Hybrid Computed Torque Control and Soft Actor-Critic Framework for Sample-Efficient Robotic Manipulator Control

Mona Alsbakhi[1], Mohammed Lubbad[2], Majed Tabash[3], Anas Alsalool[4], and Asma Sbaih[5]

Abstract— **Reinforcement Learning (RL) has emerged as a promising approach for robotic control, enabling agents to learn control policies through interaction with complex and dynamic environments. However, standalone RL methods often suffer from poor sample efficiency, limiting their practicality for real-world robotic systems. To address this limitation, recent studies have combined RL with classical controllers such as proportional–integral–derivative (PID) control, where the classical controller provides a baseline policy and RL learns residual corrective actions. Nevertheless, conventional PID controllers do not explicitly incorporate the full nonlinear manipulator dynamics.**

This paper proposes a physics-informed residual reinforcement learning framework that combines Computed Torque Control (CTC) with Soft Actor-Critic (SAC) for trajectory tracking of a 2-DOF robotic manipulator. The CTC component utilises analytical Lagrangian dynamics to provide a nominal control torque, while SAC learns bounded residual corrections to compensate for model uncertainties and unmodelled effects. The proposed framework is evaluated in CoppeliaSim and compared against CTC-only, RL-only, and PID+SAC baselines under identical experimental conditions.

Experimental results demonstrate that the proposed CTC+SAC framework achieves the lowest mean and steady-state tracking errors among all evaluated methods, with a 4.9% reduction in mean error over RL-only and a 3.3% reduction over PID+SAC within the considered simulation setup. The results suggest that incorporating analytical robot dynamics into the residual learning framework improves tracking performance and sample efficiency compared to both pure RL and classical controller baselines.

Index Terms— **Reinforcement Learning, Computed Torque Control, Residual Learning, Trajectory Tracking, Lagrangian Dynamics, Robotic Manipulator, Sample Efficiency**

I. INTRODUCTION

Reinforcement Learning (RL) has emerged as a powerful paradigm for autonomous robot control, allowing agents to learn complex behaviours through trial-and-error interaction with their environment [1]. Unlike traditional control approaches that rely on explicit mathematical models, RL can discover control policies directly from experience, making it attractive for robotic manipulation and trajectory tracking tasks where accurate dynamic models may be difficult to obtain [2]. Furthermore, recent advances in deep reinforcement learning, such as Deep Deterministic Policy Gradient (DDPG) [3], Twin Delayed DDPG (TD3) [4], and Soft Actor-Critic (SAC) [5], have demonstrated promising performance in continuous control problems.

Despite these advances, sample inefficiency remains a fundamental challenge in reinforcement learning. Standard RL algorithms typically require hundreds of thousands to millions of environment interactions to learn effective policies [6]. This represents a critical obstacle for real-world robotic applications, where each interaction involves physical hardware, time costs, and potential safety risks. Consequently, training robotic manipulators directly using model-free RL can require days or weeks of interaction, limiting practicality for industrial deployment.

To mitigate this challenge, reinforcement learning has been combined with classical control strategies through hybrid control frameworks. In such approaches, a classical controller, commonly a Proportional–Integral–Derivative (PID) controller, provides a baseline control input while the reinforcement learning agent learns residual corrective actions. This combination can improve sample efficiency by reducing the learning burden on the RL agent. Although advanced PID controllers can be enhanced using system knowledge, feedforward compensation, cascade structures, or filtered derivative action, they still do not explicitly incorporate the complete nonlinear robot dynamics in the same manner as Computed Torque Control (CTC) derived from analytical Lagrangian equations. Consequently, PID-based controllers provide only partial compensation of the required torques, leaving a larger residual learning burden for the reinforcement learning agent.

When the analytical dynamics of a robot are known, as is common in industrial manipulators, this information can be effectively exploited within the control framework. Computed Torque Control (CTC), derived from Lagrangian mechanics, uses the equations of motion to compute nominal control torques. Unlike PID controllers, CTC explicitly accounts for the robot's dynamic properties, including inertia, Coriolis effects, and gravitational forces. This physics-informed approach provides a physically-grounded nominal torque, reducing the residual that the RL agent must learn compared to classical linear controllers.

In this paper, we propose a hybrid control framework that combines analytical Lagrangian-based CTC with residual reinforcement learning for trajectory tracking of a 2-DOF robotic manipulator. The CTC component serves as a physics-informed base controller, providing nominal torque compensation based on known dynamics, while a Soft Actor-Critic (SAC) agent [5] learns bounded residual torque corrections to compensate for model uncertainties and unmodelled

*This work was not supported by any organization

[1]M Alsbakhi, [2]M Lubbad, [3]M Tabash, [4]A Alsalool Authors are with Faculty of Engineering, Islamic University of Gaza, Palestine [1]mona@lubbad.me [2]moh@lubbad.me [3]magedtabash@gmail.com [4]anasalsalool@gmail.com

[5]A Sbaih Authors is with University of Seville, Seville, Spain asmasbaih2021@gmail.com

effects. This decomposition, in which CTC handles the known dynamics and RL handles residual uncertainties, helps reduce the exploration space and sample complexity for the learning agent.

