

DOI: 10.37943/CVPA1758

Kanat Suleimenov

PhD, Postdoctoral Researcher, Department of Information Technology and Entrepreneurship

kanat.suleimenov@ut.ee, orcid.org/0000-0002-8062-8855

University of Tartu, Estonia

Relive Research LLP, Kazakhstan

Roza Beisembekova

PhD, Associate Professor, Institute of Automation and Information Technologies

r.beisembekova@satbayev.university, <https://orcid.org/0000-0002-6718-7020>

Satbayev University, Kazakhstan

Aiman Ozhikenova

PhD, Associate Professor, Institute of Automation and Information Technologies

a.ozhikenova@satbayev.university, orcid.org/0000-0001-7078-5737

Satbayev University, Kazakhstan

Assylbek Ozhiken

PhD, Postdoctoral Researcher, Institute of Automation and Information Technologies

a.ozhiken@satbayev.university, orcid.org/0000-0002-2945-8213

Satbayev University, Kazakhstan

Akim Kapsalyamov

PhD, Postdoctoral Researcher, Faculty of Engineering and Mathematics

akim.kapsalyamov@hsbi.de, orcid.org/0000-0003-3700-6214

Bielefeld University of Applied Sciences and Arts, Germany

DEEP REINFORCEMENT LEARNING CONTROL OF A TENDON-DRIVEN UNDERACTUATED FINGER: DETERMINISTIC VS STOCHASTIC

Abstract: Tendon-driven robotic fingers are underactuated and strongly nonlinear, and their unidirectional, friction-attenuated tendon tension imposes actuation constraints that any practical controller must respect. A previous study established that a deterministic learning controller (DDPG) can track a grasp trajectory for such a finger and compared it against model-based schemes. The present work extends that line of investigation by asking a narrower but unresolved question: for this same single-actuator, three-link tendon-driven architecture, how does a deterministic actor-critic policy (DDPG) compare with a stochastic, maximum-entropy policy (SAC) once a tension-limit penalty is built directly into the learning objective? Using the previously validated physics-based simulation model, both agents are trained under identical conditions with a reward that jointly penalizes tracking error and tension-constraint violation, and are evaluated across random, sinusoidal, and step disturbances using steady-state error, RMS, ISE, and IAE. The results show a consistent and interpretable trade-off: the stochastic SAC policy attains lower tracking error and better disturbance rejection, especially under time-varying and abrupt loads, at the cost of noisier control and longer training, whereas the deterministic DDPG policy converges faster and produces smoother torque and tension profiles but adapts less well to coupled disturbances. The comparative analysis has been carried out based on statistical approach, where five different levels of randomness were considered for training of each policy. The study isolates the role of the exploration mechanism and the tension penalty in shaping these behaviors and offers practical guidance on selecting between deterministic and stochastic actor-critic methods for physically constrained, underactuated tendon-driven systems.

Keywords: prosthetic finger; reinforcement learning; underactuated control; tendon-driven systems; actuation constraints; soft actor-critic

Introduction (Literary review)

Underactuated control, especially tendon-driven actuation, is promising in application for prosthetic and assistive hands due to compact, compliant, and lightweight solution mechanism that mimics a load distribution of a human finger [1]. However, tendon-driven actuation makes closed-loop control complicated. Actuation of several joints via a single actuator and force transmission introduces a position-dependent coupling, friction in pulleys, and nonlinear tension dynamics. Since that tendon can only pull, such actuation is usually unidirectional and a controller is not able to command arbitrary torque values for each joint, and it must operate within a constrained and saturated actuation environment [2-3].

Torque commands exceeded tension limits do not only degrade tracking performance, but it also creates risks of mechanical failure and unstable grasping.

Control methods for underactuated robotic fingers have been widely studied and are briefly reviewed here, with detailed discussion provided by Suleimenov et al. [4] PID control offers simplicity but limited performance under nonlinear and time-varying conditions [5]. Feedback linearization (FBL) and its adaptive variants compensate for known nonlinearities but rely strongly on accurate system models [6]. Different studies have refined FBL and showed its application in different fields of robotics, e.g. Alam et al. [7] applied FBL to soft robotics control, Capotondi et al. [8] enhanced FBL by proposing learning-based variants, Malik et al. [9] merged FBL with variable structure control scheme, and Simonini et al. [10] introduced adaptive versions that bypass the need for acceleration measurements. As a model based approach sliding mode control (SMC) became popular due to its robustness to modeling uncertainties [11]. However, initially proposed first-order SMC suffers from “chattering” phenomena and high-frequency oscillations, researchers proposed second-order SMC so called the super-twisting algorithm (STA) which improved robustness to bounded disturbances [12-13]. Dirara et al. [14] further proposed STA utilizing fuzzy logic-based uncertainty estimation approach to minimize trajectory tracking error of two-link manipulator, Lochan et al. [15] presented adaptive STA which demonstrated promising results in suppression of rapid tip deflection, precise trajectory tracking, and chattering-free control compared to other global SMC, Li et al. [16] and Wang et al. [17] have proposed model-free STA schemes based on time-delay estimation (TDE) to enhance precision without complex modeling.

Reinforcement learning (RL) provides a model-free alternative by learning control policies through environmental interaction [18-19]. Elumalai et al. [20] proposed a Collaborative Proximal Policy Optimization (CPPO) reinforcement learning algorithm that incorporates a policy feedback loop between the actor and critic networks to improve value function accuracy and stabilize policy updates in partially observable flexible joint manipulators. Viswanadhapalli et al. [21] formulated a continuous-action reinforcement learning framework using the Deep Deterministic Policy Gradient (DDPG) algorithm combined with Lyapunov-based reward shaping to optimize state-action value estimation and accelerate policy convergence for tracking and vibration control in a rotary flexible link manipulator, Rúbio et al. [22] adapted DDPG reinforcement learning algorithm to train a deep neural network controller in a continuous action space for regulating joint torques in myoelectric robotic finger prostheses, Kapsalyamov et al. [23] developed a model-free reinforcement learning velocity controller utilizing the DDPG algorithm to regulate the continuous motor torque of a Stephenson III six-bar linkage gait rehabilitation exoskeleton, Wu et al. [24] introduced a two-stage transfer reinforcement learning framework using the DDPG algorithm to solve the complex swing-up control of vertical underactuated manipulators with passive joints, He et al. [25] designed an online adaptive reinforcement learning controller based on an actor-critic neural network architecture, where the critic continuously evaluates strategy performance to guide the actor's action policy updates for flexible two-link manipulator trajectory tracking and vibration suppression.

