Reinforcement Learning-Based Model-Free Controller for Feedback Stabilization of Robotic Systems

Rupam Singh, *Member, IEEE*, and Bharat Bhushan, *Senior Member, IEEE*

***Abstract*— This article presents a reinforcement learning (RL) algorithm for achieving model-free control of robotic applications. The RL functions are adapted with the least-square temporal difference (LSTD) learning algorithms to develop a model-free state feedback controller by establishing linear quadratic regulator (LQR) as a baseline controller. The classical least-square policy iteration technique is adapted to establish the boundary conditions for complexities incurred by the learning algorithm. Furthermore, the use of exact and approximate policy iterations estimates the parameters of the learning functions for a feedback policy. To assess the operation of the proposed controller, the trajectory tracking and balancing control problems of unmanned helicopters and balancer robotic applications are solved for real-time experiment. The results showed the robustness of the proposed approach in achieving trajectory tracking and balancing control.**

***Index Terms*— Discrete algebraic Riccati equation, dynamic Lyapunov equation, least-square policy iteration, linear quadratic regulator (LQR), reinforcement learning (RL).**

I. Introduction

THE control of autonomous robots to perform complex tasks in dynamic environments has been a crucial area of control. Most of these control aspects are directly related to position control, path planning, trajectory tracking and balancing control of vehicles, etc. [1], [2]. Conventionally, many efforts have been made to achieve the control of autonomous robots, especially in the field of trajectory tracking, path planning [3]–[6], and balancing control [7]–[9].

Recently, reinforcement learning (RL)-based algorithm has been widely used to deal with these complicated control problems. The most general form of RL is Q-learning, which uses action values to improve the behavior of learning iteratively [10]. These Q-values involve states and actions of a learning process to estimate the performance of action to a state. This estimation is computed iteratively using a temporal difference update rule. Furthermore, to learn a function according to the action, the RL agents have two types of policies, on policy and off policy. In on policy, the learning process is dependent on the current action derived from the current policy, whereas in off policy, the learning process is dependent on the action derived from a different policy. The Q-learning process refers to the off-policy technique by adapting a greedy approach to learn the Q-values. In addition to Q-learning, the SARSA algorithm [11], which abbreviates for State Action Reward State Action, refers to the on policy technique to learn the Q-values. In general, RL is used to provide training of agents that develop desired skill set by trial and error which further deliver a reward for successful execution [12] and able to find optimal sequence of commands without any prior assumptions about the world. A lot of research has been done using the RL method in the field of motion planning and path tracking of robots/aerial vehicle [13]–[15]. In general, path complexity is defined as the complexity calculated from multiple path trajectories and due to which it shows the huge variation in tracking path when subjected to slow motion movement [15]. This complexity in motion or path is solved by obtaining spatial entropy from optimization of multiple path trajectories using RL. The Deep RL emerged largely due to Deep Q-Networks merits over other learning method. The Deep Q-Networks method is a robust and powerful technique to handle tracking and complex sequential decision-making problems [16]. Deep RL combines the merits of deep neural network (NN) and is able to process the high-dimensional data in to the lower space efficiently but it limited to simple problems only. These problems can be solved by Markov decision process (MDP)-based methods as they handle stochastic problems efficiently. MDP has been widely used to solve complex stochastic decision problem using mathematical framework in the field of tracking and path planning [17]. An MDP is also used to formulate RL to obtain policy function or sum of discounted rewards [18]. In extension of RL, a new path finding method has been developed named as path finding via reinforcement and imitation multiagent learning (PRIMAL). This combines the merits of RL and imitation learning and is able to learn the agents from the consequences of their position on the other agents [19]. In other work, a model predictive control (MPC) uses online optimization within a predicted framework while taking advantage of receding horizon with soft constraint [20] to provide exact path tracking problem. The advantage of RL over optimization techniques such as MPC and other learning methods [21], [22] is that it does not need a predefined controller structure and is able to give decision for navigation of previously unexplored spaces.

In the light of above advantages, the reliance on RL for interacting with daily basis has increased. In [23], an integral

Manuscript received January 11, 2021; revised August 31, 2021; accepted December 18, 2021. *(Corresponding author: Rupam Singh.)*

Rupam Singh was with the Department of Electrical Engineering, Delhi Technological University, New Delhi 110042, India. She is now with the Institute for Intelligent System Technologies, Alpen-Adria-Universität Klagenfurt, 9020 Klagenfurt, Austria (e-mail: singhrupam99@gmail.com).

Bharat Bhushan is with the Department of Electrical Engineering, Delhi Technological University, New Delhi 110042, India.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3137548.

Digital Object Identifier 10.1109/TNNLS.2021.3137548

RL-based adaptive optimal tracking controller is developed for continuous time linear uncertain systems. Furthermore, with the work in [24], the application of integral RL is extended to linear and nonlinear nonzero sum games. This approach is proven as a model-based approach with proportional–integral (PI) controller and implemented as an offline and online iterative learning using the NNs. In [25], the NN approximating ability of the RL technique is used to develop an adaptive backstepping control for a class of nonstrict-feedback discrete-time systems. This widespread deployment has resulted in critical examination of the potential consequences, especially in the field of physical robotic systems. The major aspect of these consequences adhered to the control of autonomous systems.

To answer this consequence, this article adapts the core algorithm of RL along with the fundamental controller problem in optimal control theory [26]. The linear quadratic regulator (LQR) is used as a baseline controller to define the operating performance of the systems with imperfect knowledge dynamics. Since the LQR is a locally optimal linear controller which requires the knowledge of the system dynamics, they can be combined with RL to achieve online learning through recursive estimation of an internal forward model [27]. This model learns how the current state of the system and the corresponding control signals predict the future state to model RL with LQR. Furthermore, the LQR problem is developed on the basis of a dynamical system where the equation of state evolution is described as a linear function. This linear function depends on the current state and input of the system which are subjected to quadratic cost. This makes the LQR capable of understanding the learning complexities in RL and achieves the control of unknown dynamical systems [28]. Furthermore, to optimize the control process using RL and LQR for trajectory and path tracking problems in robotics, the model-free learning algorithms have been developed.

Conventionally, the classical least square policy iteration algorithms are applied with noiseless [29] and infinite horizon discounted LQR [30] with asymptotic and nonasymptotic bounds, respectively. These applications empirically evaluated and identified that the least-square policy iteration algorithms have worst sample complexity when compared with other model-free and model-based methods. Furthermore, the work by Fazel *et al.* [31] identified that the optimal solution for convergence of policy gradients is associated with polynomial sample complexities. The major drawback is that the work only considered the noise in the system during the initial state and made a deterministic assumption for the transition of other states. Various studies were also focused toward derivative-free optimization of LQR [32], [33] and action space perturbation methods [34], but the previous methods resulted in polynomial sample complexity and the later faced problems with the parameter space perturbation. Furthermore, the variants of least-square policy iteration are studied in [35]–[37] for discounted MDPs where the rewards and feature vectors are uniformly bounded. This resulted in a greedy policy update which required the access for underlying dynamics making the process difficult to implement. From all the above literature, it is identified that the adaptive feedback control algorithms and the conventional approach of implementing RL with LQR have drawbacks due to:

1) Deterministic assumptions based on either input or state value of the system.
2) Polynomial sample complexities.
3) Inefficient convergence while operating with the autonomous systems in the physical world.

Considering the problems associated with developing an optimal controller using LQR and RL, this article developed a model-free algorithm which can learn intermediate representations of the system state instead of directly searching for the parameters of the optimal controller. Initially, to develop this this approach, the fundamental LQR problem is analyzed by mapping it with the specific instance of an MDP. This assumes that for every stabilizable pair of transition matrix, there exists a feedback matrix such that the output of the system is stable. This resulted in a stationary feedback policy which incurred an infinite horizon average cost such that there is a large suboptimality gap between the identified average cost and optimal cost of the control output. Furthermore, to solve this problem, the classical least-square policy iteration is implemented to establish an upper boundary on the sample complexities. This characterizes the asymptotic behavior of the policy gradient method applied to the LQR development and allows a direct comparison between the variations in the upper and lower bounds. Besides, it also reveals the polynomial in state dimension and horizon length worse sample complexities of the control operation. Furthermore, the classical least-square policy iteration is combined with the least-square temporal difference (LSTD) learning to develop a statistical efficient approach that describes the underlying linear dynamics using the input–output data of the system along with the available value functions of the controller. This overcomes the sample complexities and suboptimality gap which is a major concern while developing a model-free controller. The major contributions of this research are:

1) To develop model-free controllers using the LSTD learning algorithm and exact policy iteration.
2) To reduce the variability of system response and achieve feedback stabilization with only access to transition matrices of the plant.
3) To establish a useful baseline for achieving trajectory tracking and balancing control with most fundamental problems in optimal control theory.

Furthermore, a detailed overview of using these learning techniques for developing a model-free controller is discussed as follows: The problem formulation for learning LQR using RL is discussed in Section II, and the development of RL-based model-free control is discussed in Section III. The experimental analysis and implementation of the developed controller are discussed in Section IV, and the conclusion is provided in Section V.

II. Problem Formulation

MDP is a main aspect in problem formulation using RL [38]. The general definition of an infinite horizon average cost MDP is given by a quadruple $(\mathcal{X}, \mathcal{U}, p, c)$, where $\mathcal{X}$ represents the state space, $\mathcal{U}$ represents the input space, $p(\cdot|x, u) \forall x \in \mathcal{X}, u \in \mathcal{U}$ is the probability distribution over the subsequent state conditioned on the variables (x, u), and

$c(x,u)$ is the step cost. To minimize the infinite horizon average cost, the MDP aims at finding a policy $\pi = \{u_t(\cdot)\}_{t=1}^{\infty}$ such that

$$J_* = \inf_{\pi} \lim_{T\to\infty} \sup \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} c(x_t,u_t)\right] \quad \text{s.t. } x_{t+1} \sim p(\cdot|x_t,u_t) \tag{1}$$

where T is the rollout length, and $u_t(\cdot)$ is dependent on previous values $(x_1,u_1,\ldots,x_{t-1},u_{t-1},x_t)$, but not on future values.

In general, even with perfect knowledge of p and c, it is hard to solve (1) without any assumptions. Hence, RL has a fitness assumption which states that the state $\mathcal{X}$ and input $\mathcal{U}$ are finite, which solves (1) using dynamic programming algorithms such as Q-learning. These algorithms scale the size of $\mathcal{X}$ and $\mathcal{U}$ polynomially in time and space complexity. This is mostly feasible for small values of $\mathcal{X}$ and $\mathcal{U}$, and for large values obtained from the discretization of continuous space, the dynamic programming becomes intractable. Hence, this assumption is not applicable for continuous control problems. Another assumption made by RL is function approximation setting [39], which is based on a finite set of basis functions $\{\phi_i(\cdot)\}$. This assumption approximates all the relevant quantities in the span of basis function, providing a natural way of dealing with continuous state-space systems. But the drawback due to the minimal assumptions of basis functions makes this assumption a very general setting. In the light of above issues, this research prioritizes LQR to find an optimal policy π that minimizes the infinite horizon average cost problem

$$J_* = \inf_{\pi} \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} x_t^{\mathrm{T}} S x_t + u_t^{\mathrm{T}} R\right] \tag{2}$$

$$\text{s.t. } x_{t+1} = Ax_t + Bu_t + w_t,\ w_t \sim \mathcal{N}(0,W) \tag{3}$$

where S is an $n\times n$ positive definite cost matrix, R is a $d\times d$ positive definite matrix, and w_t is the process noise which is independent across time. From (1), it can be identified that the LQR problem in (2) and (3) is a specific instance of MDP with $\mathcal{X}=\mathbb{R}^n$, $\mathcal{U}=\mathbb{R}^d$, $p(\cdot|x,u)=\mathcal{N}(Ax+Bu,W)$, and $c(x,u)=x^{\mathrm{T}}Sx+u^{\mathrm{T}}Ru$.

In general, the fundamental LQR problem assumes that for every stabilizable pair (A,B), there exists a feedback matrix K such that $A+BK$ is stable. This assumption resulted in several consequences. At first, the optimal control law is a stationary feedback policy $(u_t = Kx_t)$. Furthermore, the feedback matrix K can be recovered by solving the unique positive definite solution V to the discrete algebraic Riccati equation (d_{ARE})

$$V = A^{\mathrm{T}}VA - A^{\mathrm{T}}VB\left(B^{\mathrm{T}}VB+R\right)^{-1}B^{\mathrm{T}}VA + S \tag{4}$$

and by setting $K = -\left(B^{\mathrm{T}}VB+R\right)^{-1}B^{\mathrm{T}}VA$. The positive definite solution V can be referred as $V = d_{\mathrm{ARE}}(A,B,S,R)$. This gives the optimal cost $J_* = \mathrm{tr}(VW)$.

Hence, the problem statement is given by: For the known cost matrices (S,R) with only access to transition matrices (A,B) through input–output data, find an optimal solution for the LQR problem. The developed algorithm should select a sequence of inputs $\{u_t\}$, and observe the resulting trajectory $\{x_t\}$ such that a controller $\hat{K}$ is returned to stabilize (A,B). This incurs an infinite horizon average cost $J(\hat{K})$ such that there is a large suboptimality gap between $J(\hat{K}) - J_*$.

