.1386968

[8] Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., Levine, S. (2021). How to train your robot with deep reinforcement learning: Lessons we have learned. The International Journal of Robotics Research, 40(4-5): 698-721. https://doi.org/10.1177/0278364920987859

[9] Ibrahim, S., Mostafa, M., Jnadi, A., Salloum, H., Osinenko, P. (2024). Comprehensive overview of reward engineering and shaping in advancing reinforcement learning applications. IEEE Access, 12: 175473-175500. https://doi.org/10.1109/ACCESS.2024.3504735

[10] Luo, J., Xu, C., Wu, J., Levine, S.. (2025). Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. Science Robotics, 10: eads5033. https://doi.org/10.1126/scirobotics.ads5033

[11] Kumar , V., Shinde, A.H., Mate, S., Borkar, P., Guravaiah, T., Gadewar, A.B. (2026). Deep reinforcement learning for autonomous robotic manipulation tasks. In 2026 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, pp. 1-6. https://doi.org/10.1109/ESCI68015.2026.11493246

[12] Zhao, B., Wu, Y., Wu, C.D., Sun, R.H. (2025). Deep reinforcement learning trajectory planning for robotic manipulator based on simulation-efficient training. Scientific Reports, 15: 8286. https://doi.org/10.1038/s41598-025-93175-2

[13] Maghooli, N., Mahdizadeh, O., Bajelani, M., Moosavian, S.A.A. (2025). Learning-based control for tendon-driven continuum robotic arms. Frontiers in Robotics and AI, 12: 1488869. https://doi.org/10.3389/frobt.2025.1488869

[14] Parnada, A., Qu, M., Castellani, M., Jin Chang, H., Wang, Y.J. (2026). Towards cost-effective and safe

contact-rich robotic manipulation with reinforcement learning: A review of techniques for future industrial automation. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 240(1): 3-35. https://doi.org/10.1177/09596518251350353

[15] Li, X.P., Guo, B.J., Han, J.H., Zhang, X. (2025). Research on hybrid force/position control method for robot peg-in-hole assembly. Advances in Mechanical Engineering, 17(5). https://doi.org/10.1177/16878132241304254

[16] Li, Y.H., Zheng, L., Wang, Y.H., Dong, E.B., Zhang, S.W. (2025). Impedance learning-based adaptive force tracking for robot on unknown terrains. IEEE Transactions on Robotics, 41: 1404-1420. https://doi.org/10.1109/TRO.2025.3530345

[17] Pan, Y.P., Shi, T., Li, W., Xu, B., Ahn, C.K. (2025). Robot impedance iterative learning with sparse online gaussian process. IEEE/CAA Journal of Automatica Sinica, 12(11): 2218-2227. https://doi.org/10.1109/jas.2025.125195

[18] Ding, Y.F., Zhao, J.C., Min, X.P. (2023). Impedance control and parameter optimization of surface polishing robot based on reinforcement learning. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 237(1-2): 216-228. https://doi.org/10.1177/09544054221100004

[19] Roveda, L., Maskani, J., Franceschi, P., Abdi, A., Braghin, F., Molinari Tosatti, L., Pedrocchi, N. (2020). Model-based reinforcement learning variable impedance control for human-robot collaboration. Journal of Intelligent & Robotic Systems, 100: 417-433. https://doi.org/10.1007/s10846-020-01183-3

[20] Zhou, Z.X., Yang, X.Y., Zhang, X.P. (2025). Variable impedance control on contact-rich manipulation of a collaborative industrial mobile manipulator: An imitation learning approach. Robotics and Computer-Integrated Manufacturing, 92: 102896. https://doi.org/10.1016/j.rcim.2024.102896

[21] Xu, P.J., Li, Z.Y., Liu, X., Zhao, T.R., Zhang, L., Zhao, Y.Z. (2024). Reinforcement learning-based distributed impedance control of robots for compliant operation in tight interaction tasks. Engineering Applications of Artificial Intelligence, 136: 108913. https://doi.org/10.1016/j.engappai.2024.108913

[22] Andrychowicz, O.M., Baker, B., Chociej, M., Józefowicz, R., et al. (2020). Learning dexterous in-hand manipulation. The International Journal of Robotics Research, 39(1): 3-20. https://doi.org/10.1177/0278364919887447