Reinforcement Learning-based Response Shaping Control of Dynamical Systems

Chepuri Shivani
Robotics Research Center
International Institute of Information Technology
Hyderabad, India
shivani.chepuri@research.iiit.ac.in

Harikumar Kandath
Robotics Research Center
International Institute of Information Technology
Hyderabad, India
harikumar.k@iiit.ac.in

2023 11th International Conference on Control, Mechatronics and Automation (ICCMA) | 979-8-3503-1568-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCMA59762.2023.10374645

***Abstract*—The control system design specifications for dynamical systems are typically provided in terms of the desired transient response and steady-state response. Meeting such requirements for dynamical systems whose mathematical models are unavailable is a challenging task. In this paper, we propose a learning-based controller to achieve the desired control system design specifications for unknown dynamical systems. We consider a SoTA model-free reinforcement learning agent (TD3) in the continuous state and action space setting where the agent has no knowledge of system dynamics. The selection of an appropriate reward function is a key factor that shapes the response of the dynamical system when the learning-based controller is implemented. The resulting controller is trained and tested for first-order and second-order linear systems, as well as a nonlinear system. The resulting trajectories of the closed-loop systems indicate that the transient and steady-state response can be altered by choosing the appropriate reward function while adhering to the constraints imposed on the control input.**

***Index Terms*—model-free reinforcement learning, dynamical systems, error dynamics, reward shaping, trajectory tracking control**

I. Introduction

Control system design specifications are often specified in the time domain or alternatively in the frequency domain. Most of the traditional controllers such as PID controllers, lag-lead compensators, etc. are designed based on linearized models of the dynamical system around an operating point. The parameters of such controllers are tuned to yield a desired response specified in terms of the maximum overshoot, rise time, settling time and steady-state tracking error (or equivalent gain margin, phase margin specifications in the frequency domain). A similar approach for meeting the design specifications is extended to state space methods-based design using techniques like pole placement. Optimization-based methods like LQR (linear quadratic regulator) and MPC (model predictive control) try to achieve the desired response by choosing an appropriate cost function that is minimized by the control input. The above-mentioned controllers require a high-fidelity model of the dynamical system to be controlled. Robust and adaptive controllers can yield satisfactory performance in certain cases when designed based on approximate system models. The assumptions made on the uncertainty need to be accurate for such robust and adaptive control techniques to perform well in the real-world scenarios. Data-driven approach to control system design alleviates the need for a precise mathematical model of the dynamical system to be controlled [1]. The control system design process can either rely on the constructed data-driven model or be completely model free.

Reinforcement Learning (RL), has been largely used to solve many control problems in a variety of domains such as robotics, industrial processes, gaming, human-computer interaction, etc [4], [5]. Some data-driven tracking control methods are able to achieve performance comparable to the state-of-the-art model-based controller. Model-free reinforcement learning approaches can learn to control a system in an unknown environment without the need for the underlying mathematical model, allowing a wide area of application in linear and nonlinear dynamical systems. By combining Deep Learning (DL) and RL, the methods based on Deep Reinforcement Learning (DRL) are applied to control many dynamical systems [6], [7]. In [6], the authors present a composite model-based controller in the inner loop and a reinforcement learning-based controller in the outer loop to control the transient response with an application to microgrids. The RL agent provides the set point for the inner loop model-based controller to track. In [7], a DDPG-PPO-based RL controller is implemented to control a rotary inverted pendulum in real-time. The peak overshoot value of the output is taken as the performance measure. And in [8], a control approach for minimizing tracking error using Integral Reinforcement Learning (IRL) is presented. This optimal control method is implemented on a system with unknown drift dynamics but known input dynamics. In [10] IRL is used for exponential optimal trajectory control for a completely unknown linear system. However, to the best of the authors' knowledge, controlling the evolution of error dynamics in a completely unknown dynamical system (linear or nonlinear) while maintaining control input constraints using RL is seldom explored.

In this paper, we follow a model-free RL-based approach for controller design in the continuous state-action space setting, where the agent has no knowledge of the system dynamics. The twin-delayed deep deterministic policy gradient algorithm (TD3) is a state-of-the-art algorithm that

has demonstrated its performance in this domain [2], [9]. A TD3 agent is trained to shape the evolution of the error signal with respect to approximate desired error dynamics.