III. Model-Free LQR

The fundamental representation from value and Q-function of RL for infinite horizon average cost problem is more subtly characterized than the fundamental representation for discounted infinite horizon or finite horizon settings. Considering λ_K as the infinite horizon average cost associated with the feedback policy K. The value function of K is defined as

$$V^K(x):\ \lim_{T\to\infty}\mathbb{E}\left[\sum_{t=0}^{T}\left(x_t^{\mathrm{T}}Sx_t + u_t^{\mathrm{T}}Ru_t - \lambda_K\right)|x_0 = x\right] \quad \text{s.t. } u_t = Kx_t. \tag{5}$$

The Bellman equation for value function defined in (5) is given by

$$\lambda_K + V^K(x) = c(x,Kx) + \mathbb{E}_{x'\sim p(\cdot|x,Kx)}\left[V^K(x')\right]. \tag{6}$$

Here, it is observed that $V^K(x) = x^{\mathrm{T}}Vx$, such that V solves the dynamic Lyapunov equation (d_{LE})

$$V = (A+BK)^{\mathrm{T}}V(A+BK) + S + K^{\mathrm{T}}RK \tag{7}$$

which is denoted as $V = d_{\mathrm{LE}}(A+BK, S+K^{\mathrm{T}}RK)$.

Similar to (5), the Q-function of K is given by

$$Q(x,u) := \lim_{T\to\infty}\mathbb{E}\left[\sum_{t=0}^{T}\left(x_t^{\mathrm{T}}Sx_t + u_t^{\mathrm{T}}Ru_t - \lambda_K\right)|x_0 = x, u_0 = u\right] \quad \text{s.t. } u_t = Kx_t. \tag{8}$$

The Bellman equation for Q-function associated with policy π for a fixed-point equation is given by

$$\lambda + Q(x,u) = c(x,u) + \mathbb{E}_{x'\sim p(\cdot|x,u)}\left[Q\left(x',\pi(x')\right)\right] \tag{9}$$

where $\lambda\in\mathbb{R}$ corresponds to a free parameter to hold the fixed-point equation.

Solving (9), we get $Q(x,u) = s_{\mathrm{vec}}(Q)^T s_{\mathrm{vec}}\left(\begin{bmatrix}x\\u\end{bmatrix}\begin{bmatrix}x\\u\end{bmatrix}^{\mathrm{T}}\right)$, where Q is given by

$$Q = \begin{bmatrix}S & 0\\ 0 & R\end{bmatrix} + \begin{bmatrix}A^{\mathrm{T}}\\ B^{\mathrm{T}}\end{bmatrix}V\begin{bmatrix}A & B\end{bmatrix}. \tag{10}$$

From the above equations, a natural algorithm is obtained to estimate the parameter Q of Q-function. Let $q = s_{\mathrm{vec}}(Q)$, and $Q(x,u) = \phi(x,u)^{\mathrm{T}}q$, where s_{vec}:$\mathrm{Sym}_{n\times n}\to\mathbb{R}^{n(n+1)/2}$ corresponds to a linear operator mapping the space of $n\times n$ symmetric matrix to vectors, and

$$\phi(x,u) = s_{\mathrm{vec}}\left(\begin{bmatrix}x\\u\end{bmatrix}\begin{bmatrix}x\\u\end{bmatrix}^{\mathrm{T}}\right).$$

The Q-function of K in (8) with the fact that

$$\lambda = \left\langle Q, \sigma_w^2\begin{bmatrix}I\\K\end{bmatrix}\begin{bmatrix}I\\K\end{bmatrix}^{\mathrm{T}}\right\rangle$$

is rewritten as

$$q^{\mathrm{T}}\left(\phi(x,u) - \mathbb{E}_{x'\sim p(\cdot|x,u)}\left[\phi\left(x',Kx'\right)\right] + f\right) = c(x,u) \tag{11}$$

where

$$f = s_{\text{vec}}\sigma_w^2\left(\begin{bmatrix} I \\ K \end{bmatrix}\begin{bmatrix} I \\ K \end{bmatrix}^{\text{T}}\right).$$

Therefore, for samples of the form $\{(x_t, u_t, x_{t+1})\}_{t=0}^{T-1}$, the errors in variable approach [40] are used to construct the least-square estimator for q as follows:

$$\hat{q} = \left(\sum_{t=0}^{T-1}\phi(x_t,u_t)(\phi(x_t,u_t)-\phi(x_{t+1},K_{t+1})+f)^{\text{T}}\right)^{-1} \times \sum_{t=0}^{T-1} c(x_t,u_t)\phi(x_t,u_t). \quad (12)$$

This is termed as LSTD L_{STD}-Q estimator. As L_{STD}-Q is an offline policy method, the inputs u_t form any sequence to provide sufficient information for learning. These inputs are not necessarily derived from the feedback policy K. In this research, the inputs of the form (13) are considered

$$u_t = K_{\text{play}}x_t + \eta_t, \quad \text{with } \eta_t \sim \mathcal{N}(0, \sigma_\eta^2 I) \quad (13)$$

where K_{play} is the exploration controller stabilizing (A, B), and η_t is the injected noise to provide sufficient information for learning.

Let $\phi_t := \phi(x_t, u_t)$, $\psi_t := \phi(x_t, Kx_t)$ and $c_t := x_t^{\text{T}}Sx_t + u_t^{\text{T}}Ru_t$. The (L_{STD})-Q estimator for q follows:

$$\hat{q} := \left(\sum_{t=1}^{T}\phi_t(\phi_t-\psi_{t+1}+f)^{\text{T}}\right)^{\dagger}\sum_{t=1}^{T}\phi_t c_t \quad (14)$$

where $(\cdot)^{\dagger}$ is the Moore–Penrose pseudo inverse. Furthermore, a nonasymptotic bound on the estimator quality which is measured in term of error $\|q-\hat{q}\|$ incurred by L_{STD}-Q is established.

Theorem: Consider the policies K_{play} and K stabilize (A, B). Fix $\delta \in (0, 1)$ and assume $A + BK_{\text{play}}$ and $A + BK$ are (τ, ρ)-stable. The initial state is $x_0 \sim \mathcal{N}(0, \Sigma_0)$ and inputs $u_t = K_{\text{play}}x_t + \eta_t$, with $\eta_t \sim \mathcal{N}(0, \sigma_\eta^2 I)$. Assuming, $\sigma_\eta \le \sigma_w$, the steady-state covariance denoted by P_∞ of trajectory $\{x_t\}$ is given by

$$P_\infty = d_{\text{LE}}\left(\left(A+BK_{\text{play}}\right)^{\text{T}}, \sigma_w^2 I + \sigma_\eta^2 BB^{\text{T}}\right). \quad (15)$$

The proxy variance $\bar{\sigma}^2$ is defined as

$$\bar{\sigma}^2 := \tau^2\rho^4\|\Sigma_0\| + \|P_\infty\| + \sigma_\eta^2\|B\|^2. \quad (16)$$

If T satisfies

$$T \ge \tilde{\mathcal{O}}(1) \times \max\left\{(n+d)^2, \frac{\tau^4}{\rho^4(1-\rho^2)^2}\frac{(n+d)^4}{\sigma_\eta^4} \times \sigma_w^2\bar{\sigma}^2\left\|K_{\text{play}}\right\|_+^4\|K\|_+^8\left(\|A\|^4+\|B\|^4\right)_+\right\} \quad (17)$$

with least probability $1-\delta$

$$\|\hat{q}-q\| \le \tilde{\mathcal{O}}(1)\frac{\tau^2}{\rho^2(1-\rho^2)}\frac{(n+d)}{\sigma_\eta^2\sqrt{T}}\sigma_w\bar{\sigma}\left\|K_{\text{play}}\right\|_+^2\|K\|_+^4 \times \left(\|A\|^2+\|B\|^2\right)_+\left\|Q^K\right\|_F \quad (18)$$

where $\tilde{\mathcal{O}}(1)$ hides polylog$(n, T, \|\Sigma_0\|, \|P_\infty\|, \|K_{\text{play}}\|, (T/\delta), (1/\sigma_\eta))$ factors.

Hence, theorem 1 states that to achieve error $\|q-\hat{q}\| \le \varepsilon$, $T \le \tilde{\mathcal{O}}((n+d)^4, (1/\sigma_\eta^4)((n+d)^3/\varepsilon^2))$ time steps are sufficient.

To prove the theorem, some recent advances need to be considered. First, a basic inequality of L_{STD}-Q needs to be developed which serves as a starting point to the analysis [41]. Next, we combine the self-normalized inequalities in [42] with the small ball techniques available in [43]. This is necessary to reduce the order of dependence $(1/\sigma_\eta^8)$ of results in [41] to $(1/\sigma_\eta^4)$.

Proof: Due to stability, P_t converges to a limit $P_\infty = d_{\text{LE}}((A+BK_{\text{play}})^{\text{T}}, \sigma_w^2 I + \sigma_\eta^2 BB^{\text{T}})$, where P_t is given by

$$P_t := \sum_{k=0}^{t-1}\left(A+BK_{\text{play}}\right)^k\left(\sigma_w^2 I + \sigma_\eta BB^{\text{T}}\right)\left(\left(A+BK_{\text{play}}\right)^{\text{T}}\right)^k.$$

The covariance of x_t for $t \ge 1$ is

$$\text{cov}(x_t) = \Sigma_t := P_t + \left(A+BK_{\text{play}}\right)^t\Sigma_0\left(\left(A+BK_{\text{play}}\right)^{\text{T}}\right)^t.$$

The data matrices defined by

$$\Phi = \begin{bmatrix}\phi_1^{\text{T}} \\ \vdots \\ \phi_T^{\text{T}}\end{bmatrix}, \quad \Psi_+ = \begin{bmatrix}\psi_2^{\text{T}} \\ \vdots \\ \psi_{T+1}^{\text{T}}\end{bmatrix}$$

$$c = (c_1, \ldots, c_T)^{\text{T}}, \quad \text{and } F = \begin{bmatrix} f^{\text{T}} \\ \vdots \\ f^{\text{T}}\end{bmatrix}$$

update the L_{STD}-Q estimator as follows:

$$\hat{q} = \left(\Phi^{\text{T}}(\Phi - \Psi_+ + F)\right)^{\dagger}\Phi^{\text{T}}c.$$

Furthermore, consider Ξ be the matrix defined by

$$\Xi = \begin{bmatrix}\mathbb{E}[\phi(x_2, Kx_2)|x_1, u_1]^{\text{T}} \\ \vdots \\ \mathbb{E}\left[\phi(x_{T+1}, Kx_{T+1})|x_T, u_T\right]^{\text{T}}\end{bmatrix}$$

and $\otimes_s$ defines the symmetric Kronecker product [44], [45]. From $q = s_{\text{vec}}(Q)$, the starting point of the analysis in [41] is discussed in lemma 1 as follows.

Lemma 1: Consider $L := \begin{bmatrix} I \\ K \end{bmatrix}[A\ B]$, Φ has a full column rank, and $(\|(\Phi^{\text{T}}\Phi)^{-(1/2)}\Phi^{\text{T}}(\Xi-\Psi_+)\|)/(\sigma_{\min}(\Phi)\sigma_{\min}(I - L\otimes_s L)) \le (1/2)$. The available error incurred is

$$\|\hat{q}-q\| \le \frac{\left\|\left(\Phi^{\text{T}}\Phi\right)^{-\frac{1}{2}}\Phi^{\text{T}}(\Xi-\Psi_+)q\right\|}{\sigma_{\min}(\Phi)\sigma_{\min}(I - L\otimes_s L)}. \quad (19)$$

From the proof to the lemma in [41], $\|q-\hat{q}\| \leq (\|(\Phi^{\mathrm{T}}\Phi)^{-(1/2)}\Phi^{\mathrm{T}}(\Xi-\Psi_+)\hat{q}\|)/(\sigma_{\min}(\Phi)\sigma_{\min}(I-L\otimes_s L))$ is obtained. Considering $q-\hat{q}=\Delta$, and by triangle inequality

$$\|\Delta\| \leq \frac{\left\|\left(\Phi^{\mathrm{T}}\Phi\right)^{-\frac{1}{2}}\Phi^{\mathrm{T}}(\Xi-\Psi_+)\right\|\|\Delta\|}{\sigma_{\min}(\Phi)\sigma_{\min}(I-L\otimes_s L)} + \frac{\left\|\left(\Phi^{\mathrm{T}}\Phi\right)^{-\frac{1}{2}}\Phi^{\mathrm{T}}(\Xi-\Psi_+)q\right\|}{\sigma_{\min}(\Phi)\sigma_{\min}(I-L\otimes_s L)}.$$

To apply lemma 1 for starting point analysis, the minimum singular value $\sigma_{\min}(\Phi)$ needs to be bounded. This is achieved by adapting martingale small-ball condition as stated in [43].

Proposition 1: Given a random vector $y \in \mathcal{S}^{n+d-1}$, the process $Z_t := \langle \phi_t, y\rangle$, filteration $\mathcal{F}_t := \sigma(\{u_i, w_{i-1}\}_{i=0}^t)$, and matrix $C := \begin{bmatrix} I & 0 \\ K & I \end{bmatrix}\begin{bmatrix} \sigma_w I & 0 \\ 0 & \sigma_\eta I \end{bmatrix}$ are defined, such that $(Z_t)_{t\geq 1}$ satisfies the martingale small-ball condition $(1, \sigma_{\min}^2(C), 1/324)$. This condition is given by

$$\mathbb{P}\big(|Z_{t+1}| \geq \sigma_{\min}^2(C)|\mathcal{F}_t\big) \geq 1/324.$$

Proof: Let $Y := s_{\mathrm{vec}}^{-1}(y)$, $\mu_t := Ax_t + Bu_t$, and

$$\begin{bmatrix} x_{t+1} \\ u_{t+1} \end{bmatrix} = \begin{bmatrix} I \\ K \end{bmatrix}\mu_t + \begin{bmatrix} I & 0 \\ K & I \end{bmatrix}\begin{bmatrix} w_t \\ \eta_{t+1} \end{bmatrix}$$

is available.