In the previous study by Suleimenov et al. [DDPG outperformed FBL-PD and FBL-STA in trajectory tracking of a three-link tendon-driven finger under disturbances and nonlinearities. However, its deterministic policy relies on exploration noise and may converge to local optima

[26-27]. In contrast, SAC uses a stochastic maximum-entropy policy that promotes exploration [28-29]. Its effectiveness for this constrained underactuated architecture remains to be established.

Relationship to prior work and contributions. This paper extends the work of Suleimenov et al. by comparing deterministic DDPG and stochastic SAC using the same validated model, parameters, trajectory, and evaluation metrics. An explicit tendon-tension penalty is incorporated into the learning objective, enabling analysis of the trade-off between smoothness and convergence speed versus disturbance rejection and exploration. The model and simulation environment are inherited from the work by Suleimenov et al. [4], while the contribution lies in the comparative analysis and constraint-aware objective.

Methods and Materials

Inherited dynamic model of the three-link finger

The plant model is the three-link tendon-driven finger from the work by Suleimenov et al. [4]. Only the essential formulation is restated for completeness, while the full derivation is given in the work by Suleimenov et al. [4]. As shown in Figure 1, three joints are driven by a single tendon, with torsional springs providing passive extension.

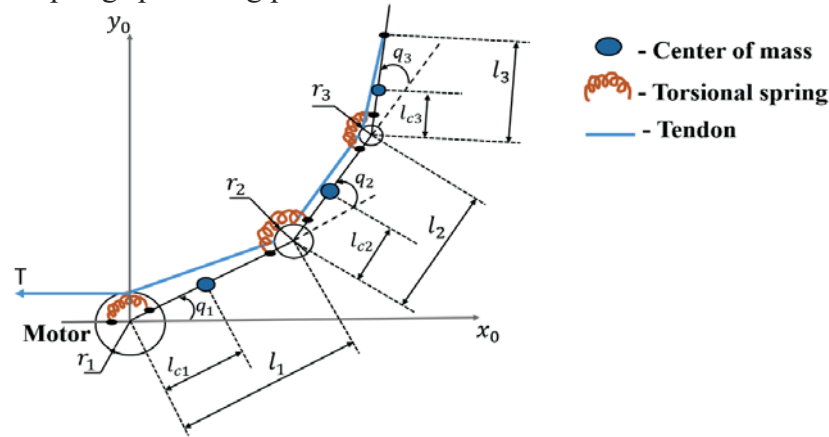


Figure 1. Three-link finger schematic, reproduced from the inherited model from the work by Suleimenov et al. [4].

The symbols r_i , l_i , l_{ci} , q_i denote, respectively, the pulley radius, phalanx length, joint-to-centroid distance, and angular position of a joint, where $i = 1, 2, 3$; T is the tendon tension; counterclockwise rotation is taken as positive. The numerical parameters, identical to those in the work by Suleimenov et al. [4], are listed in Table 1. The equations of motion, derived in the work by Suleimenov et al. [4] via the Euler-Lagrange formulation, take the standard manipulator form

$$S(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + \tau_d \quad (1)$$

where $S(q)$ is a 3-by-3 symmetric positive definite inertia matrix, $C(q, \dot{q})$ collects the Coriolis and centrifugal terms, and $G(q)$ is the gravity/spring vector, τ is the actuation torque, and τ_d is disturbance torque.

Table 1. Parameters of the robotic finger (as in the work by Suleimenov et al. [4])

Robotic finger parameters		
Parameter description	Numerical value	Unit
Link masses, [m_1 m_2 m_3]	[0.0057 0.00529 0.00497]	kg
Link lengths, [l_1 l_2 l_3]	[0.035 0.025 0.015]	m

Pulleys radii, [$r_1 \ r_2 \ r_3$] ^T	[0.015 0.003 0.002] ^T	m
Rotational inertias of the links, [$I_1 \ I_2 \ I_3$] ^T	$10^{-7} * [5.819 \ 2.755 \ 0.932]^T$	kg • m ²
Coefficient of friction between the tendon and the pulley, μ	0.05	—

Tendon transmission and the actuation constraint

Following the transmission model given in the work by Zollo et al. [30], joint torques arise from the tendon tension acting over the pulleys,

$$\begin{aligned}\tau_1 &= r_1 T \\ \tau_2 &= r_2 T \\ \tau_3 &= r_3 T\end{aligned}\tag{2}$$

where the tendon flexes the finger by wrapping around the joint pulleys and the torsional springs provide extension. However, relationship given by equation (2) is applicable when tension is constant along the tendons.

As shown in the study by Suleimenov et al. [4], we do not assume constant tension along the tendon. Accounting for tendon-pulley friction, the tension reaching each joint is attenuated according to

$$\begin{aligned}T_1 &= \frac{\tau_1}{r_1 e^{\mu q_1}} \\ T_2 &= \frac{\tau_1}{r_2 e^{\mu(q_1+q_2)}} \\ T_3 &= \frac{\tau_1}{r_3 e^{\mu(q_1+q_2+q_3)}}\end{aligned}\tag{3}$$

where μ is a friction coefficient. Equation (3) makes the actuation constraint explicit: the mapping from commanded actuator torque to delivered tension is nonlinear and configuration-dependent, the tension is unidirectional, and it cannot exceed a physical limit without risking transmission failure. This constraint, rather than the tracking task alone, motivates the constraint-aware reward introduced in this section later and is the quantity whose handling we use to compare the two policy classes.

Remark 1. The system is driven by a single motor, so only the first joint is directly actuated; the remaining joints respond passively through the tendon.

Problem statement. Given a position tracking error

$$e_1 = q_{1d} - q_1\tag{4}$$

obtain a control law u such that e_1 asymptotically converges to zero, where u represents a torque applied to the motor, q_{1d} is a reference trajectory for the first joint.

Assumption 1. In this study, it is assumed that the finger parameters such as spring stiffness coefficients and pulley radii are selected such that convergence of e_1 to zero ensures that q_2 and q_3 attain values sufficient to produce the desired human-like fingertip trajectory.

Actor-critic learning: deterministic (DDPG) and stochastic (SAC)

Both controllers are model-free, off-policy actor-critic methods that operate in continuous action spaces; the generic machinery (experience replay, target networks, deep function approximation) is standard and detailed in the work by Suleimenov et al. [4]. We highlight only the distinction that matters for this comparison.