The key contribution of this paper lies in shaping the error dynamics for systems with completely unknown dynamics using RL-based control. This method may seem counter-intuitive to the fact that error dynamics depend on the system dynamics, and forcing a trajectory may lead to invalid system states and behavior. To tackle this problem, we introduce this new reward-tuning-based reinforcement learning. The proposed method shapes the transient response by tuning the parameters in the agent's reward function, thus avoiding invalid system states and behavior. We present this approach for controlling the error dynamics of linear first and second-order systems as well as a non-linear system (pendulum). The results indicate that by tuning the reward function, the desired transient and system response for error dynamics can be achieved subjected to control input constraints.

This paper is organized as follows. The preliminaries on TD3 algorithm and the trajectory tracking problem are given in section II. The methodology followed to solve the trajectory tracking problem using RL is given in section III. Section IV illustrates the results followed by conclusions in section V.

II. Preliminaries and Problem Formulation

A. Overview of TD3 Learning Algorithm

We consider an actor-critic framework with a typical MDP where the entire physical system is the agent's environment. The observation space is denoted by O and the action space is denoted by A. Here, $x \in O$ is the observation vector and $u \in A$ is the control input vector. Twin Deyaled Deep Deterministic Policy Gradient algorithm (TD3) is a robust off-policy actor-critic method where we have an actor-network trying to learn an optimal policy $\mu^*(u|x)$ based on Q-value functions estimated by a pair of critic networks. TD3 [2] employs target policy smoothing and delayed policy updates for more stable and effective training in continuous control tasks, improving over the original Deep Deterministic Policy Gradient (DDPG) algorithm [3]. A detailed explanation of the TD3 algorithm can be found in [2].

B. Trajectory Tracking Control with Error Response Shaping

The state space representation of a general dynamical system with state $x(t) \in \mathbb{R}^m$, input $u(t) \in \mathbb{R}^n$ and output $y(t) \in \mathbb{R}^p$ is given by equation (1). Here $f(.)$ and $g(.)$ are linear or nonlinear functions of $x(t)$ and $u(t)$, whose partial derivatives of any order exist and are continuous (C^∞ functions).

$$\dot{x}(t) = f(x(t), u(t)), \quad y(t) = g(x(t)) \tag{1}$$

Let the approximate error dynamics to be followed be given by a first-order system as in equation (2). Here, $K > 0 \in \mathbb{R}^p$ is a predefined constant diagonal matrix. The tracking error is given by $e(t) = y_r(t) - y(t)$ where $y_r(t) \in \mathbb{R}^p$ is the reference output to be tracked. $\dot{e}$ is calculated as $\dot{e}(t) = \lim_{\Delta t \to 0} \frac{e(t+\Delta t) - e(t)}{\Delta t}$.

$$\dot{e}(t) + Ke(t) = 0 \tag{2}$$

The objective here is to design a data-driven RL-based controller that enforces the error trajectory to follow equation (2) for the system given in equation (1), without the knowledge of function $f(.)$. The explicit notation t (function of time) is dropped in this paper subsequently for the variables.

III. Methodology

The methodology followed here is to train the RL agent to learn a policy that satisfies the equation (2).

The agent applies a control input $u_i \in A$ to the system at time step i and receives a reward based on the obtained error $e_i(t)$ and the current system state s_i, as shown in Figure (1). The TD3 agent maximizes the expected Return (cumulative discounted reward) over a certain horizon.

The observation from the environment received by the agent is a vector $o_i \in O$ defined in the equation (3). The components of the observation vector at time step i are the current state ($x_i = [x_{1_i}, x_{2_i}, ...]$), target state ($x_{tg}$), the current value of LHS of equation (2) i.e, $m_i = \dot{e}_i + Ke_i$, the target value of RHS of equation (2) i.e, $m_{tg} = 0$, and the value of K.

$$o_i = [x_{1_i}, \ ..., \ x_{1_{tg}}, \ ... \ m_i, \ m_{tg}, \ K] \tag{3}$$

The reward received by the agent at every time step i is shaped so that the L_2-norm of $m_{tg} - m_i$ is minimized. As $m_{tg} = 0$, $\|m_{tg} - m_i\|$ is effectively equal to $\|\dot{e}_i + Ke_i\|$. The performance index (L^*) to be minimized by the TD3 agent is given by equation (4), where N is the total number of steps in training.