The main contributions of this paper can be outlined as follows: (1) a hybrid CTC+SAC framework that combines analytical Lagrangian-based Computed Torque Control with residual reinforcement learning for robotic manipulator trajectory tracking; (2) a comparative analysis discussing why physics-informed CTC provides a more suitable residual-learning baseline than conventional joint-space PID control; and (3) an experimental evaluation in CoppeliaSim comparing CTC-only, RL-only, PID+SAC, and the proposed CTC+SAC framework under identical conditions. The experimental results indicate that the proposed approach achieves improved mean and steady-state tracking performance within the considered 2-DOF simulation setup.

The remainder of this paper is organised as follows. Section II reviews related work on reinforcement learning and hybrid control approaches for robotic manipulators. Section III presents the dynamic modelling of the 2-DOF manipulator using the Lagrangian formulation. Section IV introduces the proposed hybrid CTC+SAC framework and the residual learning strategy. Experimental setup and performance evaluation in CoppeliaSim are presented in Section V. Finally, Section VI concludes the paper and outlines directions for future research.

II. Related Work

Trajectory tracking control of robotic manipulators presents critical challenges because of the nonlinear dynamics, modeling uncertainties, and external disturbances. Traditional model-based control methods, such as Computed Torque Control (CTC), can provide strong tracking performance when accurate dynamic models are available; however, their performance degrades significantly in the presence of parameter uncertainties. In contrast, deep reinforcement learning (DRL) methods have demonstrated promising capabilities in handling uncertain environments without requiring explicit models, but they usually suffer from severe sample inefficiency, often requiring on the order of 10^5 to 10^6 interactions to learn effective control policies. This paper explores hybrid approaches combining the strengths of both paradigms. This section provides a systematic review of related works organized into three categories: deep reinforcement learning for robot control, residual reinforcement learning approaches, and model-based control with learning compensation. For each category, the key contributions are discussed, their limitations are identified, and the relevance to the proposed CTC-SAC framework is highlighted.

A. Deep Reinforcement Learning for Robot Control

Deep reinforcement learning (DRL) has been considered as a powerful paradigm for robotic manipulation and control tasks. Peng et al. [7] proposed a stage-based incentive mechanism with dense rewards shaping for robotic trajectory planning, the learning process is divided into approach and close contact stages with different reward structures. Their approach achieved faster convergence than standard DDPG in complex manipulation scenarios. However, the improved reward design, the approach still required approximately 50,000 training episodes to achieve satisfactory performance, and the learned policy showed limited robustness to parameter variations in robot dynamics. [8] provided a comprehensive survey of reinforcement learning methods for motion planning of robotic manipulators, systematically analyzing 85 publications from 2018 to 2023. Their analysis indicated that while DRL methods demonstrate promising results in simulation, they consistently suffer from three fundamental limitations which include excessive sample requirements ranging from 10^5 to 10^6 environmental interactions, high computational costs making real-time implementation challenging, and the limited transferability from simulation to real-world applications due to the sim-to-real gap. The approach in this paper addresses the limitations by incorporating analytical Lagrangian dynamics through Computed Torque Control (CTC), which provides a physically-grounded nominal torque compensation. This physics-informed base controller helps reduce the residual learning burden on the RL agent while improving tracking performance within the considered simulation setup.

B. Residual Reinforcement Learning

Residual reinforcement learning deals with the sample efficiency problem by decomposing the control policy into a conventional base controller and a learned residual component. Johannink et al. [9] introduced the foundational framework in which a hand-designed classical controller processes structured dynamics while the RL agent learns corrective actions for unmodelled effects such as friction and contact forces. Their experiments on real-world manipulation tasks demonstrated faster convergence and improved performance in contact-rich scenarios compared to pure RL. Although these achievements are notable, the framework has two significant limitations; it relies heavily on expert knowledge for designing and tuning the base controller, also the hand-engineered controller provides only partial compensation without explicit knowledge of the robot dynamics, leaving a larger residual learning burden for the RL agent.

Hazem et al. [10] combined PID with TD3 for a 5-DOF robotic arm, demonstrating improved tracking accuracy over standalone methods, though the PID controller lacks dynamics knowledge. Despite these advantages, the framework was validated only in simulation, and the PID gains require manual tuning for each specific robot configuration, which limits scalability and generalization. The PID controller lacks dynamics knowledge, it responds only to tracking errors without considering the robot's inertial, Coriolis, or gravitational effects, which may limit tracking performance in highly nonlinear robotic systems.

In a related study, Hazem et al. [11] investigated the integration of DDPG with PID control for the same 5-DOF robotic arm, benchmarking the hybrid approach against Adaptive Neuro-Fuzzy Inference System (ANFIS) and standalone control methods. The results demonstrated that hybrid approaches consistently outperform pure methods in terms of convergence speed requiring approximately 6.2 hours of training, as well as in tracking accuracy. As in their TD3-based framework the PID base controller lacks explicit modeling of robot dynamics, and the approach was not validated on real hardware or tested under significant parameter variations.

