2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Off-policy Reinforcement Learning for a Robust Optimal Control Problem with Real Parametric Uncertainty

Athira Mullachery and Shaikshavali Chitraganti

Abstract— **This paper addresses an off-policy Reinforcement learning algorithm for robust linear quadratic regulator (R-LQR) problem of continuous-time linear dynamical systems with parametric uncertainties based on policy iteration framework. A modified algebraic Riccati equation is presented for the R-LQR problem and is further transformed into standard linear quadratic regulator problem. The proposed model-free off-policy R-LQR algorithm learns the control policy using generated data samples that obviate the requirement of system dynamics. Numerical simulation examples of spring-mass system with uncertain stiffness are provided to illustrate effectiveness of the approach.**

I. INTRODUCTION

The aim of optimal control problem is to minimize a objective functional subject to system dynamics under consideration [1], [2] and it finds many important applications such as power systems, missile control, underwater vehicle control etc. Solution of optimal control problem is usually obtained via variational approach or dynamic programming, where it is assumed that the system model is available. For complex systems such as underwater robotics systems and quadrotors in uncertain environment, it is extremely difficult to model precisely due to complex hydrodynamic and aerodynamic forces acting on them during the passage.

Systems to be controlled often contain inherent uncertainties arising from modeling errors or imprecise parameter quantification, categorized into unstructured and structured uncertainty [3]. A new class of controller called as robust linear quadratic regulator (R-LQR) was proposed for systems with parametric uncertainty, which generally falls under structured uncertainty [4]. The approach assumes that the system dynamics is completely known to solve the R-LQR problem, which may not be possible in every scenario.

Reinforcement learning (RL), which is an key branch of machine learning, became prominent in addressing optimal control problems, where the environment is uncertain and cannot be fully represented as a mathematical model [5]. This approach was termed as adaptive dynamic programming (ADP), which offers algorithms that generally fall under the categories of value iteration (VI) and policy iteration (PI) [6], [7]. ADP finds considerable applications in handling optimal state regulation problem [8], [9], tracking problem [10]–[12], output regulation problem [13], game theory [14]–[16] and was also extended to decentralized control of large scale systems [17], [18]. Depending upon the strategy used for data generation and policy update, ADP is mainly subdivided into on-policy and off-policy approaches. In on-policy algorithms, control policy under evaluation is used for data generation, however this can potentially introduce bias in the cost function due to added excitation noise for online implementation, which is not desirable. Also on-policy algorithms can lead to undesirable system responses during the learning phase, hence it is not widely recommended for many engineering applications. To address this issue, off-policy RL is employed to optimal control problems, where the behavior policies are utilized to collect the data and the target policy is learned online [19]. Off-policy algorithms can be designed to ensure stable and reliable system behavior throughout the learning process and it also potentially alleviate the exploration problem as data is collected using the behavior policies. There has been extensive use of off-policy algorithm in addressing optimal control problems with unknown dynamics: optimal tracking problems, zero-sum game problems and multi-agent problems in continuous-time (CT) and discrete-time (DT) setting [20]–[23].

There has been efforts to address the optimal control problems with uncertainty for CT systems in a model-free way using on-policy approach, however partial knowledge of the system dynamics were still utilized to find the optimal policy [24], [25]. In line with that, model-free CT linear quadratic regulator (LQR) for systems subjected to additive disturbances based on Kleinman's policy iteration was addressed in [26]. The approach was further extended to robust stabilization problem of DT systems with bounded mismatched uncertainties using on-policy and off-policy variants [27]. Surpassing these studies, robust control of nonlinear systems with matched and unmatched uncertainties in system and input dynamics were investigated in [28] by combining off-policy RL and neural network approximation.

To the best of our knowledge, off-policy PI algorithm has not been employed to address R-LQR problem for CT systems with parametric uncertainties. This paper explores model-free off-policy PI algorithm for R-LQR control of systems with parametric uncertainties. The standard optimal control problem and robust LQR problem is presented in section II. Model-free off-policy R-LQR PI algorithm and its implementation are provided in Section III. Numerical simulations are presented in section IV to validate the proposed algorithm and Section V provides the conclusion.