$$L^* = \min \sum_{i=1}^{N} \|\dot{e}_i + Ke_i\| \tag{4}$$

The reward function is shaped to meet this objective while being able to track the reference output. A generic expression of the reward implemented in training the TD3 agent is given by equation (5). The reward received by the agent at the i^{th} step is r_i and the discounted cumulative reward $\sum_{i=1}^{n} \gamma^{(i-1)} r_i$ is to be maximized.

The first term of equation (5) is the performance index from equation (4) which tries to drive the error dynamics of the system as per equation (2). It influences both the transient and steady-state response characteristics. Depending on the weights in equation (5) and the desired specifications, we can obtain an error curve approximate to the solution of equation (2). The second term $\lambda_1 \|e\|^2$ is added to avoid any possible steady-state error dynamics that might arise due to training hyperparameters like the number of episodes trained, etc. The third term $\lambda_2 \|\dot{e}_i\|^2$ mainly affects the transient behavior (e.g. response time) but the final response is dependent on the relative weights, therefore needs to be tuned according to the specifications. The final term $w_2 \|u\|^2$ reduces the control effort. For the experiments in this study, the weights

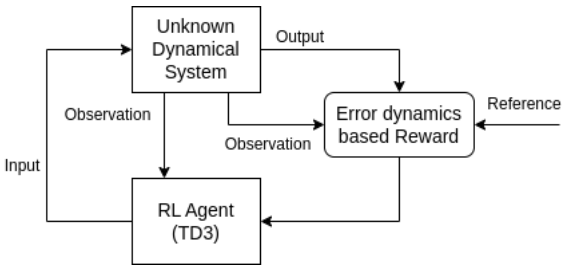

Fig. 1: Block Diagram

$w_1, w_2, \lambda_1, \lambda_2 > 0 \in \mathbb{R}$ are relative and predefined hyper parameters. The changes in error and system response due to changes in weights of equation (5) and K (from equation (2)) are described for different systems in Section IV. It is also shown that the performance objective (equation (4)) is achieved and empirically driven to zero.

$$r_i = -(w_1\|m_{tg} - m_i\| + \lambda_1\|e_i\|^2 + \lambda_2\|\dot{e_i}\|^2 + w_2\|u_i\|^2)$$
$$m_i = \dot{e_i} + Ke_i \quad m_{tg} = 0 \tag{5}$$

IV. Experiments and Results

Environments use OpenAI Gym [1] for reinforcement learning. State updates employ Euler integration on system dynamics with specified variable constraints. The network architecture and hyper-parameters for training the TD3 agent are inherited from the implementation provided in the TD3 paper [2], and are fine-tuned to learn in the current problem setting.

Exploration noise is sampled from the normal distribution $\mathcal{N} \sim (0, 0.1)$. The chosen learning rates for the actor and the critic are 0.0001 and 0.001 respectively. The actor and critic network architectures are similar to each other with 3 hidden fully connected layers (units: $16 \rightarrow 32 \rightarrow 16$) with ReLU activation. The agent is trained with an Adam optimizer, a batch size of 128, a replay buffer of length 100000, and a discounting factor (γ) of 0.99. The soft target update parameter τ is 0.001. For all the systems mentioned in this section, we consider the reward function given by the equation (5) with fine-tuned weights. The systems are trained and tested with a sampling time $\Delta t = 0.05$. After experimentation, a value of 150 is chosen for the number of episodes trained for all the systems. The change in error response parameters for change in K and weights of the reward function in equation (5) are inferred while testing. Please note that K is a scalar for all three cases as we are dealing with single-output systems. Steady-state tracking error (SSE), total control effort (CE), maximum overshoot (O), settling time (T_s) for 98% of the steady-state value, and rise time(T_r) for when the system has reached 90% of its final value, are presented in Tables I to III and can be verified from the respective figures provided in this section.