Therefore, for the given filtration

$$\begin{aligned}\mathcal{F}_t, \langle \phi_{t+1}, y\rangle &= \begin{bmatrix} x_{t+1} \\ u_{t+1} \end{bmatrix}^{\mathrm{T}} Y \begin{bmatrix} x_{t+1} \\ u_{t+1} \end{bmatrix} \\ &= \left(\begin{bmatrix} I \\ K \end{bmatrix}\mu_t + \begin{bmatrix} I & 0 \\ K & I \end{bmatrix}\begin{bmatrix} w_t \\ \eta_{t+1} \end{bmatrix}\right)^{\mathrm{T}} \\ &\quad \times Y\left(\begin{bmatrix} I \\ K \end{bmatrix}\mu_t + \begin{bmatrix} I & 0 \\ K & I \end{bmatrix}\begin{bmatrix} w_t \\ \eta_{t+1} \end{bmatrix}\right)\end{aligned}$$

which is a Gaussian polynomial of degree 2. Hence, from the results of Gaussian hypercontractivity in [46]

$$\mathbb{E}\big[|Z_{t+1}|^4|\mathcal{F}_t\big] \leq 81\mathbb{E}\big[|Z_{t+1}|^2|\mathcal{F}_t\big]^2.$$

This invokes Paley–Zygmund inequality, which concludes that for any $\theta \in (0,1)$

$$\mathbb{P}\left(|Z_{t+1}| \geq \sqrt{\theta\mathbb{E}\big[|Z_{t+1}|^2|\mathcal{F}_t\big]}|\mathcal{F}_t\right) \geq (1-\theta)^2\frac{\mathbb{E}\big[|Z_{t+1}|^2|\mathcal{F}_t\big]^2}{\mathbb{E}\big[|Z_{t+1}|^4|\mathcal{F}_t\big]} \geq \frac{(1-\theta)^2}{81}.$$

Furthermore, the updated proposition is given as follows.

Proposition 2: Let μ, C, and Y be fixed, $g\sim\mathcal{N}(0, I)$, and $\mathbb{E}[((\mu+C_g)^{\mathrm{T}}Y(\mu+C_g))^2] \geq 2\|C^{\mathrm{T}}YC\|_F^2$.

Proof: Let $Z := (\mu+C_g)^{\mathrm{T}}Y(\mu+C_g)$, and $\mathbb{E}[Z]^2 \geq \mathbb{E}[(Z-\mathbb{E}[Z]^2)]$ is known. Computing these gives $\mathbb{E}[Z] = \mu^{\mathrm{T}}Y\mu + \mathrm{tr}(C^{\mathrm{T}}YC)$. Hence, $Z-\mathbb{E}Z = g^{\mathrm{T}}C^{\mathrm{T}}YCg - \mathrm{tr}(C^{\mathrm{T}}YC) + 2\mu^{\mathrm{T}}YCg$.

Therefore,

$$\begin{aligned}\mathbb{E}\big[(Z-\mathbb{E}[Z])^2\big] &\geq \mathbb{E}\Big[\big(g^{\mathrm{T}}C^{\mathrm{T}}YCg - \mathrm{tr}\big(C^{\mathrm{T}}YC\big)\big)^2\Big] \\ &= 2\big\|C^{\mathrm{T}}YC\big\|_F^2.\end{aligned}$$

The basic properties of Kronecker product are used to invoke the proposition as follows:

$$\begin{aligned}\mathbb{E}\big[Z_{t+1}^2|\mathcal{F}_t\big] &\geq 2\big\|C^{\mathrm{T}}YC\big\|_F^2 = 2\big\|\big(C^{\mathrm{T}}\otimes C^{\mathrm{T}}\big)y\big\|^2 \\ &\geq 2\sigma_{\min}^2\big(C^{\mathrm{T}}\otimes C^{\mathrm{T}}\big) = 2\sigma_{\min}^4(C).\end{aligned}$$

Here, the block martingale small-ball condition is bounded in place, and hence, the lower bound of $\sigma_{\min}(\Phi)$ is obtained as discussed using proposition 3 [43].

Proposition 3: For $\delta \in (0,1)$, let $\sigma_\eta \leq \sigma_w$, and (20), as shown at the bottom of the page. For the condition where $A+BK_{\mathrm{play}}$ is (τ,ρ)-stable, $\sigma_{\min}(\Phi) \geq (\sigma_\eta^2/1296\sqrt{8})(1)/(1+\|K_{\mathrm{play}}\|^2)\sqrt{T}$, and $\|\Phi^{\mathrm{T}}\Phi\| \leq (12T/\delta)(1+\|K_{\mathrm{play}}\|^2)^2(\tau^2\rho^2 n\|\Sigma_0\| + \mathrm{tr}(P_\infty))^2$ are available with probability at least $1-\delta$.

Proof: Initially, a crude upper bound is computed using Markov's inequality

$$\mathbb{P}\big(\|\Phi\|^2 \geq t^2\big) = \frac{\mathbb{E}[\lambda_{\max}(\Phi^{\mathrm{T}}\Phi)]}{t^2} \leq \frac{\mathrm{tr}(\mathbb{E}[\Phi^{\mathrm{T}}\Phi])}{t^2}.$$

Next, $\mathbb{E}[\|\phi_t\|^2]$ is upper bounded.

Consider, $z_t = (x_t, u_t)$, then $\mathbb{E}[\|\phi_t\|^2] = \mathbb{E}[\|z_t\|^4] \leq 3(\mathbb{E}[\|z_t\|^2])^2$.

This bound $\mathbb{E}[\|z_t\|^2] \leq (1+\|K_{\mathrm{play}}\|^2)\mathrm{tr}(\Sigma_t) + \sigma_\eta^2 d$, as follows:

$$\begin{aligned}\sqrt{\mathbb{E}[\|\phi_t\|^2]} &\leq \sqrt{3}\Big(1+\big\|K_{\mathrm{play}}\big\|^2\Big)\mathrm{tr}(\Sigma_t) + \sigma_\eta^2 d \\ \sqrt{\mathbb{E}[\|\phi_t\|^2]} &\leq \sqrt{3}\Big(1+\big\|K_{\mathrm{play}}\big\|^2\Big)(\tau^2\rho^2 n\|\Sigma_0\| + \mathrm{tr}(P_\infty)) + \sigma_\eta^2 d \\ \sqrt{\mathbb{E}[\|\phi_t\|^2]} &\leq 2\sqrt{3}\Big(1+\big\|K_{\mathrm{play}}\big\|^2\Big)\big(\tau^2\rho^2 n\|\Sigma_0\| + \mathrm{tr}(P_\infty)\big).\end{aligned}$$

From the above bounds, the last inequality holds $\sigma_\eta^2 d \leq \sigma_w^2 n \leq \mathrm{tr}(P_\infty)$. Hence, the updated Markov's inequality is

$$\mathbb{P}\left(\|\Phi\| \geq \frac{\sqrt{T}}{\sqrt{\delta}}2\sqrt{3}\Big(1+\big\|K_{\mathrm{play}}\big\|^2\Big)\big(\tau^2\rho^2 n\|\Sigma_0\| + \mathrm{tr}(P_\infty)\big)\right) \leq \delta.$$

Furthermore, for a fixed $\varepsilon > 0$, $\mathcal{N}(\varepsilon)$ denotes an ε-net of a unit sphere $\mathcal{S}^{(n+d)(n+d+1)/2-1}$. Therefore, from the proposition available in [43] and union bound over $\mathcal{N}(\varepsilon)$: $\mathbb{P}(\min_{v\in\mathcal{N}(\varepsilon)}\|\Phi v\| \geq (\sigma_{\min}^2(C)/324\sqrt{8})\sqrt{T}) \geq 1-(\delta/2)$. From the result, it is observed that $\sigma_{\min}(\Phi) = \inf_{\|v\|=1}\|\Phi v\| \geq \min_{v\in\mathcal{N}(\varepsilon)}\|\Phi v\| - \|\Phi\|\varepsilon$, which is a union bound over two events. Hence, the proof for the proposition is concluded based on a lemma in [47] which yields $\sigma_{\min}^2(C) \geq (\sigma_n^2/2)(1)/(1+\|K_{\mathrm{play}}\|^2)$ since $\sigma_\eta \leq \sigma_w$.

Now, the self-normalized martingale terms need to be upper bounded: $\|(\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}(\Xi-\Psi_+)\|$ and

$$T > 324^2\cdot 8\left((n+d)^2\log\left(1+\frac{20736\sqrt{3}}{\sqrt{\delta}}\frac{\Big(1+\big\|K_{\mathrm{play}}\big\|^2\Big)^2\big(\tau^2\rho^2 n\|\Sigma_0\| + \mathrm{tr}\,(P_\infty)\big)}{\sigma_n^2}\right) + \log\left(\frac{2}{\delta}\right)\right). \tag{20}$$

$\|(\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}(\Xi-\Psi_+)q\|$. This is achieved by self-normalized tail bounds in [42].

Lemma 2: Let $\{\mathcal{F}_t\}$ be the filtration, $\{x_t\}$ is a $\mathbb{R}^{d1}$ process adapted to $\{\mathcal{F}_t\}$, $\{w_t\}$ is $\mathbb{R}^{d2}$ martingale difference sequence adapted to $\{\mathcal{F}_t\}$, and V is a fixed positive definite matrix of size $d_1 \times d_1$ defined as

$$\bar{V}_t = V + \sum_{s=1}^{t} x_s x_s^{\mathrm{T}}, \quad S_t = \sum_{s=1}^{t} x_s w_{s+1}^{\mathrm{T}}.$$

1) Consider $h \in \mathbb{R}^{d2}$ for any fixed unit and $\langle w_t, h\rangle$ is conditionally R-sub–Gaussian, i.e., $\forall\lambda \in \mathbb{R}, t \geq 1, \mathbb{E}[e^{\lambda\langle w_{t+1},h\rangle}|\mathcal{F}_t] \leq e^{(\lambda^2R^2/2)}$. This gives $\|\bar{V}_t^{-(1/2)}S_t\|^2 \leq 8R^2(d_2\log 5 + \log(\det(\bar{V}_t)^{(1/2)}\det(V)^{-(1/2)}/\delta))$ with probability at least $1-\delta, \forall t \geq 1$.
2) Consider $\bar{\delta}$ satisfies the condition: $\sum_{s=2}^{T+1}\mathbb{P}(\|w_s\| > R) \leq \bar{\delta}$. This gives $\|\bar{V}_t^{-(1/2)}S_t\|^2 \leq 32R^2(d_2\log 5 + \log((\det(\bar{V}_t)^{(1/2)}\det(V)^{-(1/2)}/\delta)))$ with probability at least $1-\delta-\bar{\delta}, \forall 1 \leq t \leq T$.

From the proof of corollary 1 in [42], which yields 1) using standard covering argument and union bounding over $\mathcal{N}(1/2)$ and 2) using simple stopping time argument to handle truncation. Using this lemma, the martingale difference terms can be bounded.

Furthermore, for the hypothesis of proposition 3 with probability at least $1-\delta$, $\|(\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}(\Xi-\Psi_+)q\| \leq (n+d)\sigma_w(\tau^2\rho^4\|\Sigma_0\| + \|P_\infty\| + \sigma_\eta^2\|B\|^2)^{1/2}(1+\|K\|^2)\|Q\|_F \times$ polylog $(n, \tau, \|\Sigma_0\|, \|P_\infty\|, K_{\text{play}}, (T/\delta), (1/\sigma_\eta))$, and $\|(\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}(\Xi-\Psi_+)\| \leq (n+d)^2\sigma_w(\tau^2\rho^4\|\Sigma_0\| + \|P_\infty\| + \sigma_\eta^2\|B\|^2)^{1/2}(1+\|K\|^2) \times$ polylog $(n, \tau, \|\Sigma_0\|, \|P_\infty\|, K_{\text{play}}, (T/\delta), (1/\sigma_\eta))$.

With the above aspects, the proof of the theorem can be obtained as follows:

For $k \geq 1$, $\sigma_{\min}(I - L\otimes_s L)$ is lower bounded using (τ,ρ)-stability of $A + BK$.

$$\begin{aligned}\|L^K\| &= \left\|\begin{bmatrix} I \\ K\end{bmatrix}(A+BK)^{k-1}[\,A\;\;B\,]\right\| \\ &\leq 2\|K\|_+\left\|[\,A\;\;B\,]\right\|\tau\rho^{k-1} \\ &\leq \frac{2\|K\|_+\max\left\{1,\sqrt{\|A\|^2+\|B\|^2}\right\}}{\rho}\tau\cdot\rho^k.\end{aligned}$$

Hence, for the last term of the above equation L is $((2\|K\|_+\max\{1,(\|A\|^2+\|B\|^2)^{1/2}\})/(\rho)\tau,\rho)$-stable.