Notations: Let $\mathbb{R}^n$ denote the standard Euclidean space of dimension n. The set of non-negative integers are represented

This work was supported in part by IIT Palakkad Technology IHub Foundation Technology Development Grant IPTIF/TD/IP/001

The authors are with the Department of Electrical Engineering, Indian Institute of Technology Palakkad, 678623, India 122204001@smail.iitpkd.ac.in, shaik@iitpkd.ac.in

as $\mathbb{Z}_+$. For matrix M, $M^\top$ and M^{-1} denotes the transpose and inverse respectively. diag(M) represents the diagonal matrix with values only along the main diagonal. In the case of two real symmetric matrices X and Y, $X \succ Y$ $(X \succeq Y)$ represents that $X-Y$ is positive definite (semi-definite) matrix. The symbol $\otimes$ denotes the Kronecker product and I_n represents the identity matrix of dimension n.

II. BACKGROUND

A. Optimal control problem

Consider a continuous-time linear dynamics

$$\dot{x} = Ax + Bu, \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{n_u}$ is the control input, and matrices A and B are of suitable dimension. The objective of standard LQR problem is to design a control policy

$$u = -K_u x \tag{2}$$

that minimizes the cost functional

$$J = \int_0^\infty \left[x^\top Q x + u^\top \Re u\right] dt, \tag{3}$$

where $Q \succeq 0$ and $\Re \succ 0$. The optimal solution of this problem relies on the algebraic Riccati equation (ARE)

$$PA + A^\top P + Q - PB\Re^{-1}B^\top P = 0, \tag{4}$$

where $P = P^\top \succ 0$. The above ARE is solved to obtain P, and the optimal feedback gain is obtained as follows

$$K_u = \Re^{-1}B^\top P. \tag{5}$$

B. Robust Linear Quadratic Regulator

Assumption 1: All the uncertainties are present within matrix A and are deemed to be within a bounded interval.

Under Assumption 1, the uncertainties in A can be modeled as:

$$\tilde{A} = A + \sum_{i=1}^{q} \eta_i W_i,\ \eta_i \in \mathbb{R}, \quad |\eta_i| \le 1, \tag{6a}$$

$$W_i = \mathfrak{l}_i \mathfrak{n}_i^\top, \tag{6b}$$

$$\mathscr{L} = [\mathfrak{l}_1\ \mathfrak{l}_2\ \mathfrak{l}_3\ \cdots], \quad \mathscr{N} = [\mathfrak{n}_1\ \mathfrak{n}_2\ \mathfrak{n}_3\ \cdots] \tag{6c}$$

with A representing the nominal system. Here W_i represent the structure of the uncertainty, which is scaled by η_i. It is assumed that there are q uncertain parameters and each of them is considered to be within a bounded interval.

Remark 1. Uncertainties of the type mentioned in Assumption 1 generally occurs in large space structures [29], where the parameters such as stiffness (k_S) and damping coefficient appear in matrix A are quite uncertain, but the values of all masses, which influence B is considered to be known with higher accuracy.

Now, by applying $\tilde{A}$ in ARE (4) gives the following

$$P\tilde{A} + \tilde{A}^\top P - PB\Re^{-1}B^\top P + Q = 0 \tag{7}$$

$$PA + A^\top P + \sum_{i=1}^{q} \eta_i P W_i + \sum_{i=1}^{q} \eta_i W_i^\top P - PB\Re^{-1}B^\top P + Q = 0. \tag{8}$$