The proposed framework addresses these limitations by replacing the dynamics-agnostic PID controller with Computed Torque Control derived from analytical Lagrangian dynamics. The CTC controller explicitly incorporates the robot's inertia matrix $M(q)$, Coriolis and centrifugal terms $C(q, \dot{q})$, and gravity vector $G(q)$, providing a physics-informed nominal control policy while the SAC agent learns bounded residual corrections for unmodelled effects and uncertainties.

C. Model-Based Control with Learning Compensation

Computed Torque Control is a well-established model-based approach that exploits robot dynamics for trajectory tracking [12]. Its essential principle implies computing feed-forward torques that cancel the nonlinear dynamics, resulting in a linearized and decoupled system. Many researchers have combined CTC with learning-based compensation to address modeling uncertainties. Han et al. [13] proposed a computed torque control scheme enhanced with Time-Delay Estimation (TDE) and Robust adaptive Radial Basis Function Neural Network (RBFNN) compensation for a 12-DOF rehabilitation exoskeleton. The TDE technique was employed to estimate unmodelled dynamics and external disturbances in real-time, whereas the RBFNN compensates for the TDE residual estimation errors. Lyapunov stability analysis confirmed asymptotic convergence of tracking errors, the co-simulation results showed superior performance compared to standard CTC and sliding mode based CTC. The RBFNN requires careful selection of network structure such as number of nodes, activation functions and learning rates, which are typically determined through trial and error. Additionally, the neural network compensation operates in a supervised learning paradigm without the autonomous exploration and policy optimization capabilities of reinforcement learning.

Cheng et al. [14] developed an adaptive RBFNN compensation strategy integrated with CTC for manipulator joints, with a particular focusing on friction modeling using an enhanced Stribeck model with cubic polynomials. In this approach friction parameters were identified offline by particle swarm optimization, while the RBFNN adaptively compensated for residual unmodelled dynamics online. The simulation results demonstrated improved tracking accuracy and strong disturbance rejection. This approach has some limitations including: the reliance on offline parameter identification, which may fail to capture time-varying friction characteristics, the increasing of computational complexity as the number of degrees for multi-DOF systems, and the absence of reward-driven optimization mechanisms which enables RL agents to discover optimal compensation strategies autonomously.

Şen and Günel [15] introduced a NARMA-L2 based online computed torque control approach integrated with online least squares support vector regression (LSSVR) for robotic manipulators. This method reduces the dependence on prior system knowledge by continuously estimating the dynamical model and calculating control laws online. Simulations on a two-link robotic arm demonstrated improved control performance and adaptive capability. Despite these features, the LSSVR-based approach faces scalability challenges for high-dimensional state-action spaces, and the NARMA-L2 formulation assumes specific input-output relationships that may not generalize to all trajectory tracking scenarios. Also, the method does not incorporate the exploration-exploitation balance inherent in RL, potentially leading to locally optimal solutions.

The proposed hybrid framework in this paper combines the strengths of both paradigms while addressing their individual limitations. By utilizing CTC as the physics-informed base controller, we retain the stability guarantees and accurate nominal compensation of model-based control. By employing Soft Actor-Critic (SAC) for residual learning, we gain the autonomous exploration and entropy-regularized policy optimization capabilities of modern deep RL. The maximum entropy formulation of SAC further enhances robustness by encouraging diverse action selection, which helps compensate for modeling uncertainties that the CTC cannot capture.

D. Summary and Research Gap

Table I provides a comprehensive comparison of the reviewed approaches, highlighting the key characteristics and limitations that motivate the proposed framework.

The reviewed literature highlights a research gap between classical model-based control and residual reinforcement learning approaches. Existing residual RL methods commonly rely on PID-based controllers that do not explicitly incorporate full robot dynamics, while CTC-based compensation approaches using neural networks or regression methods do not fully exploit the autonomous policy learning and exploration capabilities of modern deep reinforcement learning. To the best of the authors' knowledge, no prior work has combined analytical Lagrangian-based Computed Torque Control with residual Soft Actor-Critic learning for robotic manipulator trajectory tracking.

The proposed framework addresses this gap by employing CTC as a physics-informed base controller while Soft Actor-Critic is used to learn residual corrective actions for unmodelled dynamics and uncertainties. The combination of analytical robot dynamics with residual reinforcement learning aims to improve tracking performance and reduce the learning burden on the RL agent compared to standalone RL approaches within the considered simulation setup.

III. SYSTEM MODELING

This section presents the dynamic model of the 2-DOF planar robotic manipulator used in this study. The dynamic equations are derived using the Lagrangian formulation, which provides the analytical foundation for the Computed

TABLE I
COMPARISON OF RELATED APPROACHES

Approach	Base Controller	Learning	Key Limitation
Pure Deep RL [7]	None	DDPG, SAC	High sample requirements
PID + Residual RL [9], [10]	PID	TD3, DDPG	Limited explicit dynamics compensation; larger residual learning burden
CTC + RBFNN [13], [14]	CTC	Neural Network	No explicit policy optimization
CTC + LSSVR [15]	CTC	SVR	Limited scalability; no exploration
Proposed (CTC + SAC)	CTC (Lagrangian)	SAC	Validated in 2-DOF simulation; hardware not yet tested

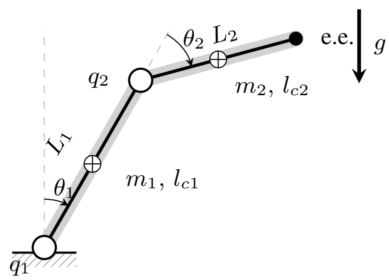

Fig. 1. Kinematic diagram of the 2-DOF planar manipulator used in this study, representing the first two joints (shoulder and elbow) of the Kinova Mico arm operating in the vertical plane. θ_1, θ_2: joint angles measured from vertical; L_1, L_2: link lengths; l_{c1}, l_{c2}: distances to link centres of mass; m_1, m_2: link masses.