[1]https://www.gymlibrary.dev/index.html

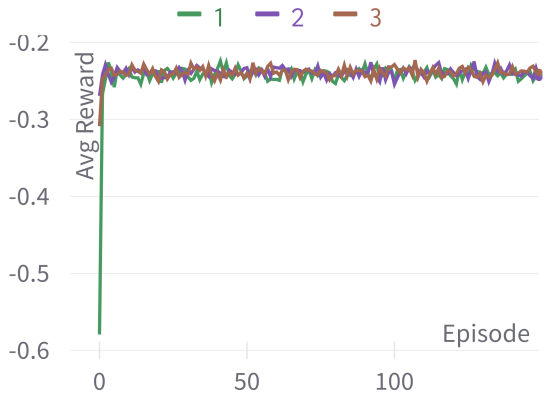

Fig. 2: Average reward vs. Episode plot for training TD3 with the first-order linear system. The three plots correspond to the parameters of entries 1, 2, and 3 of Table I respectively.

A. Linear First Order System

The TD3 agent is trained and tested for a simple linear first-order system given by the equation (6).

$$\dot{x} = -2x + 3u \ , \ y = x \tag{6}$$

Test results are presented for the initial state $x = 1.0$ and the desired final state $y_r = x_r = 0.0$ for different values of parameters in equation (5). The control action constraint is given as $-1 \leq u \leq 1$. The parameter K and the weights $(\lambda_1, \lambda_2, w_1, w_2)$ are chosen empirically and fine-tuned for training. Training is performed for 150 episodes with 1000 steps in each episode, with the reward described in equation (5). The observation vector $o_i \in \mathbb{R}^5$ is given in equation (7).

$$o_i = [x_i, \ 0.0, \ m_i, \ 0.0, \ K] \tag{7}$$

TABLE I: Empirical values of rise time (T_r), percentage absolute steady-state error (SSE), and total control effort (CE) for different values of K and weights for the first-order linear system

S.No.	$(K, \lambda_1, \lambda_2, w_1, w_2)$	Tr (sec)	SSE(%)	CE
1	(4, 0.0, 0.0, 1, 0.001)	0.55	0.0	0.79
2	(0.1, 0.0, 0.0, 1, 0.001)	23.4	0.0	39.68
3	(1, 0.0, 0.0, 1, 0.001)	2.2	0.0	1.03
4	(4, 0.8, 0.0, 1, 0.001)	0.55	0.0	0.80
5	(4, 0.8, 0.1, 0.5 , 0.001)	0.55	0.2	0.77
6	(0.01, 3.0, 0.1, 1, 0.001)	0.4	1.0	1.71
7	(0.01, 2.0, 0.1, 1 , 0.001)	2.1	5.0	3.17

From Figure (2), a rapid convergence in training is observed. From Table I and Figure (3), it can be inferred that as K increases T_r decreases (response becomes faster). To improve the response time and control effort, λ_1, λ_2 can be increased. It

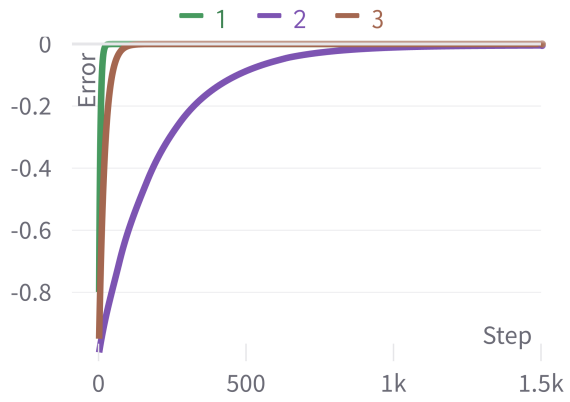

Fig. 3: Error Trajectory for testing the first-order linear system from initial state 1.0 to desired final state 0.0 with the trained TD3 agent. The three plots correspond to the parameters of entries 1, 2, and 3 of Table I respectively.