As, $\sigma_{\min}(I - L\otimes_s L) = (1)/(\|(I-L\otimes_s L)^{-1}\|)$ is known, for any unit norm υ

$$\begin{aligned}\left\|(I-L\otimes_s L)^{-1}\upsilon\right\| &= \left\|(I-L\otimes_s L)^{-1}s_{\text{vec}}\left(s_{\text{vec}}^{-1}(\upsilon)\right)\right\| \\ &= \left\|d_{\text{LE}}\left(L^{\mathrm{T}}, s_{\text{vec}}^{-1}(\upsilon)\right)\right\|_F \\ &\leq \frac{4\|K\|_+^2\left(\|A\|^2+\|B\|^2+\tau^2\right)}{\rho^2\left(1-\rho^2\right)}.\end{aligned}$$

Since the last inequality depends on the Frobenius norm of d_{LE} for a stable matrix A, the available lower bound is

$$\sigma_{\min}(I - L\otimes_s L) \geq \frac{\rho^2(1-\rho^2)}{4\|K\|_+^2\left(\|A\|^2+\|B\|^2\right)+\tau^2}.$$

From proposition 3, as long as $T \geq \tilde{\mathcal{O}}(1)(n+d)^2$ with probability $1-(\delta/2)$

$$\sigma_{\min}(\Phi) \geq c\frac{\sigma_\eta^2}{\left\|K_{\text{play}}\right\|_+^2}\sqrt{T}$$

and for the hypothesis of proposition 3, with probability $1-(\delta/2)$

$$\begin{aligned}&\left\|\left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}(\Xi-\Psi_+)q\right\| \\ &\leq (n+d)\sigma_w\sqrt{\tau^2\rho^4\|\Sigma_0\| + \|P_\infty\| + \sigma_\eta^2\|B\|^2} \\ &\quad\times\|K\|_+^2\left\|Q^K\right\|_F\tilde{\mathcal{O}}(1) \\ &\left\|\left(\Phi^{\mathrm{T}}\Phi\right)^{-1}\Phi^{\mathrm{T}}(\Xi-\Psi_+)\right\| \\ &\leq (n+d)^2\sigma_w\sqrt{\tau^2\rho^4\|\Sigma_0\| + \|P_\infty\| + \sigma_\eta^2\|B\|^2} \\ &\quad\times\|K\|_+^2\tilde{\mathcal{O}}(1).\end{aligned}$$

Initially, the condition $(\|(\Phi^{\mathrm{T}}\Phi)^{-(1/2)}\Phi^{\mathrm{T}}(\Xi-\Psi_+)\|)/(\sigma_{\min}(\Phi)\sigma_{\min}(I-L\otimes_s L)) \leq (1/2)$ is checked from lemma 1. Next, a sufficient condition, which T satisfies, is obtained as

$$\begin{aligned}T \geq\ &\mathcal{O}(1)\frac{\left\|K_{\text{play}}\right\|_+^4}{\sigma_\eta^4}\cdot(n+d)^4 \\ &\times\sigma_w^2\left(\tau^2\rho^2\|\Sigma_0\| + \|P_\infty\| + \sigma_\eta^2\|B\|^2\right) \\ &\times\|K\|_+^4\cdot\frac{\|K\|_+^4\left(\|A\|^2+\|B\|^2\right)_+^2\tau^4}{\rho^4\left(1-\rho^2\right)^2}\end{aligned}$$

and

$$\begin{aligned}T =\ &\tilde{\mathcal{O}}(1)\frac{\tau^4}{\rho^4\left(1-\rho^2\right)^2}\frac{\left(n+d^4\right)}{\sigma_\eta^4}\sigma_w^2 \\ &\times\left(\tau^2\rho^4\|\Sigma_0\| + \|P_\infty\| + \sigma_\eta^2\|B\|^2\right) \\ &\times\left\|K_{\text{play}}\right\|_+^4\|K\|_+^8\left(\|A\|^4+\|B\|^4\right)_+.\end{aligned}$$

Once the sufficient condition is satisfied, the error $\|\hat{q}-q\|$ is bounded using

$$\begin{aligned}&\tilde{\mathcal{O}}(1)\frac{\left\|K_{\text{play}}\right\|_+^2}{\sigma_\eta^2\sqrt{T}}\cdot(n+d)\sigma_w\sqrt{\tau^2\rho^4\|\Sigma_0\| + \|P_\infty\| + \sigma_\eta^2\|B\|^2} \\ &\quad\times\|K\|_+^2\left\|Q^K\right\|_F\cdot\frac{\|K\|_+^2\left(\|A\|^2+\|B\|^2\right)_+\tau^2}{\rho^2\left(1-\rho^2\right)} \\ &=\tilde{\mathcal{O}}(1)\frac{\tau^2}{\rho^2(1-\rho^2)}\frac{(n+d)}{\sigma_\eta^2\sqrt{T}}\sigma_w\sqrt{\tau^2\rho^4\|\Sigma_0\| + \|P_\infty\| + \sigma_\eta^2\|B\|^2} \\ &\quad\times\left\|K_{\text{play}}\right\|_+^2\|K\|_+^4\left(\|A\|^2+\|B\|^2\right)_+\left\|Q^K\right\|_F.\end{aligned}$$

The implementation of the RL algorithm for a plant operating with LQR control can be represented as shown in Fig. 1.

A. Policy Iteration for LQR

Let V_0 be the value function associated with stabilizing controller K_0 for (A, B). The following recursions are applied for $t = 0, 1, 2, \ldots$

$$K_{t+1} = -\left(R + B^{\mathrm{T}}V_tB\right)^{-1}B^{\mathrm{T}}V_tA \tag{21}$$

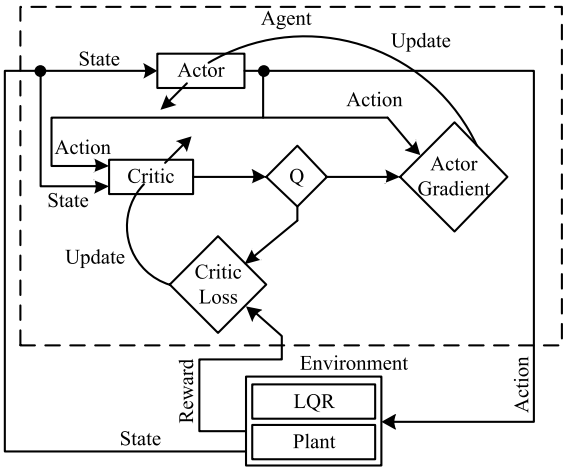

Fig. 1. Visual depiction of the agent in RL for a plant operating with LQR control.

$$V_{t+1} = d_{\mathrm{LE}}\big(A + BK_{t+1}, S + K_{t+1}^{\mathrm{T}} R K_{t+1}\big). \tag{22}$$

It is observed that the above recurrence is related, but different than the value iteration that initiates from a positive definite V_0 and recurses

$$V_{t+1} = A^{\mathrm{T}} V_t A - A^{\mathrm{T}} V_t B\big(R + B^{\mathrm{T}} V_t B\big)^{-1} B^T V_t A + S.$$

1) Exact Policy Iteration: As the behavior of the value iteration is well-studied in the literature [48], [49], this section analyzes the behavior of an exact policy iteration developed on fixed point theory [50]. The major advantage of the fixed point theory is that it handles the errors in policy iteration during the updates. Furthermore, an invariant metric δ_∞ on positive definite matrix is considered as the key component for this analysis which is given by

$$\delta_\infty(A, B) := \left\| \log\left(A^{-\frac{1}{2}} B A^{-\frac{1}{2}}\right) \right\|.$$

Initially, the matric inversion is observed

$$\begin{aligned} &S + A^{\mathrm{T}}\big(BR^{-1}B^{\mathrm{T}} + V^{-1}\big)^{-1} A \\ &\quad = S + A^{\mathrm{T}} V A - A^{\mathrm{T}} V B\big(R + B^{\mathrm{T}} V B\big)^{-1} B^{\mathrm{T}} V A =: F(V). \end{aligned}$$

For $V = F(V)$, let V_* be the unique positive definite solution. Hence, the positive definite V is given as

$$\delta_\infty(F(V), V_*) \leq \frac{\alpha}{\lambda_{\min}(S) + \alpha} \delta_\infty(V, V_*) \tag{23}$$

where $\alpha = \max\{\lambda_{\max}\big(A^{\mathrm{T}} V A\big), \lambda_{\max}(A^{\mathrm{T}} V_* A)\}$.

As the Riccati operator $F(V)$ is contractive in δ_∞ metric, the analysis is carried out by combining the contraction property with the policy iteration analysis in [51]. This is formulated as a proposition as follows.

Proposition 4: Consider S and R to be positive definite and the discrete algebraic Riccati equation (d_{ARE}) has a unique positive definite solution. For $t = 0, 1, 2, \ldots$, a stabilizing policy K_0 and positive definite $V_0 = d_{\mathrm{LE}}(A + BK_0, S + K_0^{\mathrm{T}} R K_0)$ have the following sequence of updates:

$$\begin{aligned} K_{t+1} &= -\big(R + B^{\mathrm{T}} V_t B\big)^{-1} B^{\mathrm{T}} V_t A \\ V_{t+1} &= d_{\mathrm{LE}}\big(A + BK_{t+1}, S + K_{t+1}^{\mathrm{T}} R K_{t+1}\big). \end{aligned}$$

This holds the statement as

1) K_t stabilizes $(A < B) \forall t = 0, 1, 2, \ldots$
2) $V_* \preccurlyeq V_{t+1} \preccurlyeq V_t \forall t = 0, 1, 2, \ldots$
3) $\delta_\infty(V_{t+1}, V_*) \leq \rho \cdot \delta_\infty(V_t, V_*) \forall t = 0, 1, 2, \ldots,$ with $\rho := (\lambda_{\max}(A^{\mathrm{T}} V_0 A))/(\lambda_{\min}(S) + \lambda_{\max}(A^{\mathrm{T}} V_0 A))$.

Consequently, $\delta_\infty(V_t, V_*) \leq \rho^t \cdot \delta_\infty(V_0, V_*) \forall t = 0, 1, 2, \ldots$

2) Approximate Policy Iteration: In this section, the approximate policy iteration is analyzed by developing an algorithm to estimate the parameters of Q-function for policy K. In general, the approximate policy iteration is considered as a basic form of dynamic learning. This algorithm can be used as subroutine to achieve policy optimization. The least-square policy iteration (L_{SPI}) for policy evaluation is given as follows.

Algorithm 1 Approximate Policy Iteration With L_{STD}-Q for Policy Evaluation

Step: 1 Initialize the controller and system parameters K_0, N, T, σ_η^2, and μ.
Step: 2 collect the trajectory data D_t, using input $u_k^{(t)}$.
Step: 3 **for** $t = 0, \ldots, N-1$ **do**
Step: 4 $\hat{Q}_t = \mathrm{Proj}_\mu(L_{STD}\text{-}Q(\mathcal{D}_t, K_t))$.
Step: 5 $K_{t+1} = G(\hat{Q}_t)$
Step: 6 **end for**
Step: 7 return K_N.

In the above algorithm, K_0 is the stabilizing controller, N corresponds to the number of policy iterations, T is the rollout length for estimation, σ_η^2 is the exploration variance, and μ is the lower eigenvalue bound. The trajectory D_t corresponds to samples of form $\{(x_k^{(t)}, u_k^{(t)}, x_{k+1}^{(t)})\}_{k=1}^{T}$, and $u_k^{(t)} = K_0 x_k^{(t)} + \eta_k^{(t)}$, where $\eta_k^{(t)} \sim \mathcal{N}(0, \sigma_\eta^2 I)$. In step 4, $\mathrm{Proj}_\mu(\cdot) = \arg\min_{X = X^{\mathrm{T}}: X \succcurlyeq \mu \cdot I} \|X - \cdot\|_F$ is an Euclidean projection to set of lower bounded $\mu \cdot I$ symmetric matrices. The mapping $G(\cdot)$ in step 5 is a positive definite matrix of size $(n+d) \times (n+d)$ which returns $d \times n$ matrix as follows:

$$G\left(\begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^{\mathrm{T}} & Q_{22} \end{bmatrix}\right) = -Q_{22}^{-1} Q_{12}^{\mathrm{T}}. \tag{24}$$

IV. Experiment

The development process of RL-LQR for training and control implementation with various robotic systems to achieve trajectory tracking and balance control along a reference trajectory is shown in Fig. 2.

Practically, the control signals from the actuators are often constrained by the dead-zone phenomenon due to the physical limitation. In [52], the unknown dead-zone is considered and its uncertainty is compensated by an NN. Inspired by the work in [52], various control methods dealing with the dead-zone phenomenon are studied and reported [53], [54]. In [55], the control problem of uncertainties and dead-zones with a class of multi-input–multi-output (MIMO) underactuated systems is addressed by developing an adaptive controller which uses a fuzzy structure to approximate the model unknown parts based on the composite surfaces. Similarly in [56], the uncertainties in a discrete time nonlinear system transformed from a continuous nonlinear system are approximated using NNs which assist the RL in achieving adaptive tracking control.

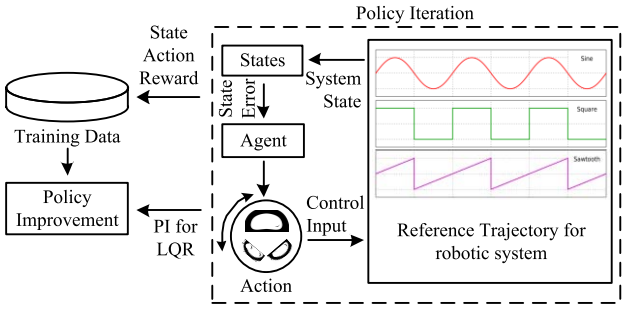

Fig. 2. Control policy learning for RL-LQR controller to achieve trajectory tracking with robotic systems.