DDPG uses a deterministic actor with action-noise exploration, offering efficient learning but remaining sensitive to hyperparameters and Q-value overestimation [31]. SAC employs a stochastic actor with entropy-based exploration and twin critics to improve learning stability as detailed in the studies by Haarnoja et al. [32] and Rusinak et al [33]. The SAC actor (Figure 3) outputs the mean and standard deviation of the action distribution, whereas DDPG (Figure 2) directly outputs the action. Both use ReLU activation in their hidden layers.



Figure 2. Actor network structure of the DDPG: FC(128) - fully connected layer with 128 neurons; FC(200) – fully connected layer with 200 neurons; FC(output) – fully connected layer that produces an output

This study presented results of application of the SAC to the underactuated three-link finger considering limitations in actuator torque and tendon tension force.

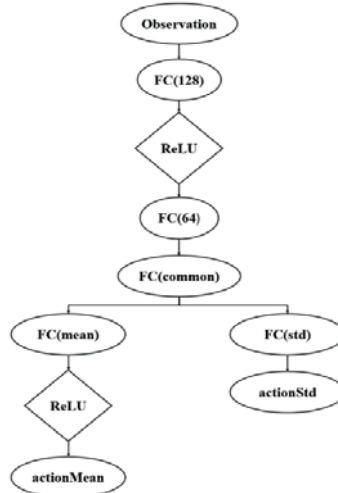


Figure 3. Actor network structure of the SAC: FC(128) - fully connected layer with 128 neurons; FC(64) – fully connected layer with 64 neurons; FC(mean) – fully connected layer that produces a mean value; FC(std) – fully connected layer that produces a standard deviation.

Constraint-aware observation space and reward

The key methodological element of this study is that the actuation constraint is encoded in the learning objective. The observation comprises the first-joint angle, its rate, the tracking error, and the previous actuator torque,

$$o = [q_1 \quad \dot{q}_1 \quad e_1 \quad \tau_{1prev}]^T \quad (5)$$

and the reward couples a tracking term with an explicit tension-violation penalty,

$$reward = -\rho_1 \times |q_{1d} - q_1| - \rho_2 \times |\max(0, T_1 - T_{max})| \quad (6)$$

where T_{max} is the admissible tension limit; $\rho_1 = \rho_2 = 1$ are reward-penalty coefficients. The first term drives tracking; the second activates only when the delivered tension would exceed T_{max} , so the agent is rewarded for staying inside the feasible actuation envelope. A reward approaching zero indicates accurate, constraint-respecting tracking, while large negative values signal divergence. This penalty is identical in form for both agents, which is what allows us to attribute

behavioral differences to the policy class rather than to the objective. The actor and critic architectures are given in Figures 2-5; the only architectural difference relevant here is that DDPG uses a single critic (Figure 4) whereas SAC uses twin critics (Figure 5) and propagates the smaller of the two Q-estimates to limit overestimation.

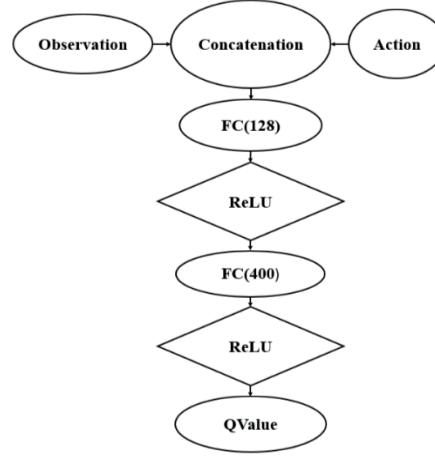


Figure 4. Critic network structure of the DDPG method: FC(128) - fully connected layer with 128 neurons; FC(400) – fully connected layer with 400 neurons.

Remark 2. Joint-angle limits are enforced implicitly through bounded inputs, dissipative dynamics, and the tension penalty in equation (6), rather than being hard-coded. While this allows the policies to learn actuation constraints, hard-limit enforcement using constrained RL or action projection remains as future work.

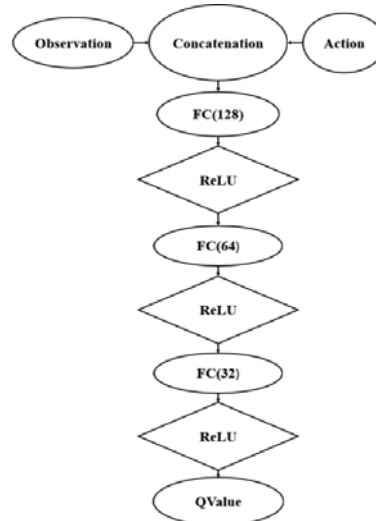


Figure 5. Critic network structure of the SAC method: FC(128) - fully connected layer with 128 neurons; FC(64) – fully connected layer with 64 neurons; FC(32) – fully connected layer with 32 neurons.

Results

The reference is the palmar-grasp trajectory used in the work by Suleimenov et al. [4]

$$q_{ref} = 0.125t^2 - 0.0138889t^3 \quad (7)$$

plotted in Figure 6. Only the first joint is assigned a reference, since it is the only directly actuated joint; the others follow through the tendon-pulley transmission.

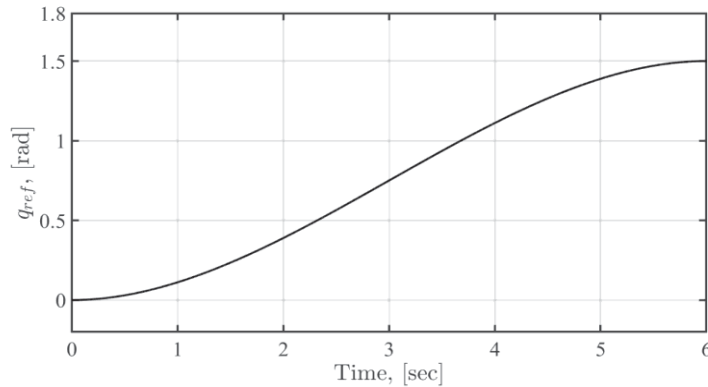


Figure 6. Reference trajectory for the first joint.

Following the work by Suleimenov et al. [4], DDPG and SAC are compared under three first-joint disturbances. Performance is evaluated using steady-state error, RMS, ISE, and IAE over 6 s simulations in MATLAB 2024b/Simulink. Results are shown in Figs. 7-15.