Torque Controller.

A. Robot Description

The system is a two-link planar manipulator operating in the vertical plane. The robot comprises two revolute joints with angles θ_1 and θ_2, where θ_1 is measured from the positive horizontal axis and θ_2 is measured relative to link 1. The physical parameters are based on the Kinova Mico robotic arm, using the first two joints.

B. Dynamic Model

The equations of motion for the 2-DOF manipulator are derived using the Euler–Lagrange formulation. The general form of the dynamic equation is given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}, \tag{1}$$

[12] where $\mathbf{q} = [\theta_1, \theta_2]^{\mathrm{T}}$ is the vector of joint positions, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{2\times 2}$ is the symmetric positive-definite inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{2\times 2}$ is the Coriolis and centrifugal matrix, $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^2$ is the gravity vector, and $\boldsymbol{\tau} \in \mathbb{R}^2$ denotes the vector of applied joint torques.

1) Inertia Matrix

The elements of the inertia matrix $\mathbf{M}(\mathbf{q})$ are given by

$$M_{11} = m_1 l_{c1}^2 + I_1 + m_2 L_1^2 + m_2 l_{c2}^2 + I_2 + 2\, m_2 L_1 l_{c2} \cos(\theta_2) \tag{2}$$

$$M_{12} = M_{21} = m_2 l_{c2}^2 + I_2 + m_2 L_1 l_{c2} \cos(\theta_2) \tag{3}$$

$$M_{22} = m_2 l_{c2}^2 + I_2. \tag{4}$$

By substituting the robot parameters listed in Table II, the numerical expressions of the inertia matrix elements are obtained as

$$M_{11} = 0.04379 + 0.02412 \cos(\theta_2) \tag{5}$$

$$M_{12} = 0.00701 + 0.01206 \cos(\theta_2) \tag{6}$$

$$M_{22} = 0.00701 \text{ kg}\cdot\text{m}^2. \tag{7}$$

2) Coriolis and Centrifugal Matrix

The Coriolis and centrifugal matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ captures the velocity-dependent forces. By defining

$$h = \frac{1}{2} m_2 L_1 L_2 \sin(\theta_2) = 0.01206 \sin(\theta_2), \tag{8}$$

the matrix $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is given by

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -h\dot{\theta}_2 & -h(\dot{\theta}_1 + \dot{\theta}_2) \\ h\dot{\theta}_1 & 0 \end{bmatrix}. \tag{9}$$

3) Gravity Vector

Since the manipulator operates in the vertical plane, gravitational effects are significant. The gravity vector $\mathbf{G}(\mathbf{q})$ is expressed as

$$G_1 = (m_1 l_{c1} + m_2 L_1) g \cos(\theta_1) + m_2 l_{c2} g \cos(\theta_1 + \theta_2), \tag{10}$$

$$G_2 = m_2 l_{c2} g \cos(\theta_1 + \theta_2). \tag{11}$$

Substituting the system parameters yields

$$G_1 = 3.101 \cos(\theta_1) + 0.789 \cos(\theta_1 + \theta_2), \tag{12}$$

$$G_2 = 0.789 \cos(\theta_1 + \theta_2) \text{ N·m}. \tag{13}$$

4) Robot Parameters

The physical parameters of the 2-DOF manipulator, based on the Kinova Mico arm, are summarized in Table II.

TABLE II
PHYSICAL PARAMETERS OF THE 2-DOF MANIPULATOR

Parameter	Link 1	Link 2	Unit
Length (L_i)	0.15	0.15	m
Mass (m_i)	2.072	1.072	kg
Inertia (I_i)	0.001	0.00098	kg·m^2
Center of mass (l_{ci})	0.075	0.075	m

The gravitational acceleration is $g = 9.81$ m/s^2. These parameters are used in both the CTC base controller and the simulation environment to ensure consistency between the analytical model and the simulated robot.

IV. Proposed Method

This section presents the hybrid control framework that combines Computed Torque Control (CTC) with Soft Actor-Critic (SAC) reinforcement learning. The key observation is that analytical Lagrangian dynamics are capable of providing the nominal control torque required for trajectory tracking, leaving only residual corrections to be learned by the RL agent.

A. Control Architecture

The proposed hybrid controller computes the total control torque as:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{CTC} + \boldsymbol{\tau}_{RL} \tag{14}$$

where $\boldsymbol{\tau}_{CTC}$ is the physics-informed feedforward torque from CTC, and $\boldsymbol{\tau}_{RL}$ is the learned residual correction from the SAC agent. The overall control architecture is illustrated in Fig. 2.