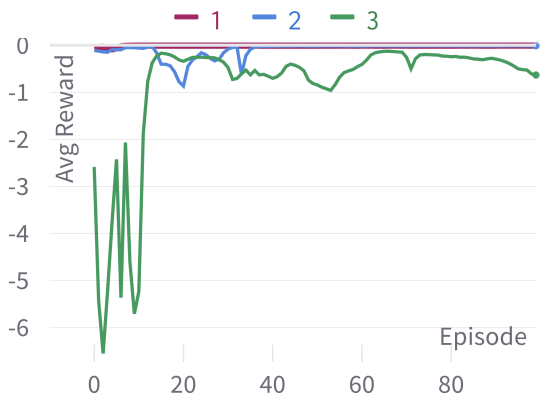

Fig. 4: Average reward vs. Episode plot for training TD3 with the second-order linear system. The three plots correspond to the parameters of entries 1, 2, and 3 of Table II respectively.

is observed that increasing λ_1 reduces SSE (6, 7 entries from Table I). Small variations in λ_2 (4, 5 entries from Table I) do not significantly vary the performance metrics for this system. The objective in the equation (4) is minimized reaching an optimal value of 0 for all the cases in Table I. This is observed from Figure (3) where e goes to zero, effectively driving $\dot{e}$ and the LHS of equation (2) to zero.

B. Linear Second Order System

The TD3 agent is trained and tested for a general linear second-order system with natural frequency ω_n and damping ratio ζ whose transfer function in the Laplace domain is given by equation (9). The environment produces its next observation by performing Euler integration of the ODE in equation (10). The system's initial and desired final state is $[1.0, 0.0]$ and $[0.0, 0.0]$ respectively, and the output is $y = x_1$, where the system state $x = [x_1, x_2]$. The control action constraint is $-1 \leq u \leq 1$. The parameter K and the weights $(\lambda_1, \lambda_2, w_1, w_2)$ are chosen empirically and fine-tuned for training. The network architecture and hyper-parameters are the same as those described in the introductory part of this section. The training is performed for 150 episodes with 1000 steps in each episode, with the reward described in equation (5). The observation vector $o_i \in \mathbb{R}^7$ is given in equation (8).

$$o_i = [x_{i1},\ x_{i2},\ 0.0,\ 0.0,\ m_i,\ 0.0,\ K] \tag{8}$$

$$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{9}$$

For $\omega_n = 1$ and $\zeta = 0.5$ (an underdamped system), the transfer function is $\dfrac{1}{s^2 + s + 1}$.

The state-space system is given by equation (10). Here, $1x_1 = y$ and $x_2 = \dot{y}$.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0.0 & 1.0 \\ -1.0 & -1.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.0 \\ 1.0 \end{bmatrix} u \,, y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{10}$$

In an underdamped second-order system, usually oscillations are seen in the system response and the error trajectory. With properly tuned RL-based reward shaping, the feedback is observed to help increase the damping ratio and reduce the oscillations. For example, overshoot and oscillations are observed for a higher K (E.g. entry 3 in Table II and Figure (5)). The settling time (T_s) from Table II gives an estimation of damping and oscillations in the response. From Figure (5), it can be inferred that a change in K causes a change in damping, T_r, and T_s. From entries 6, and 7 of Table II increasing λ_1 decreases O, T_r, T_s, and SSE but increases CE. The objective in the equation (4) is minimized reaching an optimal value of 0 for all the cases in Table II. This is observed from Figure 5 where e goes to zero, effectively driving $\dot{e}$ and the LHS of equation (2) to zero. From Tables I and II, it is observed that the change in K influences the first-order and second-order linear systems similarly. The relation between T_r, T_s, and K is inferred to be linear. But the metrics O, SSE, and CE depend on all the parameters of the reward function in equation 5, therefore related in a nonlinear fashion. This is because the RL control is nonlinear, making the closed-loop control system also nonlinear. The change in K influences the SSE in a second-order linear system more than that in a first-order linear system. The overall control effort (CE) required for the second-order linear system is observed to be higher than the first-order linear system.