Furthermore, in [57], the uncertain plant parameters in a robot manipulator are handled by developing an adaptive motion controller which tends to converge the position error to zero and ensures the closed-loop stability of the plant. In [58], the slope of dead-zone is defuzzified to a deterministic value, and the adaptive control scheme for a robot manipulator is designed via fuzzy logic. Besides the dead-zone factors, the actuators are also constrained by saturation due to the physical limitations of voltages, currents, flows, torques, and so on. These constraints can make the closed loop unstable. Hence, the literature in [59]–[61] considers the input constraints along with external disturbances. Further it is identified that all the above-mentioned studies only consider the dead-zone and the saturation independently of each other. Although these control techniques are popular in handling the effects due to the dead-zone and saturation, the selection of their control gains and the lack of uniform boundedness are the major drawbacks.

These drawbacks are further overcome by the developed RL-LQR approaches which achieve the boundedness for the closed-loop control with the LSTD approach. From the framework of the RL-LQR shown in Fig. 2, the agent which involves an actor and a critic generates the desired control input to the actuator from the actor agent and the corresponding output is evaluated by the critic agent before processing it to the actuator. Furthermore, the actor agent updates its output based on the evaluation processed by the critic agent, and the critic agent is supported by the boundedness provided by the LSTD to compensate the dead-zone and saturation with the actuator in the closed-loop control structure. Furthermore, the LSTD also aids the RL-LQR to converge at its global optimum while solving the control problem with practical systems.

In addition to the above, there are two sets of parameters that need to be handled while performing the experiments with linear systems: the design parameters corresponding to both helicopter and balancer robot system and the hyperparameters corresponding to the controller for both the environments. The design parameters involve the link lengths between the main and tail rotors in case of helicopter system and between the actuators for a ball balancer system. They define the morphology of the robot and are constant over time. Furthermore, a model hyperparameter is a configuration that is external to the model and whose value cannot be estimated from data. In this experimental condition, the model hypermeters are the number of maximum episodes that are required to be set during the training process, the step size $\alpha_0 \in \{0.01, 0.1, 1\}$, and the learning rates that are needed for the actor and critic agents. These hyperparameters are generally tuned using the grid search or a random search approach such that the algorithm converges fast and efficient while training the states and tuning the controller.

Since all the hypermeters used with the proposed approach are majorly focused in achieving optimal convergence with less computation and time complexity, the tradeoff for selection of these parameters is majorly interdependent on these outcomes. Furthermore, the tradeoff between these outcomes can be achieved during controller development through the learning process in the simulation environment.

A. Model-Free Control for Trajectory Tracking

The problem of trajectory tracking in unmanned vehicles is realized by modeling the working phenomenon of the helicopter considering the pitch and yaw motion. The main and tail rotors of the helicopter are responsible for pitch and yaw motion around the X- and Z-axes, respectively. The pitch motion is achieved by generating the vertical thrust, whereas the yaw motion is achieved by generating the horizontal movement. These rotors are operated by voltage inputs which return torque τ_1 for the main rotor and torque τ_2 for the tail rotor. The pitch motion in the vertical plane is termed as pitch angle ψ and yaw motion in the horizontal plane is termed as yaw angle ξ. The complete system modeling is divided into two parts. In the first part, the mathematical modeling of the setup can be explained based on dc motor modeling embedded with main and tail rotors, and in second part, the mechanics of the system is explained based on the vertical and horizontal planes. Furthermore, to characterize the mechanical and electrical models of the motor, a set of differential equations are formulated as follows.

For main rotor and vertical movement

$$\frac{d}{dt}\psi = \dot{\psi} \tag{25}$$

$$\frac{d}{dt}\dot{\psi} = \frac{a_1}{I_1}\tau_1^2 + \frac{b_1}{I_1}\tau_1 - \frac{M_g}{I_1}\sin\dot{\psi} + \frac{0.0326}{2I_1}\sin(2\psi)\left(\dot{\xi}\right)^2 - \frac{B_{1\psi}}{I_1}\dot{\psi} - \frac{k_{gy}}{I_1}a_1\cos(\psi)\dot{\xi}\tau_1^2 - \frac{k_{gy}}{I_1}b_1\cos(\psi)\dot{\xi}\tau_1 \tag{26}$$

$$\frac{d}{dt}\tau_1 = -\frac{T_{m1}}{m2}\tau_1 + \frac{k_1}{T_{m2}}u_1. \tag{27}$$

For tail rotor and horizontal movement

$$\frac{d}{dt}\xi = \dot{\xi} \tag{28}$$

$$\frac{d}{dt}\dot{\xi} = \frac{a_2}{I_2}\tau_2^2 + \frac{b_2}{I_2}\tau_2 - \frac{B_{1\xi}}{I_2}\dot{\xi} - \frac{1.75}{I_2}k_c a_1\tau_1^2 - \frac{1.75}{I_2}k_c b_1\tau_1 \tag{29}$$

$$\frac{d}{dt}\tau_2 = -\frac{T_{t1}}{T_{t2}}\tau_2 + \frac{k_2}{T_{t2}}u_2 \tag{30}$$

where a_1, a_2, b_1, and b_2 are the static characteristic parameters, the tail and main rotor moment of inertia are given by I_1 and I_2, respectively, the gravity momentum is defined by M_g, the main and tail motor's fraction momentum function parameters are given by $B_{1\psi}$ and $B_{1\xi}$, respectively, the denominator

parameters are T_{m1}, T_{m2}, T_{t1}, and T_{t2}, the motor gains are k_1 and k_2, and the momentum gain of cross reaction is given as k_c.

Considering the above differential equations, the state, input, and output vectors for the physical parameters of the system are given by

$$x = \begin{bmatrix} \psi \\ \dot{\psi} \\ \xi \\ \dot{\xi} \\ \tau_1 \\ \tau_2 \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \quad y = \begin{bmatrix} \psi \\ \xi \end{bmatrix}. \tag{31}$$

These vectors can represent the two-degree motion of helicopter as a continuous time linear system, which can be given by

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \tag{32}$$

where, A, B, C, and D can be obtained from (1)–(6) as

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{M_g}{I_1} & -\frac{B_{1\psi}}{I_1} & 0 & 0 & \frac{b_1}{I_1} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{B_{1\psi}}{I_2} & -1.75k_c\frac{b_1}{I_2} & \frac{b_2}{I_2} \\ 0 & 0 & 0 & 0 & -\frac{T_{m1}}{T_{m2}} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{T_{t1}}{T_{t2}} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{k_1}{T_{m2}} & 0 \\ 0 & \frac{k_2}{T_{t2}} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

These vectors can represent the two-degree motion of the helicopter as a continuous time linear system. Furthermore, to achieve control development for unmanned helicopter, the continuous time linear system is expressed as a linear discrete time system in time invariant state space. The developed model-free control is implemented for achieving trajectory tracking in unmanned helicopters using output feedback version of on-policy approach. The selected algorithm is implemented with the theoretical system model of unmanned helicopter for generating data required for learning the algorithm. The data in this experiment correspond to various trajectories that define the states of the helicopter are obtained from the simulated model. The responses of the helicopter pitch and yaw motors for various trajectories are shown in Fig. 3.

Initially, a reference point is developed to implement the output feedback version of the algorithm. Furthermore, the developed controller is modeled for a real-time system using

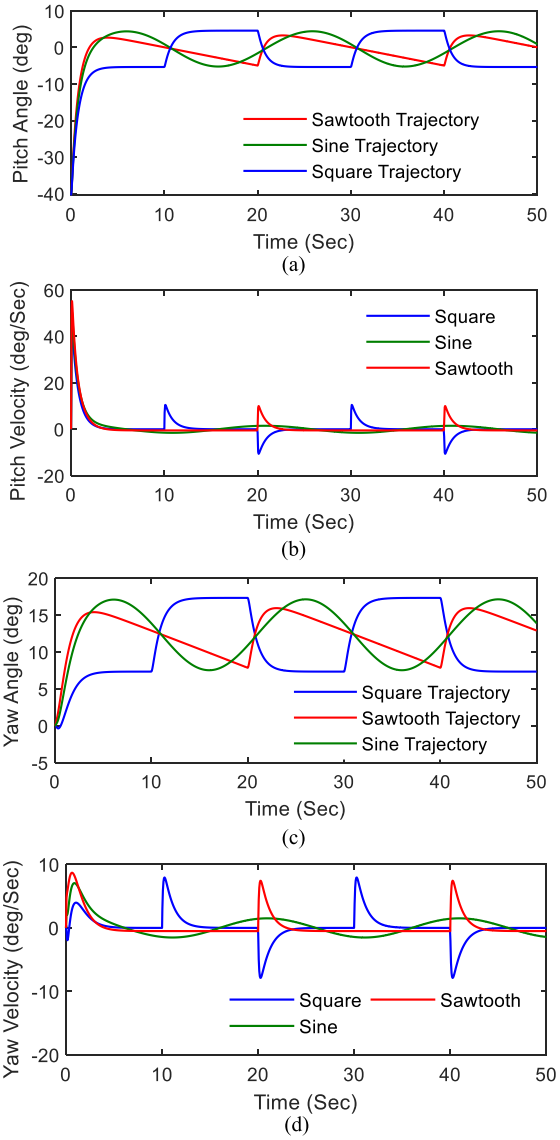

Fig. 3. Response of helicopter pitch and yaw motors for different trajectories. (a) Pitch angle. (b) Pitch velocity. (c) Yaw angle. (d) Yaw velocity.

Quanser two degrees of freedom helicopter [62]. The experimental setup of the helicopter model considered for this experiment is shown in Fig. 4. The learning process is associated with the LQR controller implemented with the system. Furthermore, the errors of both the pitch and yaw angles are computed by comparing them with the reference developed. The experiment is performed using both full state and output measurements of the model.

To assess the performance of the developed controller, the pitch and yaw characteristics of RL-LQR are compared with an adaptive fuzzy controller (AFC) [63] and a classical LQR controller. A sine reference trajectory with amplitude 5 is considered for assessing the pitch and yaw path tracking. The RL algorithm is trained for 30 episodes with 100 step size and 0.001 learning rate. The corresponding results are shown in Fig. 5.

The results depict the tracking response of the helicopter with RL-based state feedback controller on simulation and real-time system with reference to the desired trajectory for

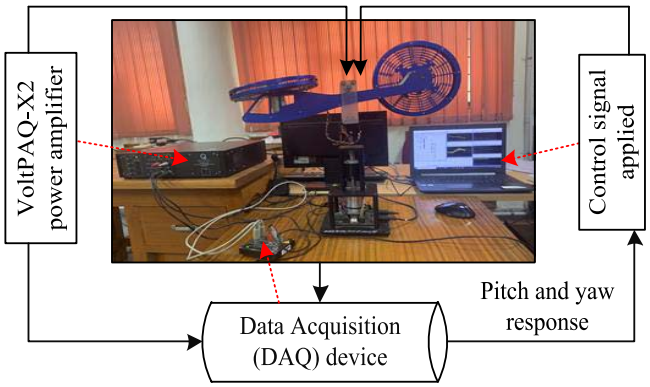

Fig. 4. Experimental setup of a helicopter system with the main and tail rotors.

TABLE I

ROOT MEAN SQUARE ERROR OF PITCH AND YAW OUTPUTS FOR LQR AND RL-LQR CONTROLLER

Controllers Real-Time	Pitch	Yaw
LQR	7.4012	5.4324
AFC	7.137	5.2173
RL-LQR	6.8636	4.9981

TABLE II

STEP RESPONSE CHARACTERISTICS OF PITCH AND YAW RESPONSE OF REAL-TIME SYSTEM

Controller	Pitch Control Response			Yaw Control Response		
	t_s	M_P	e_{ss}	t_s	M_P	e_{ss}
LQR	5.07	0.981	3.19	4.86	169	5.15
AFC	4.92	0.83	2.74	4.55	70.1	4.68
RL-LQR	4.67	0.559	1.7	4.19	30.6	3.22

pitch angle movement [Fig. 5(a)], pitch velocity [Fig. 5(b)], yaw angle movement [Fig. 5(c)], and yaw velocity [Fig. 5(d)]. The control objective of the work is to track the desired trajectory with minimum settling time. The pitch angle trajectory reaches the desired trajectory at 1 s and settles at 6 s and yaw angle settles within 5 s. The pitch velocity reduced to 16°/s when subjected to real-time environments but sufficient enough for pitch rise to balancing and yaw velocity increases to 15°/s due to cross coupling effect. Furthermore, the convergence of RL for operating the LQR to achieve efficient trajectory tracking is shown in Fig. 6. The total training is carried up to 100 episodes to verify whether there is a deviation during a long run with the data training process. The reward convergence for the application of RL with the helicopter system is achieved around 15–20 episodes.

The root mean square error for the controller action on real-time system is shown in Table I. The settling time (t_s), peak overshoot (M_P), and steady-state error (e_{ss}) are minimum in case of RL-LQR which are 4.67 s, 0.559%, and 1.7 cm for pitch control and 4.19 s, 30.6%, and 3.22 cm, respectively, for yaw control as shown in Table II. As the behavior policy of the real-time controller is calculated using the discrete linear model of the system, the learning algorithm converges quickly to settle the helicopter motion with respect to the defined trajectory. The results proved the effectiveness of the model-free controller using RL for trajectory tracking of unmanned helicopters.