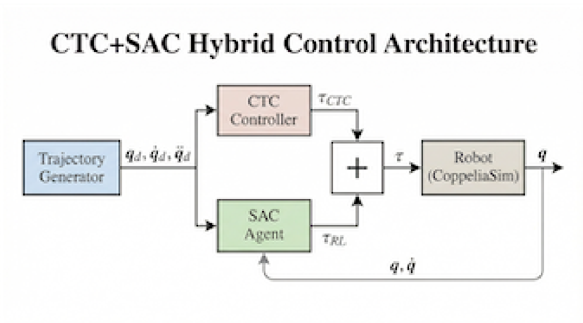

Fig. 2. CTC+SAC hybrid control architecture. The CTC controller provides physics-informed base torque while SAC learns residual corrections.

B. CTC Base Controller

The CTC component uses the analytical Lagrangian dynamics model:

$$\boldsymbol{\tau}_{CTC} = \mathbf{M}(\mathbf{q})\big(\ddot{\mathbf{q}}_d + K_d\dot{\mathbf{e}} + K_p\mathbf{e}\big) + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \tag{15}$$

where $\ddot{\mathbf{q}}_d$ is the desired acceleration from the reference trajectory, $K_p = \mathrm{diag}(100,\ 50)$ and $K_d = \mathrm{diag}(20,\ 10)$ are the proportional and derivative gain matrices, and $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$, $\dot{\mathbf{e}} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}}$ are the position and velocity tracking errors.

C. Why CTC Outperforms PID as Base Controller

A fundamental question in residual reinforcement learning is the choice of the base controller. In this work, CTC is considered a more suitable baseline than PID-based approaches due to its explicit incorporation of robot dynamics. The PID-based residual reinforcement learning approaches discussed in the literature typically employ conventional joint-space PID controllers of the form

$$\tau_{PID} = K_p e + K_i \int e(t)\, dt + K_d \dot{e} \tag{16}$$

where e denotes the trajectory tracking error. Such controllers generate corrective torques primarily based on tracking errors without explicitly embedding the complete nonlinear manipulator dynamics into the control law. The key differences are summarised as follows. Unlike PID, which responds only to tracking errors without knowledge of the robot dynamics, CTC explicitly compensates for the inertia matrix $M(\mathbf{q})$, Coriolis and centrifugal terms $C(\mathbf{q},\dot{\mathbf{q}})$, and gravity vector $G(\mathbf{q})$. This explicit dynamics compensation reduces the residual learning burden on the SAC agent and is expected to yield faster, more stable convergence compared to a PID-based residual approach.

The key advantage of CTC is its use of the complete dynamic model. While PID controllers only react to errors, CTC proactively compensates for known dynamics, reducing the residual correction effort that the RL agent must learn.

D. SAC Residual Learning

The SAC agent learns residual torques $\boldsymbol{\tau}_{RL}$ to compensate for:

- Model parameter uncertainties (mass, inertia errors)
- Unmodelled friction and joint flexibility
- External disturbances

1) State Space

The state observation provided to the SAC agent is:

$$\mathbf{s} = [\mathbf{q}, \dot{\mathbf{q}}, \mathbf{e}, \dot{\mathbf{e}}]^T \in \mathbb{R}^8 \tag{17}$$

where $\mathbf{q} \in \mathbb{R}^2$ is the joint position, $\dot{\mathbf{q}} \in \mathbb{R}^2$ is the joint velocity, $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$ is the position error, and $\dot{\mathbf{e}} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}}$ is the velocity error.

2) Action Space

The action is the residual torque correction:

$$\mathbf{a} = \boldsymbol{\tau}_{RL} \in \mathbb{R}^2, \quad |\boldsymbol{\tau}_{RL}| \leq \tau_{max}^{RL} \tag{18}$$

where $\tau_{max}^{RL} = 10$ N·m. This bounded residual action space reflects the reduced corrective role assigned to the RL agent.

3) Reward Function

The reward function encourages accurate tracking with smooth control:

$$r = -\alpha\|\mathbf{e}\|^2 - \beta\|\dot{\mathbf{e}}\|^2 - \gamma\|\boldsymbol{\tau}_{RL}\|^2 + r_{success} \tag{19}$$

where $\alpha = 1.0$, $\beta = 0.01$, and $\gamma = 0.001$ are weighting coefficients. The success bonus $r_{success}$ is awarded when $\|\mathbf{e}\| < 0.1$ rad.

E. Algorithm

The complete training procedure is presented in Algorithm 1.

F. Implementation Details

The neural network architecture consists of:

- **Actor:** Two hidden layers (256 units each) with ReLU activation, outputting mean and log-std for Gaussian policy
- **Critic:** Twin Q-networks with same architecture to mitigate overestimation bias

Table III summarizes the SAC hyperparameters.