C. Inverted Pendulum

The nonlinear dynamics of an inverted pendulum are given in equation (11). The system consists of a uniform bar with a fixed rotatable end while the other end is free. The goal of the controller is to swing this pendulum bar to a desired target angle and stay there stabilized for a certain time by finding and applying an input torque at the free end.

$$\dot{x_2} = \frac{3g}{2l} \sin x_1 + \frac{3u}{ml^2} \,,\ \dot{x_1} = x_2 \,,\ y = x_1 \tag{11}$$

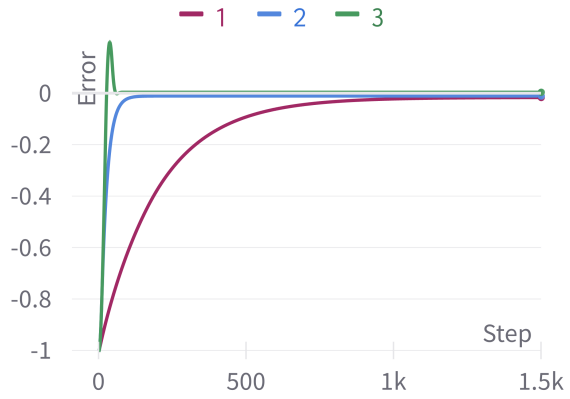

Fig. 5: Error Trajectory for testing the second-order linear system from initial state 1.0 to desired final state 0.0 with the trained TD3 agent. The three plots correspond to the parameters of entries 1, 2, and 3 of Table II respectively.

TABLE II: Table shows the empirical values of rise time (T_r), settling time (T_s), percentage absolute steady-state error (SSE), percentage absolute overshoot (O) and total control effort (CE) for different values of K and weights for the second-order linear system

S.No.	$(K, \lambda_1, \lambda_2, w_1, w_2)$	Tr (sec)	Ts (sec)	O(%)	SSE(%)	CE
1	(0.1, 0.0, 0.0, 1.0, 0.0001)	23.75	54.5	0.0	1.7	85.65
2	(1, 0.0, 0.0, 1, 0.0001)	2.7	5.0	0.0	1.0	17.26
3	(6, 1.0, 0.0, 1.0, 0.0001)	1.6	3.8	19	0.3	42.68
4	(4, 0.0, 0.0, 1.0, 0.0001)	1.4	1.2	0.5	0.5	31.80
5	(4, 1.0, 0.0, 1.0, 0.0001)	2.2	3.3	0.6	0.6	17.92
6	(0.01, 2.0, 0.1, 1.0, 0.0001)	2.3	17.8	6.4	3.3	15.04
7	(0.01, 3.0, 0.1, 1.0, 0.0001)	1.5	2.2	0.3	0.3	26.88

The observation vector $o_i \in \mathbb{R}^9$ is given in equation (12).

$$o_i = [\cos(x_{1_i}),\ \sin(x_{1_i}),\ x_{2_i},\ \cos(x_{tg_i}),\ \sin(x_{tg_i}),\ x_{2_{tg}},\ m_i,\ 0.0,\ K] \tag{12}$$

Here, x_1 is the angle theta (θ) in radians, and x_2 is the angular velocity or the time derivative of θ ($\dot{\theta}$) in radians/sec. $x = [x_1, x_2]$ is the state vector. The control input vector (counterclockwise positive torque) is a scalar denoted by u in Nm. Here, $x_1 \in [0, 2\pi)$ rad, $x_2 \in [-8.0, 8.0]$ rad/s, $u \in [-2.0, 2.0]$ Nm. The reference output is $y_r = x_{1,d}$. All the constraints and parameters are inherited from the Pendulum-v1 environment of OpenAI Gym.

The agent is trained for 1000 episodes with 200 steps in each episode. After every episode the initial state is chosen at random and the final state is the given target state in radians for all the episodes. The network architecture and hyper-parameters are the same as those described in the introductory part of this section.

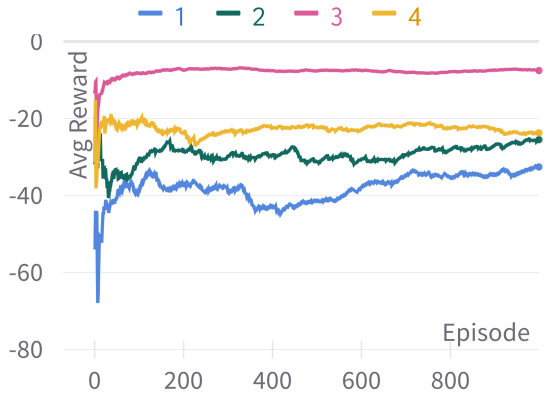

Fig. 6: Average reward vs. Episode for training TD3 with a simple pendulum in OpenAI Gym (Pendulum-v1). The four plots correspond to the parameters of entries 1, 2, 3, and 4 of Table III respectively.