Now, using (8), by further simplification and by applying the inequality $2|ab| \le \gamma a^2 + \frac{1}{\gamma} b^2$ for scalars $a \in \mathbb{R}$ and $b \in \mathbb{R}$ with arbitrary $\gamma > 0$, robust algebraic Riccati equation (R-ARE) can be obtained [4] as

$$PA + A^\top P + \left(Q + \gamma \mathscr{N}\mathscr{N}^\top\right) - P\left(-\frac{1}{\gamma}\mathscr{L}\mathscr{L}^T + B\Re^{-1}B^\top\right)P = 0. \tag{9}$$

To achieve stability and robustness, the R-ARE (9) must be solved to find P, and then we can determine the optimal policy using (5), but solution of (9) requires the complete knowledge of system matrices A, B that may be difficult to obtain in real-time scenarios. The following remark shows the translation of R-LQR to standard LQR problem.

Remark 2. The R-ARE (9) can be reformulated as follows

$$PA + A^\top P + \tilde{Q} - P\bar{B}\bar{\Re}^{-1}\bar{B}^\top P = 0, \tag{10}$$

where $\bar{B} = [\mathscr{L} \quad B]$, $\bar{\Re} = \begin{bmatrix} -\gamma I_{q\times q} & 0 \\ 0 & \Re \end{bmatrix}$ and $\tilde{Q} = \left(Q + \gamma \mathscr{N}\mathscr{N}^\top\right)$. Consider a cost function

$$\tilde{J} = \int_0^\infty x^\top \left[\tilde{Q} + K^\top \bar{\Re} K\right] x\, dt \tag{11}$$

with dynamics

$$\dot{x} = Ax + \bar{B}\bar{u}, \tag{12}$$

where

$$\bar{u} = \underbrace{\begin{bmatrix} K_w & K_u \end{bmatrix}^\top}_{\triangleq K} x. \tag{13}$$

Here $\bar{u}$ consists of two gains, K_w is the gain contributed by the uncertain part and K_u is the control input gain. The R-LQR problem is now transformed into the standard LQR problem with cost function of the form (11), whose solution can be obtained by solving R-ARE (10).

Iterative Lyapunov equation corresponds to (10) can be obtained as follows

$$A_k^{i\top} P^i + P^i A_k^i + \tilde{Q} + K^{i\top} \bar{\Re} K^i = 0 \tag{14}$$

with the feedback gain update

$$K^{i+1} = \bar{\Re}^{-1}\bar{B}^\top P^i. \tag{15}$$

where $A_k^i = A - \bar{B}K^i$. This is a model-based policy iteration method for CT linear system [30], where (14) is used for policy evaluation and (15) for policy improvement.

III. Off-policy Reinforcement learning for R-LQR

This section presents an off-policy RL algorithm to solve the R-ARE (9) in a model-free framework. Off-policy RL helps to learn policies using data generated by policies other than the optimal ones. This is achieved with the help of behavior policy ($\bar{u}$) for data collection and target policy (K^i) for improvement. Rewrite the system (12) as follows