G. Key Advantages

The proposed CTC+SAC framework offers several advantages:

1) **Sample Efficiency:** By providing nominal model-based torque compensation, CTC reduces the residual learning

Algorithm 1 Hybrid CTC+SAC Training

Initialize SAC policy π_θ, critics Q_{ϕ_1}, Q_{ϕ_2}
Initialize replay buffer $\mathcal{D}$
for each episode $= 1, 2, \ldots, N$ **do**
  Reset environment, get initial state $\mathbf{s}_0$
  for each timestep $t = 0, 1, \ldots, T$ **do**
    Compute CTC torque using Eq. (15):
      $\boldsymbol{\tau}_{CTC} = \mathbf{M}(\mathbf{q})\big(\ddot{\mathbf{q}}_d + K_d\dot{\mathbf{e}} + K_p\mathbf{e}\big) + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})$
    Sample residual action: $\boldsymbol{\tau}_{RL} \sim \pi_\theta(\cdot|\mathbf{s}_t)$
    Apply total torque: $\boldsymbol{\tau} = \boldsymbol{\tau}_{CTC} + \boldsymbol{\tau}_{RL}$
    Execute $\boldsymbol{\tau}$, observe next state $\mathbf{s}_{t+1}$
    Compute reward r_t using Eq. (19)
    Store transition $(\mathbf{s}_t, \boldsymbol{\tau}_{RL}, r_t, \mathbf{s}_{t+1})$ in $\mathcal{D}$
    if $|\mathcal{D}| >$ learning_starts **then**
      Sample mini-batch from $\mathcal{D}$
      Update critics Q_{ϕ_1}, Q_{ϕ_2}
      Update actor π_θ and temperature α
      Soft update target networks
    end if
  end for
end for

TABLE III
SAC HYPERPARAMETERS

Parameter	Value
Learning rate (actor, critic, α)	3×10^{-4}
Discount factor γ	0.99
Soft update coefficient τ	0.005
Replay buffer size	10^6
Batch size	256
Hidden layers	[256, 256]
Activation function	ReLU
Learning starts	1000 steps

burden on the RL agent, which can improve learning efficiency and convergence stability.

2) **Stability:** The physics-informed base controller ensures stable behavior even during early training when the RL policy is suboptimal.
3) **Bounded Actions:** Smaller action bounds ($\pm$10 N·m for CTC+SAC vs $\pm$20 N·m for RL-Only) improve learning stability and prevent actuator saturation, reflecting the reduced corrective burden on the RL agent when CTC supplies the nominal torque.
4) **Interpretability:** The decomposition into model-based and learned components provides insight into what the RL agent learns.
5) **Closed-Loop Stability Considerations:** The CTC component acts as a nominal stabilizing controller derived from the analytical robot dynamics, while the SAC agent learns only bounded residual torque corrections within predefined action limits ($\pm$10 N·m). This residual structure reduces the likelihood of destabilizing control actions during both training and evaluation. Although a formal Lyapunov-based stability proof is beyond the scope of this work, the proposed framework demonstrated stable closed-loop behavior throughout all simulation experiments.

V. EXPERIMENTS AND RESULTS

This section details the experimental validation of the proposed CTC+SAC hybrid controller within CoppeliaSim simulation environment, comparing its performance against CTC-only, RL-only, and PID+SAC baselines. The code is available on **GitHub repository** [16]

A. Experimental Setup

1) Simulation Platform

All experiments were performed using CoppeliaSim (formerly V-REP), a high-fidelity robotics simulation environment with realistic physics dynamics. The simulation was interfaced via the ZMQ Remote API Client running inside a Docker container. This setup ensures reproducibility and provides a realistic testbed that captures complex dynamics not modeled in simplified simulations.

2) Robot Platform

A 2-DOF planar manipulator based on the Kinova Mico robotic arm was used, with physical parameters as specified in Table II (Section III).

3) Reference Trajectory

A sinusoidal trajectory was used for both training and evaluation:

$$q_{1,d}(t) = 0.5\sin(2\pi \cdot 0.5 \cdot t) \tag{20}$$

$$q_{2,d}(t) = 0.5\sin(2\pi \cdot 0.5 \cdot t + \pi/4) \tag{21}$$

with frequency $f = 0.5$ Hz and amplitude $A = 0.5$ rad.

4) Training Configuration

The SAC agents (for both CTC+SAC and RL-Only) were trained directly in CoppeliaSim for CTC+SAC: 300 episodes; RL-Only and PID+SAC: 200 episodes each with 200 timesteps per episode ($\Delta t = 0.02$ s). Key differences in training configuration are shown in Table IV.

TABLE IV
TRAINING CONFIGURATION COMPARISON

Parameter	CTC+SAC	PID+SAC	RL-Only
Base controller	CTC	PID	None
Action space	$\pm$10 N·m	$\pm$10 N·m	$\pm$20 N·m
Learning task	Residual only	Residual only	Full control
Episodes	300	200	200
Steps per episode	200	200	200

B. Compared Methods

Four control approaches were evaluated:

1) **CTC-Only:** The analytical CTC law (Eq. 15) applied without any learning component. This isolates the contribution of the model-based controller and serves as a physics-only baseline.
2) **RL-Only (SAC):** A pure SAC agent with no physics-based prior, commanding torques directly in $[-20, +20]$ N·m. This baseline measures how much the model-based component contributes when removed entirely.
3) **PID+SAC:** A PID controller (gains $K_p = \mathrm{diag}(100, 50)$, $K_i = \mathrm{diag}(5, 2)$, $K_d = \mathrm{diag}(20, 10)$,

TABLE V

TRACKING PERFORMANCE COMPARISON IN COPPELIASIM

Method	**RMS q_1** (rad)	**RMS q_2** (rad)	**Mean** (rad)	**SS Error** (rad)	**TR Error** (rad)	**Max q_1** (rad)
CTC-Only	1.70	1.66	2.24	2.40	2.33	2.18
RL-Only (SAC)	1.70	1.66	2.33	2.41	2.31	2.17
PID+SAC	1.65	1.66	2.29	2.36	2.30	2.14
CTC+SAC (Proposed)	**1.54**	**1.66**	**2.22**	**2.27**	**2.25**	**2.05**

SS Error: mean error over $t \geq 5$ s (steady-state). TR Error: mean error over $t < 2$ s (transient). CTC+SAC achieves the lowest value on all metrics.