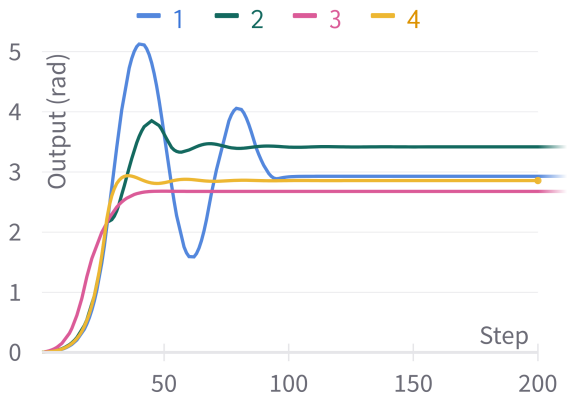

Fig. 7: System Response for testing the trained TD3 agent with the simple pendulum system from 0.0 rad (upright position) to π rad (mean position). The four plots correspond to the parameters of entries 1, 2, 3, and 4 of Table III respectively.

From Table III, it can be observed that there is no direct (or linear) relationship between the reward shaping parameters and the performance metrics due to the underlying non-linearities. First-order like dynamics (following equation (2)) are achieved with some SSE (Figure (7), (8), entry 3 of Table III) and the performance index in equation (4) settles at a non-zero value proportional to SSE for all cases presented. The control effort required is much higher compared to linear systems. There is no apparent relation to increase the performance by decreasing SSE. Increasing K increases damping (Figure (7), 1 and 3 entries of Table III), but it also affects the steady-state value. To decrease the SSE, increasing λ_1 as per entry 2 in table III seems insufficient. By varying λ_2 some change can be observed in Tr, or Ts, and damping. It is noted that each performance metric is dependent on a subset of reward parameters and they cannot be directly inferred. Figure (8) shows the absolute error, and the maximum percentage

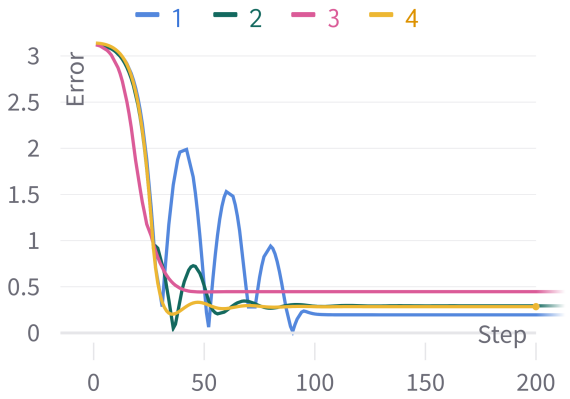

Fig. 8: Trajectory of Absolute Error in rad for testing trained TD3 agent with the simple pendulum system from 0.0 rad (upright position) to π rad (mean position). The four plots correspond to the parameters of entries 1, 2, 3, and 4 of Table III respectively.

overshoot is recorded with proper inference in Table III. The performance metrics CE, SSE, O, Tr, and Ts do not seem to have a direct relationship with the reward parameters.

TABLE III: Empirical values of rise time (T_r), settling time (T_s), percentage absolute steady-state error (SSE) in rad, percentage absolute overshoot(O) and total control effort (CE) in Nm for different K and weights for the simple pendulum

S.N	$(K, \lambda_1, \lambda_2, w_1, w_2)$	Tr (sec)	Ts (sec)	O(%)	SSE(%)	CE
1	(5, 0.0, 0.0, 1, 0.001)	5.2	5.5	60.5	6.05	3312.44
2	(4, 2, 0.1, 1, 0.001)	4.05	5.8	22.29	9.23	7972.81
3	(2, 0.0, 0.0, 1, 0.001)	1.45	2.5	0.0	14.01	7999.94
4	(3, 0.0, 0.0, 1, 0.001)	1.8	5	0.0	8.91	7958.67
5	(5, 2, 0.1, 0.05, 0.001)	6	7.5	0.0	8.59	7990.14
6	(1, 0.5, 0.0, 1, 0.001)	6	10	0.0	7.96	7990.14