$$\dot{x} = A_k^i x + \bar{B}(K^i x + \bar{u}), \tag{16}$$

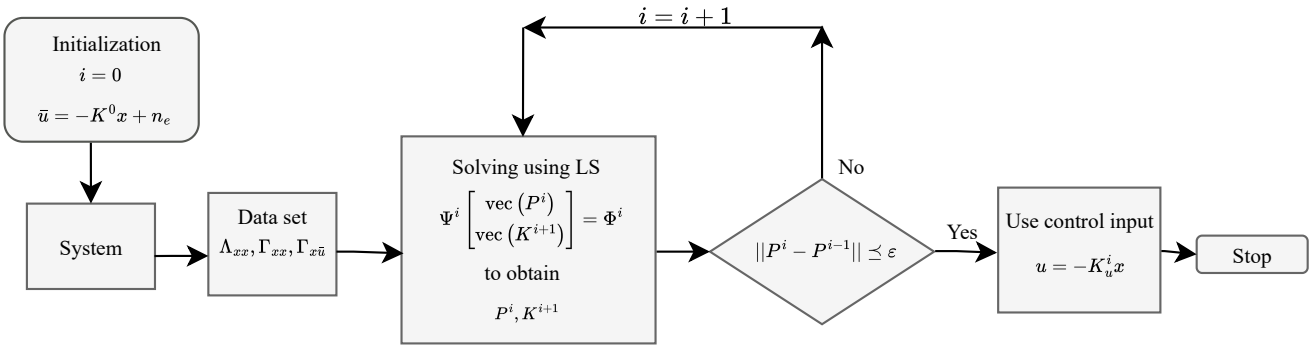


Fig. 1: Schematic of off-policy R-LQR PI algorithm

where $A_k^i = A - \bar{B}K^i$. Consider the derivative of a Lyapunov function $x^\top P^i x$ with respect to time and substituting (14), (16) and $\bar{B}^\top P^i = \bar{\Re} K^{i+1}$ (15) gives

$$\begin{aligned}\frac{d}{dt}(x^\top P^i x) &= x^\top \left(A_k^{i\top} P^i + P^i A_k^i\right) + 2(K^i x + \bar{u})^\top \bar{B}^\top P^i x \\ &= -x^\top \left(\tilde{Q} + K^{i\top} \bar{\Re} K^i\right) + 2(K^i x + \bar{u}) \bar{\Re} K^{i+1} x.\end{aligned} \tag{17}$$

Rearranging (17) and applying integration over $[t, t+\delta t]$ gives

$$\begin{aligned}&x(t+\delta t)^\top P^i x(t+\delta t) - x(t)^T P^i x(t) \\ &\quad = -\int_t^{t+\delta t} x^T \left(\tilde{Q} + K^{i\top} \bar{\Re} K^i\right) x \, d\tau \\ &\quad\quad + 2\int_t^{t+\delta t} \left(\bar{u} + K^i x\right)^T \bar{\Re} K^{i+1} x \, d\tau.\end{aligned} \tag{18}$$

Let

$$\Lambda_{xx} = \left[\xi_1, \xi_2, \cdots, \xi_{l-1}\right]^\top, \text{ where } \xi_i = (x \otimes x)|_{t_i}^{t_{i+1}} \tag{19a}$$

$$\Gamma_{xx} = \left[\chi_1, \chi_2, \cdots, \chi_{l-1}\right]^\top, \text{ where } \chi_i = \int_{t_i}^{t_{i+1}} (x \otimes x) \, d\tau \tag{19b}$$

$$\Gamma_{x\bar{u}} = \left[\mu_1, \mu_2, \cdots, \mu_{l-1}\right]^\top, \text{ where } \mu_i = \int_{t_i}^{t_{i+1}} (x \otimes \bar{u}) \, d\tau, \tag{19c}$$

where $t_{i+1} = t_i + \delta t$, $\delta t > 0$, and $l \in \mathbb{Z}_+$. Now using (19) and vec operator, (18) is modified as follows

$$\Psi^i \begin{bmatrix} \text{vec}(P^i) \\ \text{vec}(K^{i+1}) \end{bmatrix} = \Phi^i, \tag{20}$$

where

$$\begin{aligned}\Psi^i &= \left[\Lambda_{xx}, -2\Gamma_{xx}(I_{n_x} \otimes K^{i\top} \bar{\Re}) - 2\Gamma_{x\bar{u}}(I_{n_x} \otimes \bar{\Re})\right], \\ \Phi^i &= -\Gamma_{xx} \, \text{vec}(\tilde{Q} + K^{i\top} \bar{\Re} K^i).\end{aligned}$$

Then, (20) can be solved as follows

$$\begin{bmatrix} \text{vec}(P^i) \\ \text{vec}(K^{i+1}) \end{bmatrix} = \left(\Psi^{i\top} \Psi^i\right)^{-1} \Psi^{i\top} \Phi^i. \tag{21}$$

From the solution of (21), the policy K_u^i can be extracted and then the resulting control input will be used.