Performance in Scenario 1: bounded random disturbance

Scenario 1 is the most challenging case, with a random disturbance bounded by 0.02 Nm applied to the first joint. As shown in Figures 7-9, both agents exhibit noisy responses, with SAC-RL showing greater fluctuations than DDPG. Despite this, the quantitative results indicate superior tracking performance for the SAC agent, achieving $e_{ss} = 0.0031$ rad, RMS 0.0889 rad, ISE $0.0034 \text{ rad}^2 \times s$, and IAE $0.0123 \text{ rad} \times s$, compared to the DDPG-RL, which yields $e_{ss} = 0.0072$ rad, RMS 0.1125 rad, ISE $0.0093 \text{ rad}^2 \times s$, and IAE $0.0276 \text{ rad} \times s$. The increased noise in the SAC-controlled signals can be attributed to its stochastic policy, which introduces variability in the control actions (torque) and, consequently, the tendon tension, as shown in Figs. 7-9.

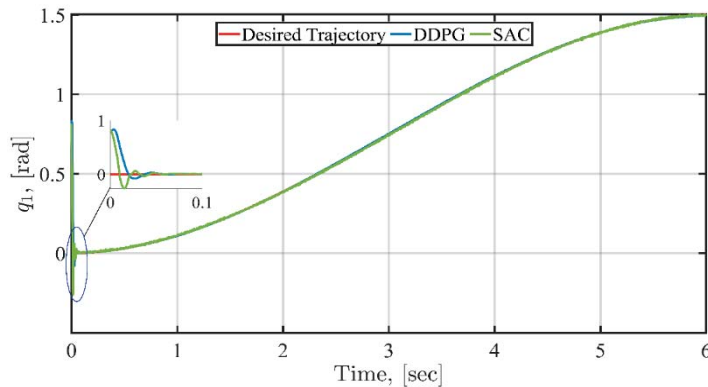


Figure 7. Reference trajectory tracking for the first joint in Scenario 1.

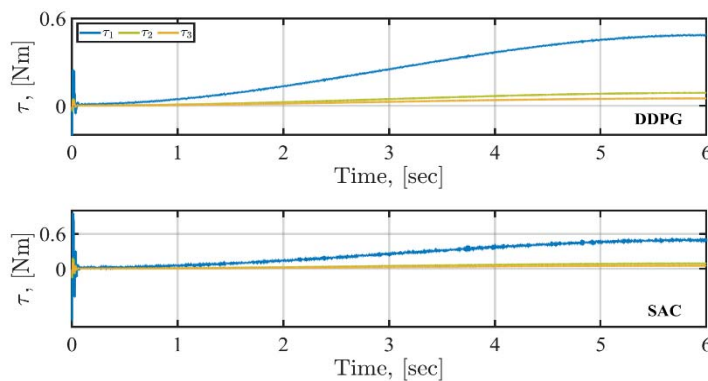


Figure 8. Torque commands in Scenario 1, where $\tau_{\max}=1\text{Nm}$.

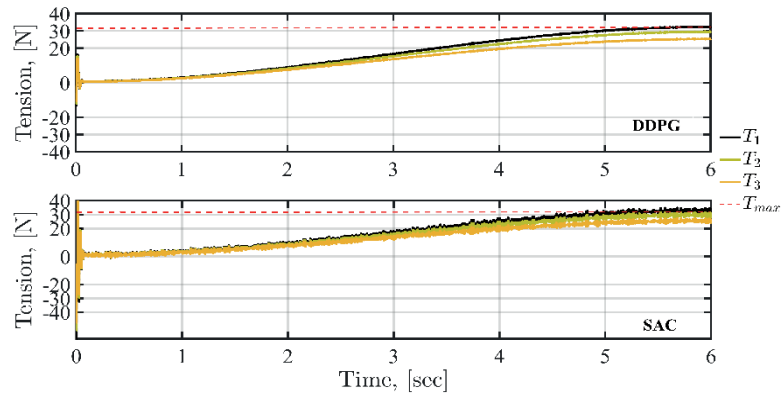


Figure 9. Tension force in each pulley in Scenario 1, where $T_{\max}=31\text{N}$.

Performance in Scenario 2: sinusoidal disturbance

Scenario 2 shows a more pronounced performance difference. The agents were evaluated under 0.02 Nm sinusoidal load torque applied to the first joint.

Due to the coupled dynamics and single actuation mechanism, the disturbance propagates across the joints. As shown in Figures 10–12, DDPG exhibits noticeable tracking oscillations, whereas SAC maintains smoother tracking with superior disturbance rejection.

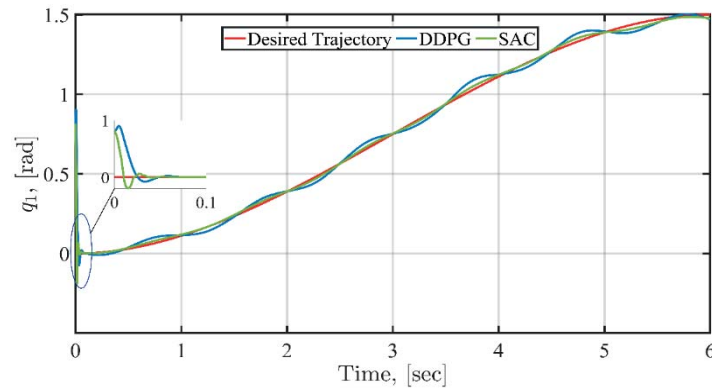


Figure 10. Reference trajectory tracking for the first joint in Scenario 2.

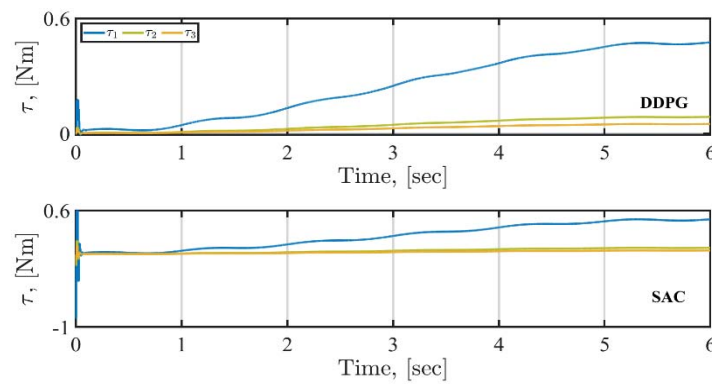


Figure 11. Torque commands in Scenario 2, where $\tau_{\max}=1\text{Nm}$.