V. Conclusions and Future Work

A model-free reinforcement learning-based tracking and response shaping control in continuous time for linear or nonlinear dynamical systems using a TD3 agent is implemented for two linear systems and a nonlinear system. For this purpose, an appropriate reward function is selected to yield the desired error dynamics approximately. Even when the system under consideration is linear, the closed-loop system is nonlinear due to the nonlinear feedback control introduced by RL-based control. Because of this, most of the performance parameters and reward shaping parameters are also related in a nonlinear fashion. We can observe a linear relation between K and T_r, T_s in linear systems but cannot infer any relation between the reward parameters and the performance metrics SSE, CE. For the nonlinear system (pendulum), we have control over the transient response and damping given the agent is properly tuned. The performance metrics SSE, CE, T_r, T_s, and O(maximum overshoot) do not seem to have an apparent relation with the reward parameters.

The key inference here is that the desired specifications for settling time, steady state error, etc. can be achieved by tuning the reward shaping parameters $(K, \lambda_1, \lambda_2, w_1, w_2)$. Our future work involves exploring a mapping between the performance parameters and the reward parameters to directly achieve the control system design specifications through the proposed method.

Acknowledgment

The authors acknowledge the financial support provided by MeiTY (Ministry of Electronics and Information Technology, Govt. of India) under the project, "Capacity Building for Human Resource Development in Unmanned Aircraft System (Drone and related Technology)".

References

[1] C. De Persis and P. Tesi, "Formulas for Data-Driven Control: Stabilization, Optimality, and Robustness," in IEEE Transactions on Automatic Control, vol. 65, no. 3, pp. 909-924, March 2020, doi: 10.1109/TAC.2019.2959924.

[2] Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing function Approximation Error in Actor-Critic Methods. ArXiv. /abs/1802.09477

[3] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. ArXiv. /abs/1509.02971

[4] Kober, J., Peters, J. (2014). Reinforcement Learning in Robotics: A Survey. In: Learning Motor Skills. Springer Tracts in Advanced Robotics, vol 97. Springer, Cham. https://doi.org/10.1007/978-3-319-03194-1_2

[5] H. Kandath, J. Senthilnath and S. Sundaram, "Mutli-agent consensus under communication failure using Actor-Critic Reinforcement Learning," 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 2018, pp. 1461-1465, doi: 10.1109/SSCI.2018.8628943.

[6] Venkataramanan, A. (2022). Reinforcement Learning —Based Transient Response Shaping for Microgrids. ArXiv. /abs/2207.05020

[7] Bhourji, R.S., Mozaffari, S. Alirezaee, S. Reinforcement Learning DDPG–PPO Agent-Based Control System for Rotary Inverted Pendulum. Arab J Sci Eng (2023). https://doi.org/10.1007/s13369-023-07934-2

[8] Modares, H., Lewis, F. L. (2014). Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. Automatica, 50(7), 1780-1792. https://doi.org/10.1016/j.automatica.2014.05.011

[9] Mock, J. and Muknahallipatna, S. (2023) A Comparison of PPO, TD3 and SAC Reinforcement Algorithms for Quadruped Walking Gait Generation. Journal of Intelligent Learning Systems and Applications, 15, 36-56. doi: 10.4236/jilsa.2023.151003.

[10] C. Chen, H. Modares, K. Xie, F. L. Lewis, Y. Wan and S. Xie, "Reinforcement Learning-Based Adaptive Optimal Exponential Tracking Control of Linear Systems With Unknown Dynamics," in IEEE Transactions on Automatic Control, vol. 64, no. 11, pp. 4423-4438, Nov. 2019, doi: 10.1109/TAC.2019.2905215.

[11] Van Nguyen Thi Thanh, Tien Ngo Manh, Cuong Nguyen Manh, Dung Pham Tien, Manh Tran Van, Duyen Ha Thi Kim, and Duy Nguyen Duc, "Autonomous Navigation for Omnidirectional Robot Based on Deep Reinforcement Learning," International Journal of Mechanical Engineering and Robotics Research, Vol. 9, No. 8, pp. 1134-1139, August 2020. doi: 10.18178/ijmerr.9.8.1134-1139