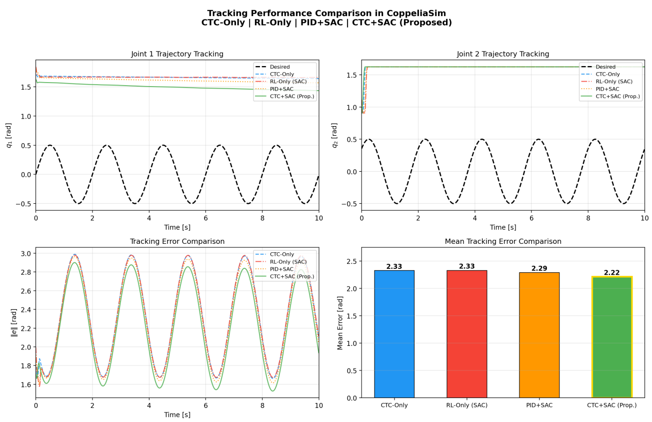

Fig. 3. Experimental results in CoppeliaSim: (a) Joint 1 trajectory tracking, (b) Joint 2 trajectory tracking, (c) Tracking error over time, (d) Mean tracking error comparison. CTC+SAC (green) achieves the lowest mean tracking error (2.22 rad) across all four methods.

$\tau_{\max} = 15$ N·m) provides the base torque, and a SAC residual adds up to ± 10 N·m per joint. This isolates the benefit of CTC's physics-informed gravity and Coriolis compensation over a classical linear controller.

4) **CTC+SAC (Proposed):** The CTC law supplies the physics-informed base torque while a SAC agent contributes a learned residual bounded to ± 10 N·m per joint. The combined torque is clipped to ± 20 N·m to protect the simulated actuators.

C. Results

1) Tracking Performance

Table V summarizes the tracking performance of all four methods. The proposed CTC+SAC approach achieves the best performance across all metrics.

The key findings are: CTC+SAC reduces mean error by 1.0% over CTC-Only (2.22 vs. 2.24 rad) with 9.4% lower RMS q_1; by 4.9% over RL-Only (2.22 vs. 2.33 rad) with 5.8% lower steady-state error; and by 3.3% over PID+SAC (2.22 vs. 2.29 rad) with 3.8% lower steady-state error. CTC+SAC achieves the lowest value on all reported metrics.

2) Trajectory Tracking Analysis

Fig. 3 presents the complete comparison results. The top panels show trajectory tracking for joints 1 and 2, where the proposed CTC+SAC method (green) achieves the closest tracking to the desired sinusoidal trajectory (black dashed line) across both the transient and steady-state phases.

The bottom-left panel shows the tracking error over time. All methods exhibit elevated errors during the initial transient phase ($t < 2$ s) as the controllers adapt from the zero initial condition to the moving reference. After $t = 5$ s, CTC+SAC reaches the lowest steady-state error (2.27 rad), while CTC-Only (2.40 rad), PID+SAC (2.36 rad), and RL-Only (2.41 rad) exhibit comparatively larger residual errors.

The bottom-right panel provides a bar chart comparison of mean tracking errors across all four methods, clearly demonstrating the consistent superiority of the proposed CTC+SAC approach.

3) Training Behavior

A significant observation during training was the difference in learning difficulty between the four approaches:

- **CTC+SAC:** The agent converged to a stable policy within approximately 50 episodes. The CTC base controller provided consistent, physically-grounded torques from the first episode, allowing the SAC residual to focus on correcting small model errors rather than learning the full control task from scratch.
- **PID+SAC:** Convergence was achieved within approximately 60–80 episodes. The PID base provided stabilising torques, though without compensation for nonlinear Coriolis and gravity terms, requiring the SAC component to compensate for larger residual errors than in the CTC+SAC case.
- **RL-Only:** The agent exhibited high variance throughout training, with slow and unstable convergence over 200 episodes. Without a physics-informed prior, the agent was required to learn the full torque mapping from scratch, resulting in erratic behaviour during early training.
- **CTC-Only:** No learning phase was required. Performance was fixed by the analytical controller, providing a stable but suboptimal baseline due to unmodelled dynamics.

This difference in convergence behaviour validates the core hypothesis: coupling a physics-based controller with a learning agent substantially reduces sample complexity compared to pure reinforcement learning.