Remark 3. For the existence of unique solution pair (P^i, K^{i+1}), the following rank condition [21] must be satisfied.

$$rank\left(\begin{bmatrix}\Gamma_{xx} & \Gamma_{x\bar{u}}\end{bmatrix}\right) = \frac{n_x(n_x+1)}{2} + (n_u + n_w) n_x, \tag{22}$$

where n_x, n_u and n_w are number of states, input and uncertain parameters. If condition (22) is satisfied, then Ψ^i has full column rank $\forall i \in \mathbb{Z}_+$. The above requirement is termed as persistence of excitation (PE) condition in ADP, which is necessary to solve (20). There must be an exploration noise n_e injected to the control input to satisfy the PE condition and online implementation. Exploration signals that are generally used for different practical applications include random noise, exponentially decaying signals and sum of sinusoids [14], [31] etc.

The procedure for off-policy R-LQR PI is provided in Algorithm 1 and schematic is shown in Fig. 1.

Algorithm 1 Model-free off-policy R-LQR PI algorithm

1: Initialization: Set $i = 0$ and start using a stabilizing behavior policy $\bar{u} = -K^0 x + n_e$ over the interval $[t_0 \; t_l]$.
2: Collect online data to compute Ψ^i and Φ^i.
3: Solve (21) using least squares to obtain P^i and K^{i+1}.
4: Check for the convergence $\left\|P^i - P^{i-1}\right\| \leq \varepsilon$, if not, increment $i = i+1$ and go to 3.
5: Extract and update the control policy $u = -K_u^i x$.

Online information (19) can be collected by applying the initial stabilizing policy K^0 and then data can be recorded in Λ_{xx}, Γ_{xx} and $\Gamma_{x\bar{u}}$ matrices. This information can be used to build Ψ^i and Φ^i, and least square equation (21) can be solved iteratively till the convergence criterion is satisfied. In the last step, from the resulting K^i, the control gain K_u^i is extracted and is applied to the system.

Theorem 1. *Convergence of off-policy R-LQR PI algorithm: Starting from a initial stabilizing policy $\bar{u} = -K^0 x$, if condition (22) is satisfied, then sequences $\left\{P^i\right\}_{i=0}^{\infty}$ and $\left\{K^i\right\}_{i=1}^{\infty}$ obtained by solving (21) converge towards the optimal solution P^* of R-ARE (9) and K^* respectively.*

Proof. With the given stabilizing gain K^0, let P^i, $i=0,1,2...$ be the solution of the iterative Lyapunov equation

$$A_k^{i\top}P^i + P^iA_k^i + \tilde{Q} + K^{i\top}\bar{\Re}K^i = 0,$$

and the optimal gain is recursively determined by

$$K^{i+1} = \bar{\Re}^{-1}\bar{B}^\top P^i.$$

Then, the conditions defined below hold [30]:

1) $A-\bar{B}K^i$ is Hurwitz.
2) $P^* \preceq P^{i+1} \preceq P^i$.
3) $\lim_{i\to\infty} P^i = P^*$ and $\lim_{i\to\infty} K^i = K^*$.

From the above properties, (14) converges to optimal P^* and K^*. The solution pair $(\hat{P}^i, \hat{K}^{i+1})$ obtained from

$$\Psi^i \begin{bmatrix} \text{vec}\left(\hat{P}^i\right) \\ \text{vec}\left(\hat{K}^{i+1}\right) \end{bmatrix} = \Phi^i$$

must satisfy (14) according to (18). From remark 3, such a solution pair is always unique. Therefore, given the uniqueness of the solution, it can be shown that $\hat{P}^i = P^i$ and $\hat{K}^{i+1} = K^{i+1}$. Hence the solution of P^i to Lyapunov recursion (14) and the corresponding $K^{i+1} = \bar{\Re}^{-1}\bar{B}^\top P^i$ are identical to the solution of (20) $\forall i$. Therefore, the off-policy R-LQR (20) is equivalent to (14), which converges to the optimal solution as mentioned by the properties of Lyapunov recursion. Thus, convergence of the off-policy R-LQR PI is proved. □