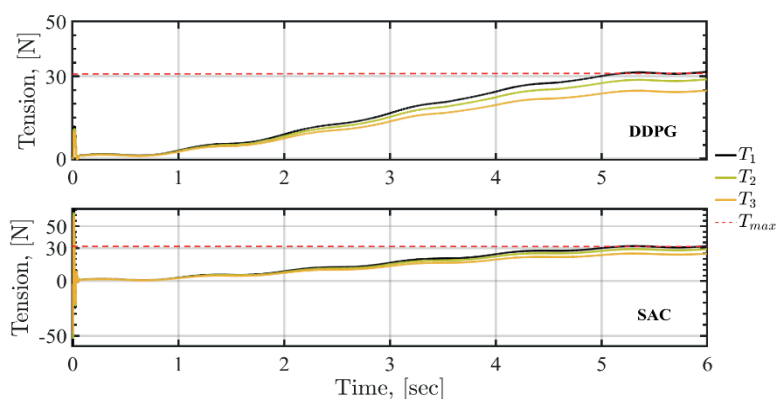


Figure 12. Tension force in each pulley in Scenario 2, where $T_{max}=31N$.

The comparative data of the DDPG and SAC agents are presented in Table 2.

Performance in Scenario 3: step disturbance

Furthermore, the agents trained in Scenario 1 were evaluated under a step disturbance applied to the first joint at $t = 3$ s. The reference tracking performance of both agents is presented in Figure 13. At the initial stage of the simulation, both agents exhibit fluctuations, with the SAC agent showing more pronounced oscillatory behavior. Upon application of the step torque, both controllers experience a degradation in tracking performance, resulting in deviations from the reference trajectory.

Table 2. Performance comparison of DDPG and SAC

Performance criteria	DDPG	SAC
Steady-state error, e_{ss}	0.0072 (Scenario 1), 0.0438 (Scenario 2), 0.0564 (Scenario 3)	0.0031 (Scenario 1), 0.0151 (Scenario 2), 0.0369 (Scenario 3)
Root mean square, RMS	0.1125 (Scenario 1), 0.1167 (Scenario 2), 0.1153 (Scenario 3)	0.0889 (Scenario 1), 0.0895 (Scenario 2), 0.0894 (Scenario 3)
Integral of squared error, ISE	0.0093 (Scenario 1), 0.0134 (Scenario 2), 0.0128 (Scenario 3)	0.0034 (Scenario 1), 0.0042 (Scenario 2), 0.0045 (Scenario 3)
Integral of absolute error, IAE	0.0276 (Scenario 1), 0.1414 (Scenario 2), 0.1195 (Scenario 3),	0.0123 (Scenario 1), 0.0609 (Scenario 2), 0.0635 (Scenario 3)

The quantitative results summarized in Table 2 indicate that the SAC agent slightly outperforms the DDPG in terms of steady-state error, while achieving significantly better performance in terms of RMS, ISE, and IAE, with values of RMS 0.0894, ISE 0.0045, and IAE 0.0635, compared to the DDPG agent with RMS 0.1153, ISE 0.0128, and IAE 0.1195. The initial fluctuations observed in the SAC response can be attributed to its stochastic policy, which introduces variability in the generated control actions during execution.

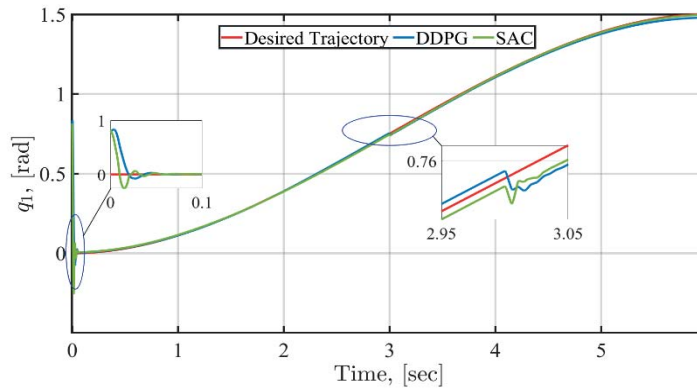


Figure 13. Reference trajectory tracking for the first joint in Scenario 3.

This behavior is evident in the torque and tendon tension signals shown in Figs. 14–15. SAC produces noisier control inputs, particularly at the beginning of the simulation, whereas the deterministic DDPG policy yields smoother torque and more stable tension profiles within acceptable limits.

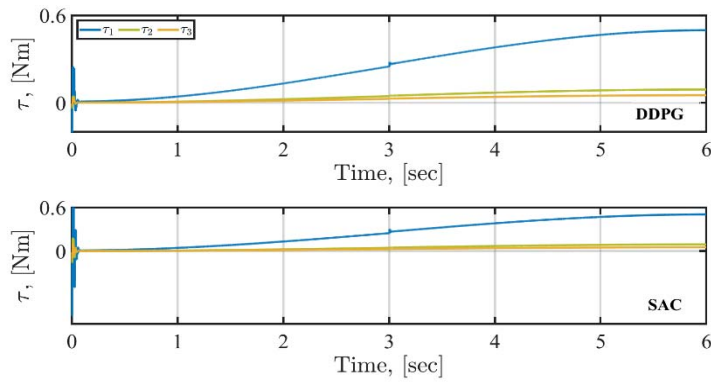


Figure 14. Torque commands in Scenario 3, where $\tau_{\max}=1\text{Nm}$.

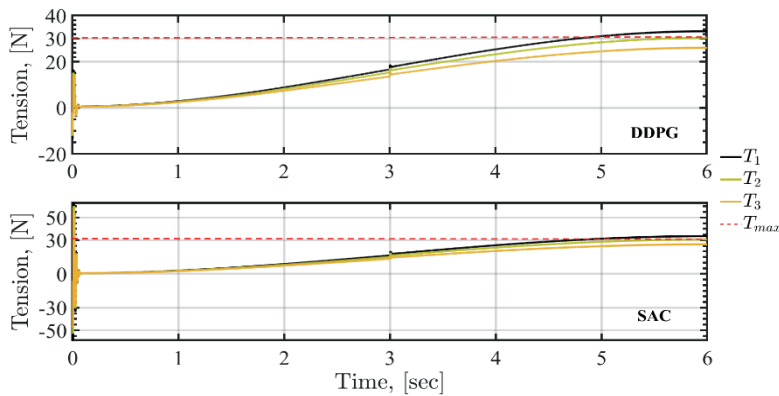


Figure 15. Tension force in each pulley in Scenario 3, where $T_{\max}=31\text{N}$.