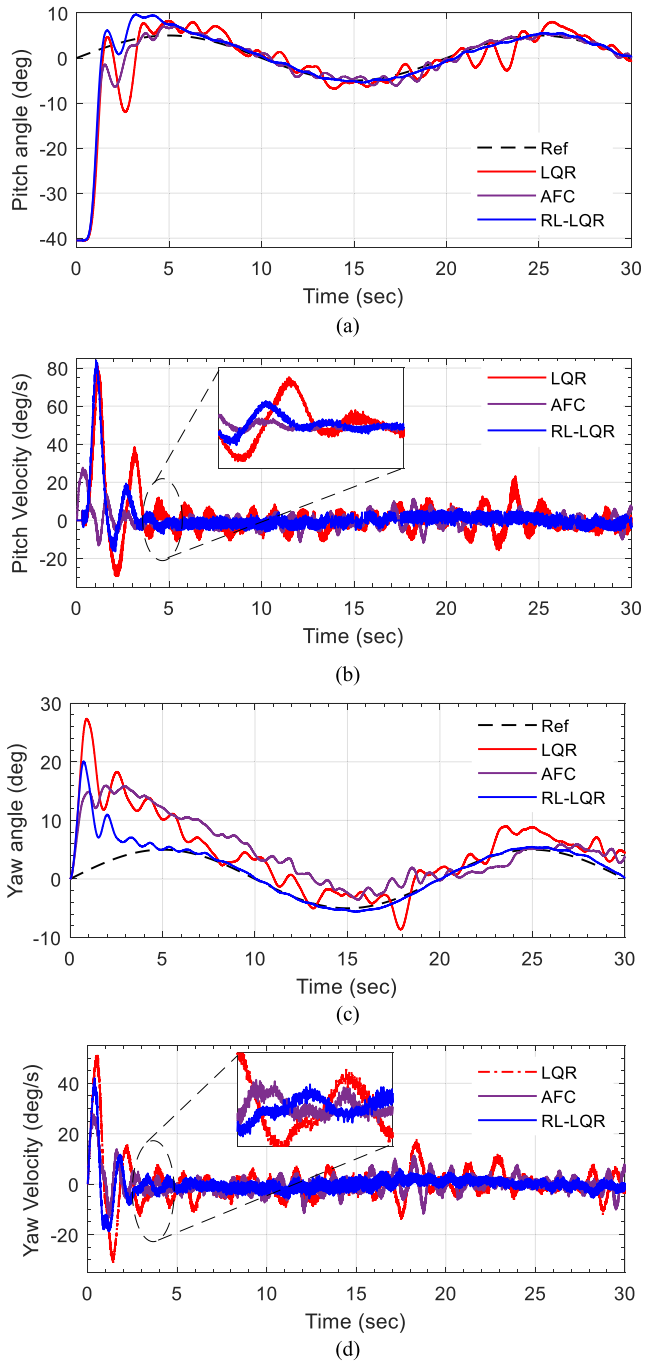

Fig. 5. Output responses of the real-time experiment of the helicopter system for sine trajectory. (a) Pitch angle for sine trajectory. (b) Pitch velocity for sine trajectory. (c) Yaw angle for sine trajectory. (d) Yaw velocity for sine trajectory.

Controller Fitness: To analyze the fitness of the control actions implemented using the robotic systems to achieve trajectory tracking, the integral of the square error (ISE) [64] is estimated as

$$\text{ISE} = \int e^2(t)dt \tag{33}$$

where e is the tracking error between the reference and measured trajectories.

The tracking error of all the control actions RL-LQR, AFC, and LQR applied on the helicopter system to follow

TABLE III

TRACKING ERROR COMPARISON FOR SINE TRAJECTORY OPERATION OF REAL-TIME HELICOPTER SYSTEM

Controller	Pitch Angle	Yaw Angle
RL-LQR	7.91e+03	5.94e+03
AFC	1.18e+04	7.61e+03
LQR	4.34e+04	2.18e+04

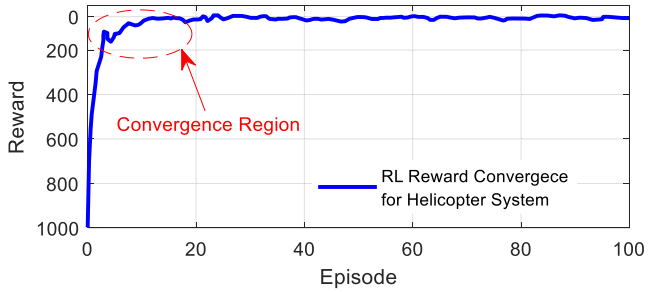

Fig. 6. Reward convergence of RL for the helicopter system.

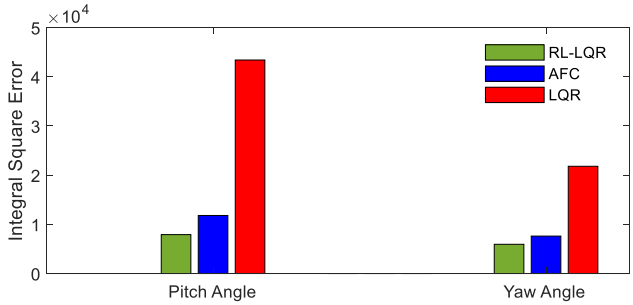

Fig. 7. Tracking error comparison of RL-LQR and LQR controllers.

a sine trajectory in real-time experiment are measured, and the ISE is estimated. The corresponding results are illustrated in Fig. 7 and Table III.

B. Model-Free Control for Balancing

The problem of balancing control in robotic systems is realized by modeling a ball balancer system operating with two degrees of freedom. This represents a typical benchmark problem for testing the operation of the control action in achieving, position tracking, balancing control, and visual servo control. For a general ball balancer system, the coordinates have similar servo dynamics. This makes the modeling of the system easier as the control through one direction (X-axis) represents a similar approach for the control through the alternative direction (Y-axis). The nonlinear equation of motion for X-axis control of ball balancer is obtained from the Quanser ball balancer model [65] as follows:

$$\ddot{x}_d(t) = \frac{2M_{\text{ball}}\, g \beta_l R_{\text{arm}} R_{\text{ball}}^2}{L_{\text{plate}}\left(M_{\text{ball}} R_{\text{ball}}^2 + J_b\right)} \tag{34}$$

$$\ddot{\beta}_l(t) = \frac{(K_m u(t) - \dot{\beta}(t))}{\tau} \tag{35}$$

where x_d is the position of the ball, β_l is the load angle, M_{ball} is the mass of the ball, J_b is the moment of inertia of the ball, α is the plate angle, R_{ball} is the ball radius, the distance between the output gear shaft and the servo motor is R_{arm}, g is the universal gravitational constant, the plate length is defined by L_{plate}, K_m, and τ are the motor speed and time constants, respectively.

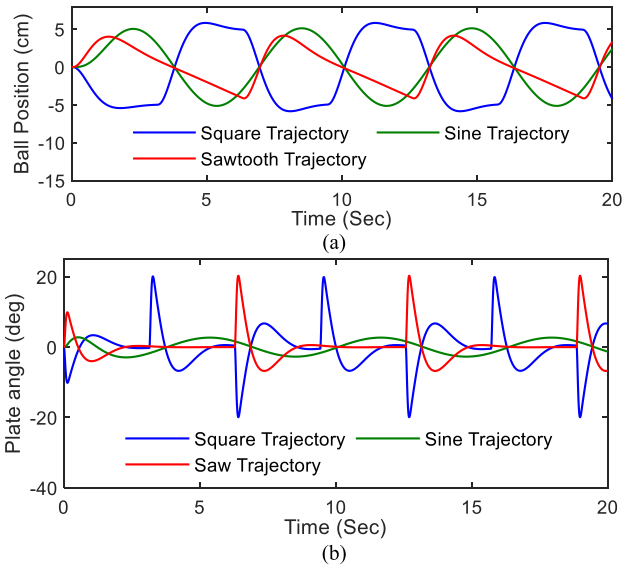

Fig. 8. Response of the ball balancer for ball movement in different trajectories. (a) Ball position. (b) Plate angle.

The transfer function of the ball position for an input β_l and output x_d is given by

$$P_b(s) = \frac{x_d(s)}{\beta_l(s)} = \frac{K_b}{s^2} \tag{36}$$

where $K_b = (2M_{\text{ball}}\, g \beta_l R_{\text{arm}} R_{\text{ball}}^2 / L_{\text{plate}}(M_{\text{ball}} R_{\text{ball}}^2 + J_b))$.

Furthermore, the transfer function of plate angle control with the servo motor is given as

$$P_s(s) = \frac{\beta_l(s)}{V_m(s)} = \frac{K_g}{s(\tau s + 1)} \tag{37}$$

where K_g is the static gain.

For a cascaded connection between the ball balancer module and the servo motors, the transfer function is given as

$$P(s) = P_s(s) P_b(s) = \frac{\beta_l(s)}{V_m(s)} = \frac{K_g}{s^3(\tau s + 1)}. \tag{38}$$

This further aids in estimating the state-space representation of the system as

$$\begin{bmatrix} \dot{x}_d(t) \\ \ddot{x}_d(t) \\ \dot{\beta}_l(t) \\ \ddot{\beta}_l(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & K_b & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{\tau} \end{bmatrix} \begin{bmatrix} x_d(t) \\ \dot{x}_d(t) \\ \beta_l(t) \\ \dot{\beta}_l(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K_m}{\tau} \end{bmatrix} u(t). \tag{39}$$

To achieve the control for the ball balancer system, the action of LQR is adapted. Furthermore, the developed learning function is adapted using full state data as discussed in Section III. Initially, the system measurements are obtained by simulating the model discussed above using stabilizing behavior policy. The system is learned with different balancing situations by varying the position of the ball with respect to different trajectories as shown in Fig. 8. To ensure that the system is being operated within the affine control region without unbalancing, an exploratory noise is added.

The simulation results in Fig. 8 display the data corresponding to the ball position and plate angle for various trajectories. Furthermore, the real-time implementation of developed

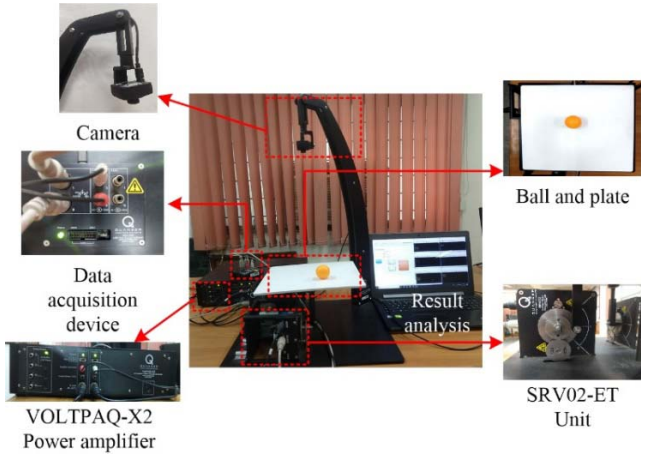

Fig. 9. Laboratory setup of the balancer robotic system to operate with model-free control.

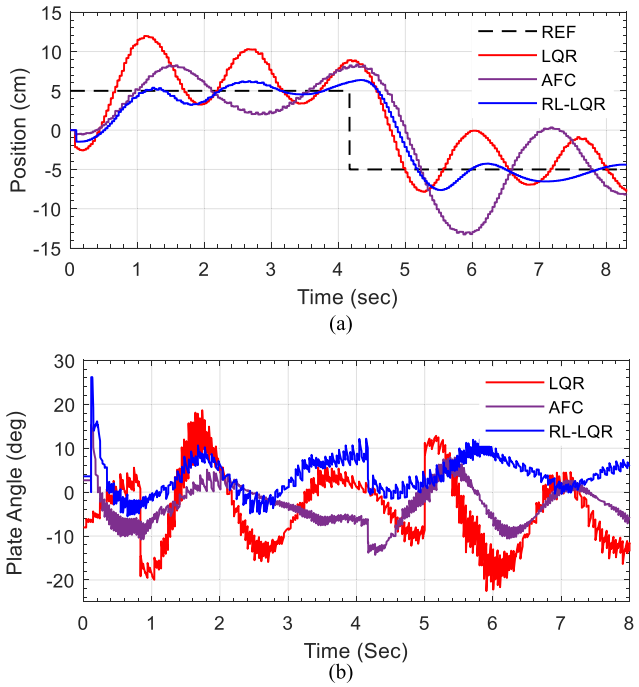

Fig. 10. Output responses of the real-time ball balancer system for square trajectory. (a) Ball position. (b) Plate angle.

learning function is achieved with the help of the Quanser ball balancer system shown in Fig. 9. The details of the Quanser ball balancer system are established in [66]. Using larger exploration noise, the performance of the ball balancer system for achieving balancing control is observed under square trajectory. The ball position and plate angle for real-time operation of the ball balancer system with RL-LQR, AFC, and classical LQR controller are shown in Fig. 10.