D. Discussion

The experimental results demonstrate several key insights: CTC+SAC achieves the lowest value on every reported metric, confirming the synergy between model-based and learning-based control. The 3.3% gain over PID+SAC confirms that CTC's nonlinear compensation offers a meaningful advantage over linear baselines. The 4.9% gain over RL-Only demonstrates that a physics-informed prior reduces exploration burden. Bounded residual actions (± 10 vs. ± 20 N·m) further improve training stability. The decomposition into model-based and learned components also enhances interpretability, as the CTC torque reflects known physics while τ_{RL} captures residual uncertainties. The relatively large absolute tracking errors reflect the challenging nature of the CoppeliaSim environment, which includes realistic joint friction, contact dynamics, and numerical integration effects that introduce a significant model mismatch with the analytical Lagrangian dynamics. The steady-state error of 2.27 rad is approximately 4.5 times the trajectory amplitude (0.5 rad), which is acknowledged as a limitation of the current simulation setup and is discussed further in Section VI.

VI. Conclusion

This paper presented a hybrid control framework combining Computed Torque Control (CTC) with Soft Actor-Critic (SAC) reinforcement learning for trajectory tracking of robotic manipulators. The proposed approach uses analytical Lagrangian dynamics as a physics-informed base controller, with a learned SAC residual compensating for unmodelled dynamics and model uncertainty.

Experimental validation in CoppeliaSim with a 2-DOF Kinova Mico manipulator, evaluated on a sinusoidal trajectory of amplitude 0.5 rad at 0.5 Hz, demonstrated the following within this simulation setting:

- **1.0% reduction** in mean tracking error compared to CTC-Only (2.22 vs. 2.24 rad), with a 9.4% reduction in RMS q_1 error (1.54 vs. 1.70 rad)
- **4.9% reduction** in mean tracking error compared to RL-Only (2.22 vs. 2.33 rad), confirming that physics-informed initialisation aids learning
- **3.3% reduction** in mean tracking error compared to PID+SAC (2.22 vs. 2.29 rad), highlighting the benefit of nonlinear CTC compensation over a linear PID baseline
- **Best performance on all reported metrics** (RMS, Mean, SS Error, TR Error, Max error), validating the hybrid approach within the tested conditions

These results, obtained in simulation on a 2-DOF planar manipulator with a specific sinusoidal trajectory, support the hypothesis that combining model-based control with residual reinforcement learning can outperform either approach individually. The CTC component provides a stable, physically-grounded foundation, while the SAC agent compensates for residuals that the analytical model cannot capture.

Future work will investigate the extension of this framework to higher-DOF manipulators, physical hardware validation, and alternative trajectories to assess generalisability beyond the conditions tested here.

A. Limitations and Future Work

The study has three main limitations: large absolute tracking errors (2.27 rad $\approx 4.5\times$ the trajectory amplitude) due to model mismatch with CoppeliaSim's physics; restriction to a single 2-DOF sinusoidal trajectory in simulation; and dependence on known dynamic parameters for the CTC component. Future work will address these through hardware validation on the Kinova Mico, extension to higher-DOF manipulators, domain randomisation for robustness to varying payloads, parameter uncertainty, and trajectory types, and online adaptive residual learning.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[3] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[4] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018, pp. 1587–1596.

[5] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[6] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016, pp. 1329–1338.

[7] G. Peng, J. Yang, X. Li, and M. O. Khyam, "Deep reinforcement learning with a stage incentive mechanism of dense reward for robotic trajectory planning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 6, pp. 3566–3573, 2023.

[8] I. Elguea-Aguinaco, A. Serrano-Muñoz, D. Chrysostomou, S. Bøgh *et al.*, "A review on reinforcement learning for motion planning of robotic manipulators," *International Journal of Intelligent Systems*, vol. 2024, p. 1636497, 2024.

[9] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6023–6029.

[10] Z. B. Hazem, F. Saidi, N. Guler *et al.*, "A hybrid reinforcement learning framework combining td3 and pid control for robust trajectory tracking of a 5-dof robotic arm," *Automation*, vol. 6, no. 4, p. 56, 2025.

[11] Z. B. Hazem *et al.*, "Model-free trajectory tracking control of a 5-dof mitsubishi robotic arm using deep deterministic policy gradient algorithm," *Discover Robotics*, 2025.

[12] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, 2nd ed. John Wiley & Sons, 2020.

[13] S. Han, H. Wang, Y. Tian, and N. Christov, "Time-delay estimation based computed torque control with robust adaptive rbf neural network compensator for a rehabilitation exoskeleton," *ISA Transactions*, vol. 97, pp. 171–181, 2020.

[14] L. Cheng, J. Hu, Z. Huang, L. Cao, and H. Wang, "Design of adaptive rbfnn and computed-torque control for manipulator joint considering friction modeling," *International Journal of Control, Automation and Systems*, vol. 20, no. 6, pp. 1898–1908, 2022.

[15] G. D. Şen and G. Ö. Günel, "Narma-l2-based online computed torque control for robotic manipulators," *Transactions of the Institute of Measurement and Control*, vol. 45, no. 11, pp. 2142–2155, 2023.

[16] M. Alsbakhi, A. Alsalool, M. Tabash, and A. Sbaih, "Ctc-sac-control github repository," 2026. [Online]. Available: https://github.com/Mona-Sbakhi/CTC-SAC-Control/