Statistical analysis of DDPG and SAC performance

A statistical analysis was conducted under Scenario 1 using five random seeds. Both DDPG and SAC were retrained and evaluated for each seed using the same performance criteria. The results are presented in Table 3.

Table 3. Performance metric data for DDPG and SAC performance trained under five different seeds for Scenario 1.

Seed\Per formanc e metric	Steady- state error DDPG	Steady- state error SAC	DDPG RMS	SAC RMS	DDPG ISE	SAC ISE	DDPG IAE	SAC IAE
0	0.0072	0.0031	0.1125	0.0889	0.0093	0.0034	0.0276	0.0123
1	0.0032	0.0028	0.0969	0.0872	0.0047	0.0033	0.014253	0.009360
2	0.0108	0.0041	0.1012	0.0868	0.0055	0.0033	0.018624	0.016595
3	0.0028	0.0053	0.1029	0.0884	0.0060	0.0035	0.015877	0.014820
4	0.0067	0.0001	0.1062	0.0867	0.0062	0.0032	0.024552	0.010911

A paired *t*-test has been performed to assess statistical differences between DDPG and SAC agents. Table 4 shows mean-standard deviation numbers for each performance metric over five different seeds for DDPG in the second column and for SAC in the third column. The fourth column shows paired differences of the calculated mean values for DDPG and SAC. Last two columns present confidence intervals and t-test p-values for each metric based on the calculated paired differences, respectively. From Table 4, it is shown that reduction of tracking error is statistically significant for SAC in terms of RMS and ISE metrics at a significance level of $\alpha = 0.05$. The confidence intervals for the mean difference of DDPG and SAC were [0.00971, 0.02297] for RMS, and [0.00086, 0.00514] for ISE.

Table 4. Statistical data of performance for five random seeds

Performance metric	$\mu_{DDPG} \pm \sigma_{DDPG}$	$\mu_{SAC} \pm \sigma_{SAC}$	Mean paired difference, $\mu_{DDPG} - \mu_{SAC}$	95% Confidence interval (CI) for paired difference	Paired t-test p-value
Steady-state error	0.00614 $\pm$ 0.00328	0.00308 $\pm$ 0.00193	0.0036	[-0.00194, 0.00806]	0.1645
RMS	0.10394 $\pm$ 0.00584	0.08760 $\pm$ 0.00099	0.01634	[0.00971, 0.02297]	0.0024
IAE	0.02018 $\pm$ 0.00571	0.01280 $\pm$ 0.00292	0.00738	[-0.00087, 0.01564]	0.0679
ISE	0.00634 $\pm$ 0.00175	0.00334 $\pm$ 0.00011	0.00300	[0.00086, 0.00514]	0.0176

where μ_{DDPG} , μ_{SAC} are mean values of corresponding metric across five random seeds for DDPG and SAC agents, respectively; σ_{DDPG} , σ_{SAC} are standard deviation values of corresponding metric across five random seeds for DDPG and SAC agents, respectively.

These two confidence intervals do not include zero, which strongly indicates difference in performance of two agents. However, in terms of IAE metric, SAC did not show significant statistical significance at $\alpha = 0.05$ since the paired t-test p-value (p-value = 0.0679) is greater than the given significance threshold. These observations conclude that SAC showed lower mean IAC across the five different runs, while statistical significance of SAC was shown in terms of RMS and ISE metrics.

Training of the DDPG and SAC agents

Figures 16-17 illustrate the training processes of the DDPG and SAC agents, respectively. The DDPG agent converged after 220 episodes, requiring approximately 30 minutes to complete training. In contrast, the SAC agent required 580 episodes to satisfy the stopping criteria, with a total training time of approximately 1.5 hours. Due to its deterministic policy, the DDPG agent began improving from the early stages of training, with noticeable progress observed within the first 15 episodes.

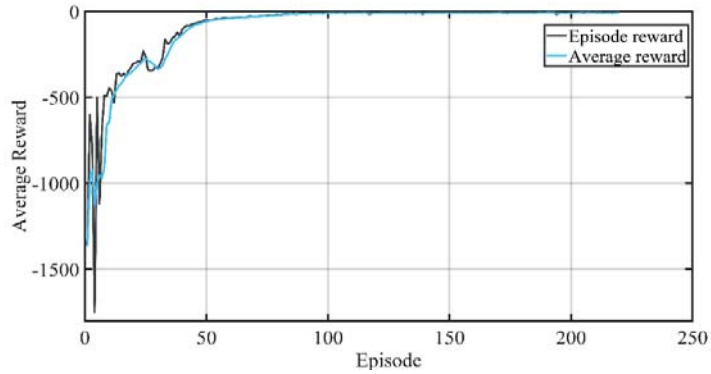


Figure 16. Learning process for DDPG.

It reached performance levels close to the stopping criteria by approximately the 100th episode. Conversely, the SAC agent, characterized by a stochastic policy, exhibited slower initial convergence and did not demonstrate consistent improvement within the first 100 episodes.

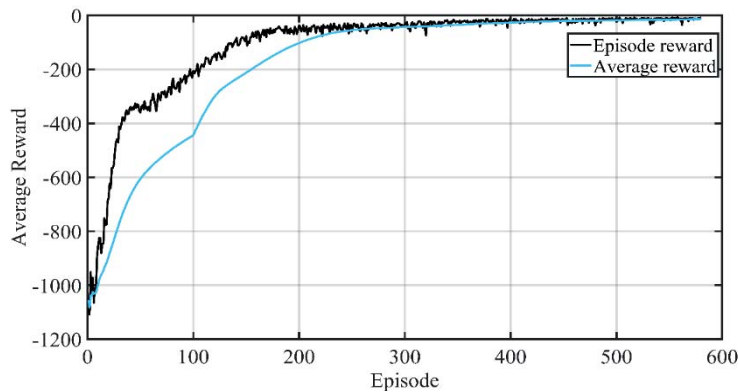


Figure 17. Learning process for SAC.

Its performance began to improve more steadily after approximately the 100th episode and approached the stopping criterion only after around 300 episodes. To regulate the exploration–exploitation trade-off, the SAC agent’s target entropy is set to -3, resulting in a more stochastic policy during training. These results highlight the fundamental differences between the two approaches. While the SAC agent requires a longer training duration due to its stochastic exploration strategy, it provides more stable learning dynamics. In contrast, the DDPG agent converges more quickly but exhibits greater initial fluctuations as it searches for a suitable policy. The training parameters are given in Table 5.