From Fig. 10(a), it is observed that the ball position follows the reference trajectory with fewer oscillations for the RL-LQR controller which makes the ball movement to slow down and achieve system stability. Subsequently, the plate angle variation for RL-LQR in Fig. 10(b) is initially between $+5^\circ$ and -15°, and after the falling edge of the reference trajectory the plate angle varies between -18° and $+18^\circ$ due to the movement of the ball to stabilize around the trajectory. In both the cases, the response of RL-LQR is better compared with

TABLE IV

STEP RESPONSE CHARACTERISTICS OF BALL POSITION CONTROL IN REAL-TIME SYSTEM

Controllers	Peak time (t_p)	Settling time (t_s)	Peak overshoot (M_p)	Steady state error (e_{ss})
LQR	2.4 sec	2.76 sec	22.9%	1.151 cm
AFC	2.1 sec	2.62 sec	18.3%	1.013 cm
RL-LQR	1.62 sec	2.09 sec	12.4%	0.487 cm

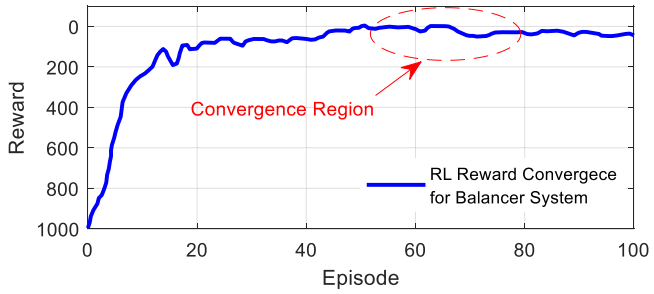

Fig. 11. Reward convergence of RL for the balancer system.

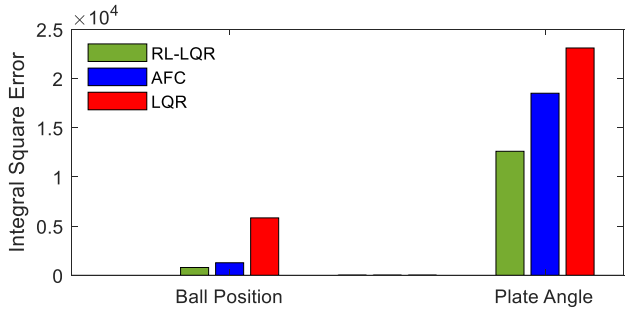

Fig. 12. Tracking error comparison of RL-LQR and LQR controllers for ball position and plate angle.

the AFC and classical LQR controller. The convergence of RL for operating the LQR to achieve efficient trajectory tracking is shown in Fig. 11. The convergence is partially achieved around 50–60 episodes and later on varies around the zero value to settle around 70–80 episodes.

Furthermore, the superiority of the developed controller and its application with the balancer robotic system are assessed by estimating the step response characteristics for position control. The corresponding results are given in Table IV and compared with the other classical and adaptive control techniques applied with the balancer system. The results identified that the response of improved RL-LQR is better for all the measured characteristics.

The ball position tracking error of all the control actions RL-LQR, AFC, and LQR applied on the balancer robotic system to follow a square trajectory in real-time experiment are measured, and the ISE is estimated. The corresponding results are illustrated in Fig. 12 and Table V.

Furthermore, the complexity analysis is carried out in terms of computational complexity and time cost for the training process of RL to optimize the action of LQR.

C. Complexity Analysis

1) Computation Complexity: The LSTD approach is widely known to guarantee convergence with any given linear/nonlinear function approximator for an appropriate step

TABLE V

TRACKING ERROR COMPARISON FOR REFERENCE TRAJECTORY OPERATION OF REAL-TIME BALANCING CONTROL SYSTEM

Controller	Ball Position	Plate Angle
RL-LQR	7.89e+02	1.26e+04
AFC	1.27e+03	1.85e+04
LQR	5.83e+03	2.31e+04

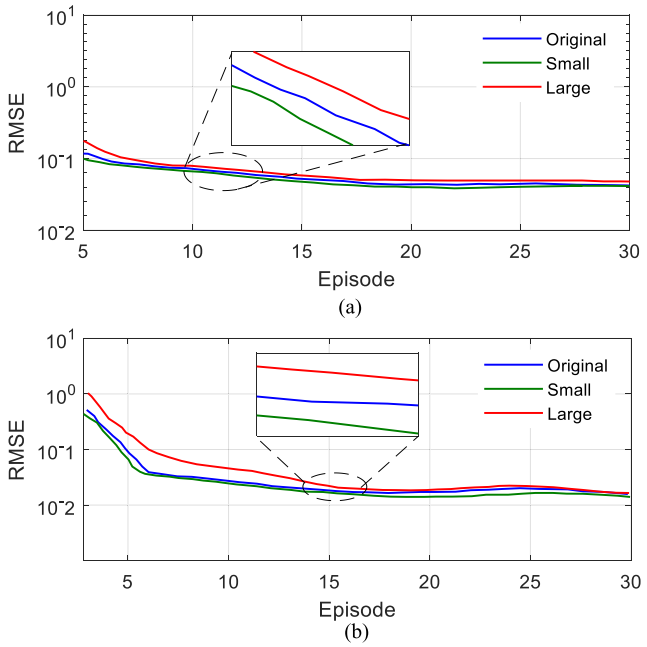

Fig. 13. Comparison of computation complexities for RL training with original, small, and large datasets. (a) Computational complexity for helicopter system operation. (b) Computational complexity for balancer system operation.

size [67]. Furthermore, it is also established that the learning process with the LSTD approach is computationally inexpensive and requires $O(n)$ computation per time step. Here, the term n corresponds to the number of state values used with the approximator. To analyze the computational complexity of the developed iterative learning process, the results of three different problem sizes are compared. The problem sizes include three states (small problem), nine states (original problem), and 27 states (large problem) for both the robotic systems considered in the experiment. As the problem-solving approach is an iterative approach, it trains the data of all the states for 30 episodes starting from state N and terminating at state 0. The step size α used in the experiments is estimated as

$$\alpha_t = \alpha_0 \frac{N_0 + 1}{N_0 + \text{Episode}}. \tag{40}$$

N_0 and α_0 are selected for the least-squared temporal difference by finding the best parameters in the set $\alpha_0 \in \{0.01, 0.1, 1\}$ and $N_0 \in \{100, 1000, 10^6\}$. Furthermore, the computational complexity analysis is developed based on the best set of parameters for each problem. Besides, to update only one component at each time step, the update parameter for the RL-LQR is set to one.

The performance of all three problem sizes is averaged over 30 episodes for both the systems as shown in Fig. 13. For each episode, the algorithm follows the given trajectory data.

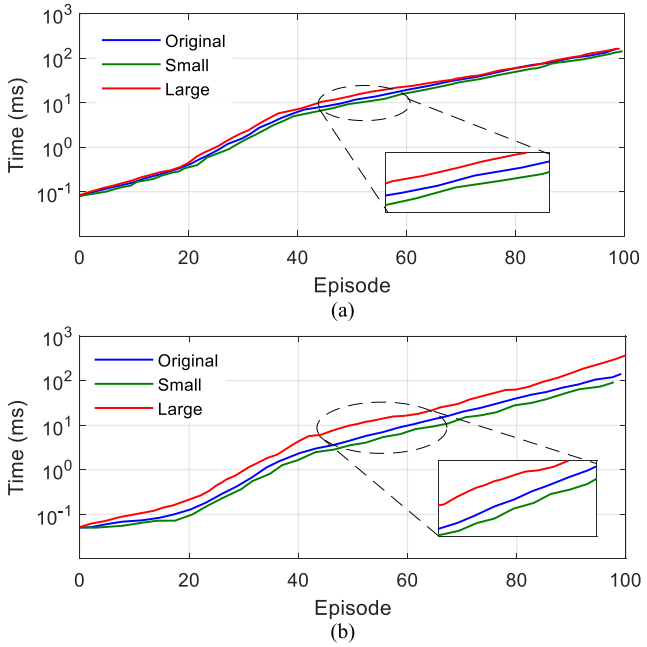

Fig. 14. Comparison of time complexities for RL training with original, small, and large datasets. (a) Time complexity for helicopter system operation. (b) Time complexity for balancer system operation.

As the complexity of the problem increases, the performance of the developed algorithm follows very closely to the original problem error. Furthermore, the relative difference for the large problem is most dramatic, as all the data efficiency of the least-squares approaches is more pronounced in larger problems.

2) Time Cost Analysis: The time cost analysis is achieved by estimating the time complexity which varies with the size of the input array. If there are n state values, and for any given state s, $\phi(s)$ has at most k nonzero elements, then the LSTD-based RL algorithm is $O(mn + k^2)$ per time step [68]. Based on this statement, the sparse vector algebra is used with all the three problem sizes to take advantage of the sparsity of $\phi(s)$ and state transition matrix. Furthermore, the analysis is performed by timing the experiments on an Intel i7 processor with clock speed ranging from 2.9 to 4.2 GHz with the increasing number of states. The results were consistent with the timing analyses, and the total running time for 100 episodes is shown for both the robotic systems in Fig. 14 on a log scale.

All the above results proved the effectiveness of the model-free controller using RL for trajectory tracking and balancing control of robotic systems.

V. CONCLUSION

In this article, an LSTD learning algorithm is adapted for developing a model-free controller for robotic applications. The Q-function of RL was used with the learning algorithm, and a model-free state feedback controller is developed by establishing LQR as a baseline controller. The classical least-square policy iteration technique is used to establish the boundary conditions for complexities incurred by the learning algorithm. The developed model-free controller is tested for trajectory tracking and balancing control of robotic systems.

In both the cases, the controller generates the required action by learning the various states of the system through simulation. The performance of the developed controllers is assessed by performing real-time experimental analysis. The step response characteristics of the controller for the helicopter system showed a settling time of 4.67 s, peak overshoot of 0.559%, and steady-state error of 1.7 cm for pitch control and 4.19 s, 30.6%, and 3.22 cm for yaw control. Similarly, the step response characteristics of the controller for the helicopter system showed a peak time of 1.62 s, settling time of 2.09 s, peak overshoot of 12.4%, and steady-state error of 0.487 cm for the ball position control. Furthermore, it is identified that the RL-LQR controller under both the experimental conditions converges approximately at zero under 30 episodes and has a least computational and time complexity. Through the results, it is identified that the developed approach is highly efficient in terms of convergence speed, and computational and time complexity. Furthermore, it also does not require additional computation power and has similar performance when tested with different machines.

Furthermore, the work can be extended to achieve policy optimization using Markovian Jump for RL-based control algorithms. Few of these applications with output feedback control problem and state estimation problem are discussed in [69] and [70]. This approach will be targeted at implementation with large-scale systems where the computational complexity grows as the system and input size increase. In addition, the comparison between the developed algorithm and the suggested or emerging RL approaches will be carried out as a potential review analysis.

REFERENCES

[1] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, and J. Plosila, "Swarms of unmanned aerial vehicles—A survey," *J. Ind. Inf. Integr.*, vol. 16, Dec. 2019, Art. no. 100106, doi: 10.1016/j.jii.2019.100106.

[2] B. K. Patle, A. Pandey, D. R. K. Parhi, and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technol.*, vol. 15, pp. 582–606, Aug. 2019, doi: 10.1016/j.dt.2019.04.011.

[3] P. E. I. Pounds and A. M. Dollar, "Stability of helicopters in compliant contact under PD-PID control," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1472–1486, Dec. 2014, doi: 10.1109/TRO.2014.2363371.

[4] J. P. Ortiz, L. I. Minchala, and M. J. Reinoso, "Nonlinear robust H-infinity PID controller for the multivariable system quadrotor," *IEEE Latin Amer. Trans.*, vol. 14, no. 3, pp. 1176–1183, Mar. 2016, doi: 10.1109/TLA.2016.7459596.

[5] Z. Li, J. Yu, X. Xing, and H. Gao, "Robust output-feedback attitude control of a three-degree-of-freedom helicopter via sliding-mode observation technique," *IET Control Theory Appl.*, vol. 9, no. 11, pp. 1637–1643, Jul. 2015, doi: 10.1049/iet-cta.2014.1068.

[6] F. Chen, R. Jiang, K. Zhang, B. Jiang, and G. Tao, "Robust backstepping sliding mode control and observer-based fault estimation for a quadrotor UAV," *IEEE Trans. Ind. Electron.*, vol. 63, no. 8, pp. 5044–5056, Aug. 2016, doi: 10.1109/TIE.2016.2552151.

[7] M. Arbulu, A. Padilla, and F. Ramirez, "Geometric balancing control of humanoid robots," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2013, pp. 2136–2141, doi: 10.1109/ROBIO.2013.6739785.

[8] B.-J. Lee, Y.-D. Kim, and J.-H. Kim, "Balance control of humanoid robot for HuroSot," *IFAC Proc. Volumes*, vol. 38, no. 1, pp. 215–220, 2005, doi: 10.3182/20050703-6-CZ-1902.02088.

[9] Y. Wang, Q. Zhu, R. Xiong, and J. Chu, "Standing balance control for position control-based humanoid robot," *IFAC Proc. Volumes*, vol. 46, no. 20, pp. 429–436, 2013, doi: 10.3182/20130902-3-CN-3020.00064.

[10] D. Zhang and J. J. P. Tsai, *Machine Learning Applications in Software Engineering*, vol. 16. Singapore: World Scientific, 2005.

[11] Y.-H. Wang, T.-H.-S. Li, and C.-J. Lin, "Backward Q-learning: The combination of Sarsa algorithm and Q-learning," *Eng. Appl. Artif. Intell.*, vol. 26, no. 9, pp. 2184–2193, Oct. 2013, doi: 10.1016/j.engappai.2013.06.016.