Table 5. Parameters of DDPG and SAC methods

DDPG	Parameter	SAC
0.005 s	Simulation time step	0.005 s
$5e^{-4}$	Actor’s learning rate	$1e^{-4}$
$1e^{-4}$	Critic’s learning rate	$5e^{-4}$
1	Gradient threshold for the actor	1

1	Gradient threshold for the critic	1
	Target smooth factor	0.005
	Target update frequency	1
1	Reward weights	1
$[-1, 1]$	Action bounds	$[-1, 1]$
-5	Stopping criteria is based on evaluation statistic with a threshold	-5
0.3	Exploration noise standard deviation	-
$1e^{-4}$	Exploration noise standard deviation decay rate	-
400	Minibatch size	300
1	Discount factor	1
1200	Maximum steps per episode	1200
2000	Maximum number of episodes	2000
$1e^6$	Experience buffer length	$1e^6$
-	Target entropy	-3

Discussion

Read together with the work by Suleimenov et al. [4], these results complete a two-axis picture of learning-based control for this finger: the prior study placed deterministic RL against model-based control, and the present one places deterministic against stochastic RL under an explicit actuation constraint. Across all three disturbance types, SAC demonstrated better disturbance-rejection performance than DDPG, with the largest improvements observed under the time-varying sinusoidal disturbance and the abrupt step disturbance. This improvement was accompanied by noisier control actions and a longer training time, indicating an empirical trade-off between disturbance rejection, control smoothness, and computational cost. SAC's stochastic entropy-regularized policy may contribute to this behavior; however, because SAC and DDPG also differ in their critic structures, target-update mechanisms, and other algorithmic components, the present comparison cannot attribute the observed improvement exclusively to the exploration mechanism. The results therefore represent a comparison between the complete SAC and DDPG formulations under the evaluated conditions.

The deterministic DDPG policy is the mirror image: faster, smoother, and steadier in its tension profile, but less able to compensate coupled, time-varying disturbances, so its tracking degrades more in disturbance-rich settings. Crucially, both agents learned to keep the delivered tension within its limit under the penalty of (6), which indicates that a soft constraint encoded in the reward is sufficient to shape feasible behavior for both policy classes, while leaving open whether it is sufficient when guarantees are required.

For controller selection in constrained, underactuated tendon-driven fingers, SAC is preferable when disturbance rejection and tracking accuracy are prioritized and some control roughness is acceptable. Conversely, DDPG is attractive when smooth control, predictable tendon tension, and faster, lower-cost training is more important, particularly for reducing mechanical wear.

Conclusion

Building directly on our earlier comparison of model-based and deterministic learning control [4], this paper compared deterministic (DDPG) and stochastic (SAC) actor-critic policies for trajectory tracking of a single-actuator, three-link tendon-driven finger, with the tendon-tension limit encoded explicitly in the learning objective. The central result is an interpretable trade-off: the stochastic policy delivers better disturbance rejection and tracking accuracy at the cost of noisier control and longer training, while the deterministic policy offers smoother, faster, and more

predictable control with weaker adaptability to coupled, time-varying disturbances. Both policies learned to respect the actuation constraint under the proposed reward. These findings give concrete guidance for selecting between deterministic and stochastic actor-critic methods in physically constrained underactuated systems, and they point to constrained-RL formulations with hard feasibility guarantees, together with experimental validation, as the natural continuation of this line of work. This study has several limitations. Tendon constraints are imposed through reward penalties rather than guaranteed, and sensitivity to the penalty weight was not evaluated. The SAC-DDPG comparison does not isolate individual algorithmic components. Statistical validation used only five seeds in one scenario. Future work should include constrained RL, ablations, broader statistical testing, robustness analysis across parameter variations, and hardware validation. Furthermore, this study is limited to robustness validation of each RL method for external disturbances, however, robustness validation against parametric uncertainties would be the next step of the current work.

References

- [1] Zhang, T., Zheng, K., Tao, H., & Liu, J. (2025). A Soft Wearable Modular Assistive Glove Based on Novel Miniature Foldable Pouch Motor Unit. *Advanced Intelligent Systems*, 7(11), 2500274. <https://doi.org/10.1002/aisy.202500274>
- [2] Palli, G. (2007). Model and control of tendon actuated robots [PhD dissertation, Università di Bologna]. https://amsdottorato.unibo.it/id/eprint/341/1/GianlucaPalli_PhDthesis.pdf
- [3] Borghesan, G., Palli, G., & Melchiorri, C. (2010, May). Design of tendon-driven robotic fingers: Modeling and control issues. In *2010 IEEE International conference on robotics and automation* (pp. 793-798). IEEE. <https://doi.org/ezproxy.utlib.ut.ee/10.1109/ROBOT.2010.5509899>
- [4] Suleimenov, K., Kapsalyamov, A., Abdikenov, B., Ozhikenova, A., Igembay, Y., & Ozhikenov, K. (2025). Comparative Analysis of Model-Based and Data-Driven Control for Tendon-Driven Robotic Fingers. *Mathematics*, 13(22), 3669. <https://doi.org/10.3390/math13223669>
- [5] Astrom, K. J. (1995). PID controllers: theory, design, and tuning. The international society of measurement and control. <https://www.wiley.com/en-us/shop/general-introductory-industrial-engineering/pid-controllers-theory-design-and-tuning-2nd-edition-p-9781556175169>
- [6] Isidori, A., van Schuppen, J., Sontag, E., Thoma, M., & Krstic, M. (1995). Communications and control engineering. Nonlinear control systems. <https://doi.org/10.1007/978-1-84628-615-5>
- [7] Alam, U. K., Shedd, K., & Haghshenas-Jaryani, M. (2023). Trajectory control in discrete-time nonlinear coupling dynamics of a soft exo-digit and a human finger using input-output feedback linearization. *Automation*, 4(2), 164-190. <https://doi.org/10.3390/automation4020011>
- [8] Capotondi, M., Turrise, G., Gaz, C., Modugno, V., Oriolo, G., & De Luca, A. (2020). Learning feedback linearization control without torque measurements. In *2020 I-RIM Conference*. <https://doi.org/10.5281/zenodo.4781489>
- [9] Málík, R., Okienková, K., Vasko, D., Kardoš, J., Tárník, M., & Paulusová, J. (2024). Robust Control of a MIMO Laboratory Motion System Using Variable Structure Control and the Computed Torque Method With the Velocity Limitation. *IEEE Access*, 12, 143869-143882. <https://doi.org/10.1109/ACCESS.2024.3471678>
- [10] Simonini, G., Baracca, M., Cavaliere, T. V., Bicchi, A., & Salaris, P. (2025). A novel formulation for adaptive computed torque control enabling low feedback gains in highly dynamical tasks. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2025.3561635>
- [11] Utkin, V. I. (2013). Sliding modes in control and optimization. Springer Science & Business Media.

- [12] Van, M., & Ge, S. S. (2020). Adaptive fuzzy integral sliding-mode control for robust fault-tolerant control of robot manipulators with disturbance observer. *IEEE Transactions on Fuzzy Systems*, 29(5), 1284-1296. <https://doi.org/10.1109/TFUZZ.2020.2973955>
- [13] Shtessel, Y., Edwards, C., Fridman, L., & Levant, A. (2014). Sliding mode control and observation (Vol. 10, pp. 978-970). New York: Birkhäuser. <https://doi.org/10.1007/978-0-8176-4893-0>
- [14] Dirara, H. G., Yareshe, F. T., & Abdissa, C. M. (2025). Design and analysis of adaptive fuzzy super-twisting sliding mode controller for uncertain 2-DOF robotic manipulator. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2025.3581449>
- [15] Lochan, K., Seneviratne, L., & Hussain, I. (2025). Adaptive global super-twisting sliding mode control for trajectory tracking of two-link flexible manipulators. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2025.3557202>
- [16] Li, L., Su, Y., Kong, L., Jiang, K., & Zhou, Y. (2022). TDE-based adaptive super-twisting multivariable fast terminal slide mode control for cable-driven manipulators with safety constraint of error. *IEEE Access*, 11, 6656-6664. <https://doi.org/10.1109/ACCESS.2022.3232555>
- [17] Wang, Y., Yan, F., Chen, J., & Chen, B. (2018). Continuous nonsingular fast terminal sliding mode control of cable-driven manipulators with super-twisting algorithm. *IEEE Access*, 6, 49626-49636. <https://doi.org/10.1109/ACCESS.2018.2868988>
- [18] Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction (Vol. 1, No. 1, pp. 9-11). Cambridge: MIT press. <https://mitpress.mit.edu/9780262193986/reinforcement-learning/>
- [19] Sumiea, E. H., Abdulkadir, S. J., Alhussian, H. S., Al-Selwi, S. M., Alqushaibi, A., Ragab, M. G., & Fati, S. M. (2024). Deep deterministic policy gradient algorithm: A systematic review. *Heliyon*, 10(9). <https://doi.org/10.1016/j.heliyon.2024.e30697>
- [20] Elumalai, V. K. (2025). A proximal policy optimization based deep reinforcement learning framework for tracking control of a flexible robotic manipulator. *Results in Engineering*, 25, 104178. <https://doi.org/10.1016/j.rineng.2025.104178>
- [21] Viswanadhapalli, J. K., Elumalai, V. K., Shah, S., & Mahajan, D. (2024). Deep reinforcement learning with reward shaping for tracking control and vibration suppression of flexible link manipulator. *Applied Soft Computing*, 152, 110756. <https://doi.org/10.1016/j.asoc.2023.110756>
- [22] Rúbio, G. D. P., Costa, M. C. B., & Vimieiro, C. B. S. (2025). A New Proposal for Intelligent Continuous Controller of Robotic Finger Prostheses Using Deep Deterministic Policy Gradient Algorithm Through Simulated Assessments. *Robotics*, 14(4), 49. <https://doi.org/10.3390/robotics14040049>
- [23]. Kapsalyamov, A., Brown, N. A., Goecke, R., Jamwal, P. K., & Hussain, S. (2025). Velocity control of a Stephenson III six-bar linkage-based gait rehabilitation robot using deep reinforcement learning. *Neural computing and applications*, 37(7), 5671-5682. <https://doi.org/10.1007/s00521-024-10944-2>
- [24] Wu, J., Chen, S., Wang, Y., Zhang, P., Su, C. Y., & Sato, D. (2024). A transfer reinforcement learning-based control method for vertical n-link underactuated manipulator with two passive joints. *IEEE/ASME Transactions on Mechatronics*, 30(3), 1797-1806. <https://doi.org.ezproxy.utlib.ut.ee/10.1109/TMECH.2024.3419803>
- [25] He, W., Gao, H., Zhou, C., Yang, C., & Li, Z. (2020). Reinforcement learning control of a flexible two-link manipulator: An experimental investigation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(12), 7326-7336. <https://doi.org.ezproxy.utlib.ut.ee/10.1109/TSMC.2020.2975232>
- [26] Fujimoto, S., Hoof, H., & Meger, D. (2018, July). Addressing function approximation error in actor-critic methods. *In International conference on machine learning (pp. 1587-1596)*. PMLR. <https://doi.org/10.48550/arXiv.1802.09477>

- [27] Aydogmus, O., & Yilmaz, M. (2023). Comparative analysis of reinforcement learning algorithms for bipedal robot locomotion. *IEEE Access*, 12, 7490-7499. <https://doi.org/10.1109/ACCESS.2023.3344393>
- [28] Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018, April). Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1). <https://doi.org/10.1609/aaai.v32i1.11694>
- [29] Ziebart, B. D. (2010). Modeling purposeful adaptive behavior with the principle of maximum causal entropy. Carnegie Mellon University.
- [30] Zollo, L., Roccella, S., Guglielmelli, E., Carrozza, M. C., & Dario, P. (2007). Biomechatronic design and control of an anthropomorphic artificial hand for prosthetic and robotic applications. *IEEE/ASME Transactions On Mechatronics*, 12(4), 418-429. <https://doi.org/10.1109/TMECH.2007.901936>
- [31] Nwachukwu, S. E., Folly, K. A., & Awodele, K. O. (2025). A comparative study between soft actor-critic (SAC) and deep deterministic policy gradient (DDPG) algorithms for solar PV MPPT control under partial shading conditions. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2025.3561807>
- [32] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, July). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning* (pp. 1861-1870). Pmlr. <https://doi.org/10.48550/arXiv.1801.01290>
- [33] Rusinak, D., Kelemen, M., & Virgala, I. (2026). A Systematic Benchmark of Reinforcement Learning Formulations for Tendon-Driven Continuum Robot Control. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2026.3690990>