[12] M. Breyer, F. Furrer, T. Novkovic, R. Siegwart, and J. Nieto, "Comparing task simplifications to learn closed-loop object picking using deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1549–1556, Apr. 2019, doi: 10.1109/LRA.2019.2896467.

[13] G. Chalvatzaki, X. S. Papageorgiou, P. Maragos, and C. S. Tzafestas, "Learn to adapt to human walking: A model-based reinforcement learning approach for a robotic assistant rollator," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3774–3781, Oct. 2019, doi: 10.1109/LRA.2019.2929996.

[14] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2096–2103, Oct. 2017, doi: 10.1109/LRA.2017.2720851.

[15] N. J. Cho, S. H. Lee, I. H. Suh, and H.-S. Kim, "Relationship between the order for motor skill transfer and motion complexity in reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 293–300, Apr. 2019, doi: 10.1109/LRA.2018.2889026.

[16] A. Viseras and R. Garcia, "DeepIG: Multi-robot information gathering with deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 3059–3066, Jul. 2019, doi: 10.1109/LRA.2019.2924839.

[17] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 392–399, doi: 10.1109/ITSC.2014.6957722.

[18] K. Lee, S. Choi, and S. Oh, "Sparse Markov decision processes with causal sparse Tsallis entropy regularization for reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1466–1473, Jul. 2018, doi: 10.1109/LRA.2018.2800085.

[19] G. Sartoretti *et al.*, "PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2378–2385, Jul. 2019, doi: 10.1109/LRA.2019.2903261.

[20] C. Greatwood and A. G. Richards, "Reinforcement learning and model predictive control for robust embedded quadrotor guidance and control," *Auto. Robots*, vol. 43, no. 7, pp. 1681–1693, Oct. 2019, doi: 10.1007/s10514-019-09829-4.

[21] S. Gangapurwala, A. Mitchell, and I. Havoutis, "Guided constrained policy optimization for dynamic quadrupedal robot locomotion," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3642–3649, Apr. 2020, doi: 10.1109/LRA.2020.2979656.

[22] B. Li and Y. Wu, "Path planning for UAV ground target tracking via deep reinforcement learning," *IEEE Access*, vol. 8, pp. 29064–29074, 2020, doi: 10.1109/ACCESS.2020.2971780.

[23] 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," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4423–4438, Nov. 2019, doi: 10.1109/TAC.2019.2905215.

[24] Q. Zhang and D. Zhao, "Data-based reinforcement learning for nonzero-sum games with unknown drift dynamics," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2874–2885, Aug. 2019, doi: 10.1109/TCYB.2018.2830820.

[25] W. W. Bai, T. S. Li, and S. C. Tong, "NN reinforcement learning adaptive control for a class of nonstrict-feedback discrete-time systems," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4573–4584, Nov. 2020, doi: 10.1109/TCYB.2020.2963849.

[26] S. J. Bradtke, "Reinforcement learning applied to linear quadratic regulation," in *Proc. 5th Adv. Neural Inf. Process. Syst.*, 1993, pp. 295–302.

[27] E. Abramova, "Combining reinforcement learning and optimal control for the control of nonlinear dynamical systems," Ph.D. dissertation, Imperial College London, London, U.K., May 2015, pp. 1–182, doi: 10.25560/39968.

[28] S. A. A. Rizvi and Z. Lin, "Reinforcement learning-based linear quadratic regulation of continuous-time systems using dynamic output feedback," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4670–4679, Nov. 2020, doi: 10.1109/TCYB.2018.2886735.

[29] S. J. Bradtke, "Incremental dynamic programming for on-line adaptive optimal control," Dept. Comput. Sci., Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. 94-62, 1995.

[30] M. Palanisamy, H. Modares, F. L. Lewis, and M. Aurangzeb, "Continuous-time Q-learning for infinite-horizon discounted cost linear quadratic regulator problems," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 165–176, Feb. 2015, doi: 10.1109/TCYB.2014.2322116.

[31] M. Fazel, R. Ge, S. M. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 1467–1476.

[32] D. Malik, A. Pananjady, K. Bhatia, K. Khamaru, P. L. Bartlett, and M. J. Wainwright, "Derivative-free methods for policy optimization: Guarantees for linear quadratic systems," 2018, *arXiv:1812.08305*.

[33] H. Mania, A. Guy, and B. Recht, "Simple random search provides a competitive approach to reinforcement learning," 2018, *arXiv:1803.07055*.

[34] A. Vemula, W. Sun, and J. A. Bagnell, "Contrasting exploration in parameter and action space: A zeroth-order optimization perspective," 2019, *arXiv:1901.11503*.

[35] A. Lazaric, M. Ghavamzadeh, and R. Munos, "Finite-sample analysis of least-squares policy iteration," *J. Mach. Learn. Res.*, vol. 13, pp. 3041–3074, Oct. 2012.

[36] D. E. Miller and M. Rossi, "Simultaneous stabilization with near optimal LQR performance," *IEEE Trans. Autom. Control*, vol. 46, no. 10, pp. 1543–1555, Oct. 2001, doi: 10.1109/9.956050.

[37] A. Antos, C. Szepesvári, and R. Munos, "Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path," *Mach. Learn.*, vol. 71, no. 1, pp. 89–129, Apr. 2008, doi: 10.1007/s10994-007-5038-2.

[38] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 4th ed. Belmont, MA, USA: Athena Scientific, 2017.

[39] J. N. Tsitsiklis and B. Van Roy, "Feature-based methods for large scale dynamic programming," *Mach. Learn.*, vol. 22, no. 1, pp. 59–94, Mar. 1996, doi: 10.1007/BF00114724.

[40] S. J. Bradtke and A. G. Barto, "Linear least-squares algorithms for temporal difference learning," *Mach. Learn.*, vol. 22, nos. 1–3, pp. 33–57, Jan. 1996, doi: 10.1007/BF00114723.

[41] Y. Abbasi-Yadkori, N. Lazic, and C. Szepesvari, "Model-free linear quadratic control via reduction to expert prediction," 2018, *arXiv:1804.06021*.

[42] Y. Abbasi-Yadkori, D. Pal, and C. Szepesvari, "Online least squares estimation with self-normalized processes: An application to bandit problems," 2011, *arXiv:1102.2670*.

[43] M. Simchowitz, H. Mania, S. Tu, M. I. Jordan, and B. Recht, "Learning without mixing: Towards a sharp analysis of linear system identification," 2018, *arXiv:1802.08334*.

[44] K. Schäcke, "On the Kronecker product," M.S. thesis, Univ. Waterloo, Waterloo, ON, Canada, 2013.

[45] H. Zhang and F. Ding, "On the Kronecker products and their applications," *J. Appl. Math.*, vol. 2013, pp. 1–8, Jun. 2013, doi: 10.1155/2013/296185.

[46] V. I. Bogachev, *Gaussian Measures*, vol. 62. Providence, RI, USA: American Mathematical Society, 1998.

[47] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "Regret bounds for robust adaptive control of the linear quadratic regulator," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, Dec. 2018, pp. 4192–4201.

[48] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000. [Online]. Available: http://infoscience.epfl.ch/record/233814

[49] B. Lincoln and A. Rantzer, "Relaxing dynamic programming," *IEEE Trans. Autom. Control*, vol. 51, no. 8, pp. 1249–1260, Aug. 2006, doi: 10.1109/TAC.2006.878720.

[50] H. Lee and Y. Lim, "Invariant metrics, contractions and nonlinear matrix equations," *Nonlinearity*, vol. 21, no. 4, pp. 857–878, Apr. 2008, doi: 10.1088/0951-7715/21/4/011.

[51] D. P. Bertsekas, "Value and policy iterations in optimal control and adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 500–509, Mar. 2017, doi: 10.1109/TNNLS.2015.2503980.

[52] R. R. Selmic and F. L. Lewis, "Deadzone compensation in motion control systems using neural networks," in *Proc. IEEE Int. Conf. Control Appl.*, vol. 1, Sep. 1998, pp. 288–292, doi: 10.1109/CCA.1998.728426.

[53] A. H. Tahoun, "A new online delay estimation-based robust adaptive stabilizer for multi-input neutral systems with unknown actuator nonlinearities," *ISA Trans.*, vol. 70, pp. 139–148, Sep. 2017, doi: 10.1016/j.isatra.2017.07.012.

[54] Q. Zhou, S. Zhao, H. Li, R. Lu, and C. Wu, "Adaptive neural network tracking control for robotic manipulators with dead zone," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3611–3620, Dec. 2019, doi: 10.1109/TNNLS.2018.2869375.

[55] T. Yang, N. Sun, and Y. Fang, "Adaptive fuzzy control for a class of MIMO underactuated systems with plant uncertainties and actuator deadzones: Design and experiments," *IEEE Trans. Cybern.*, early access, Feb. 2, 2021, doi: 10.1109/TCYB.2021.3050475.

[56] S. Li, L. Ding, H. Gao, Y.-J. Liu, N. Li, and Z. Deng, "Reinforcement learning neural network-based adaptive control for state and input time-delayed wheeled mobile robots," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 11, pp. 4171–4182, Nov. 2020, doi: 10.1109/TSMC.2018.2870724.

[57] T. Yang, N. Sun, Y. Fang, X. Xin, and H. Chen, "New adaptive control methods for n-link robot manipulators with online gravity compensation: Design and experiments," *IEEE Trans. Ind. Electron.*, vol. 68, no. 1, pp. 539–548, Jan. 2022, doi: 10.1109/TIE.2021.3050371.

[58] A. H. Tahoun, "Time-varying multiplicative/additive faults compensation in both actuators and sensors simultaneously for nonlinear systems via robust sliding mode control scheme," *J. Franklin Inst.*, vol. 356, no. 1, pp. 103–128, Jan. 2019, doi: 10.1016/j.jfranklin.2018.08.027.

[59] Y. Zhao, Z. Duan, and G. Wen, "Finite-time consensus for second-order multi-agent systems with saturated control protocols," *IET Control Theory Appl.*, vol. 9, no. 3, pp. 312–319, Feb. 2015, doi: 10.1049/iet-cta.2014.0061.

[60] H. Su, M. Z. Q. Chen, X. Wang, and J. Lam, "Semiglobal observer-based leader-following consensus with input saturation," *IEEE Trans. Ind. Electron.*, vol. 61, no. 6, pp. 2842–2850, Jun. 2014, doi: 10.1109/TIE.2013.2275976.

[61] Q. Zhou, H. Li, C. Wu, L. Wang, and C. K. Ahn, "Adaptive fuzzy control of nonlinear systems with unmodeled dynamics and input saturation using small-gain approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 8, pp. 1979–1989, Aug. 2017, doi: 10.1109/TSMC.2016.2586108.

[62] *2-DOF Helicopter*, Quanser, Markham, ON, Canada, 2011.

[63] L.-X. Wang, "Stable adaptive fuzzy control of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 146–155, May 1993, doi: 10.1109/91.227383.

[64] M. A. Rahimian and M. S. Tavazoei, "Improving integral square error performance with implementable fractional-order PI controllers," *Optim. Control Appl. Methods*, vol. 35, no. 3, pp. 303–323, May 2014, doi: 10.1002/oca.2069.

[65] *2 DOF Ball Balancer Student Workbook*, Quanser, Markham, ON, Canada, 2013, p. 26.

[66] *2D Ball Balancer User Manual*, Quanser Educ. Solutions, Markham, ON, Canada, 2013.

[67] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Trans. Autom. Control*, vol. 42, no. 5, pp. 674–690, May 1997, doi: 10.1109/9.580874.

[68] S. Pitis, "Source traces for temporal difference learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, Apr. 2018, pp. 3952–3959.

[69] Y. Tian, H. Yan, H. Zhang, X. Zhan, and Y. Peng, "Dynamic output-feedback control of linear semi-Markov jump systems with incomplete semi-Markov kernel," *Automatica*, vol. 117, Jul. 2020, Art. no. 108997, doi: 10.1016/j.automatica.2020.108997.

[70] H. Yan, J. Sun, H. Zhang, X. Zhan, and F. Yang, "Event-triggered H_∞ state estimation of 2-DOF quarter-car suspension systems with nonhomogeneous Markov switching," *IEEE Trans. Syst. Man, Cybern., Syst.*, vol. 50, no. 9, pp. 3320–3329, Sep. 2020, doi: 10.1109/TSMC.2018.2852688.

Rupam Singh (Member, IEEE) received the B.Tech. degree in electrical and electronics engineering from Hindustan College of Science and Technology, Mathura, India, in 2013, the M.Tech. degree in control system from Amity University, Noida, India, in 2016, and the Ph.D. degree in intelligent control and robotics from the Department of Electrical Engineering, Delhi Technological University, New Delhi, India, in 2021.

She is currently working as a Post-Doctoral Fellow with the Institute for Intelligent System Technologies, Alpen-Adria-Universität, Klagenfurt, Austria. She has many publications in peer-reviewed journals and presented her research articles in several international conferences. Her area of research is artificial intelligence, machine learning, control systems, condition monitoring, and their application in robotics and unmanned vehicles.

Bharat Bhushan (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees from Delhi College of Engineering, University of Delhi, New Delhi, India, in 1992, 1996, and 2012, respectively.

He is currently a Professor with the Department of Electrical Engineering, Delhi Technological University, New Delhi. His area of research is control systems, intelligent control, soft computing techniques, and bioinspired algorithms.