IV. SIMULATIONS AND RESULTS

In this section, off-policy R-LQR PI controller is compared with the nominal-LQR (N-LQR) (4) design, which is without uncertainties through simulation examples. Precise knowledge of the system matrices are not known apriori for learning the optimal solutions. For more clarity to the readers, to conduct simulations we utilize the system matrices for data generation only, while our algorithm just depends on state and input data. We present two examples, one involving single parametric uncertainty and the other involving double parametric uncertainty, given as example 1 and example 2 respectively.

A. Example 1: Single spring-mass system

Spring-mass system with an uncertain stiffness $k_S \in [0.5, 2]$ that connects two masses $m_1 = m_2 = 1$kg is considered as shown in Fig. 2. The objective is to control position $y = x_2$ of mass m_2 by applying control u on mass m_1. The system

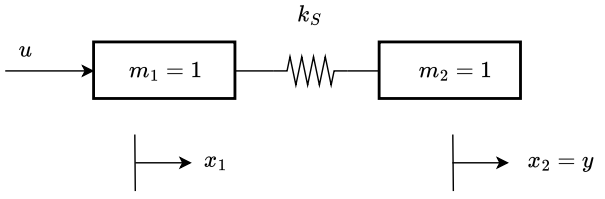


Fig. 2: Single spring-mass system

matrices are considered as:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_S & k_S & 0 & 0 \\ k_S & -k_S & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

The design parameters are chosen as, $\Re = 0.01$, $\gamma = 1$, $Q = \text{diag}\begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$, and $k_S = 1.25$, which is the midpoint value of uncertain stiffness range. The uncertain part of $\tilde{A}$ is chosen as

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -w & w & 0 & 0 \\ w & -w & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ l \\ -l \end{bmatrix} \begin{bmatrix} -n \\ n \\ 0 \\ 0 \end{bmatrix}^\top$$

with $\mathscr{L} = [0\ 0\ l\ -l]^\top$, $\mathscr{N}^\top = [-n\ n\ 0\ 0]$, $w = 0.75$ and $l = n = 0.866$. Depending on the value of η =-1 or 1, k_S is considered to have its minimum and maximum value of 0.5 and 2 according to (6a).

B. Example 2: Two spring-mass system

Consider the example of three masses $m_1 = m_2 = m_3 = 1$kg coupled by means of two uncertain springs k_{S1}, $k_{S2} \in [0.5, 2]$ as shown in Fig. 3. The position $y = x_3$ must be regulated by control signals u_1 and u_2 acting on mass m_1 and m_2. The system dynamics can be represented as follows

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -k_{S1} & k_{S1} & 0 & 0 & 0 & 0 \\ k_{S1} & -k_{S1}-k_{S2} & k_{S2} & 0 & 0 & 0 \\ 0 & k_{S2} & -k_{S2} & 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix},$$

where $k_{S1} = k_{S2} = 1.25$. The uncertain part is given by,

$$\mathscr{L} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ l & 0 \\ -l & l \\ 0 & -l \end{bmatrix} \quad \mathscr{N} = \begin{bmatrix} -n & 0 \\ n & -n \\ 0 & n \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},$$

where $l = n = 0.866$. Consider $\Re = 0.01I_2$, $\gamma = 1$, $Q = \text{diag}[0\ 0\ 1\ 0\ 0\ 0]$. The direct solution using N-ARE (4) and

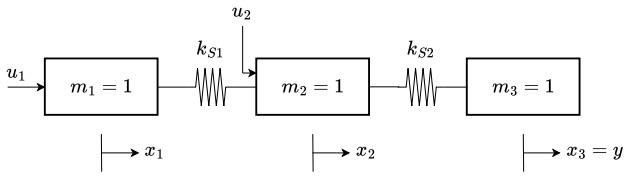
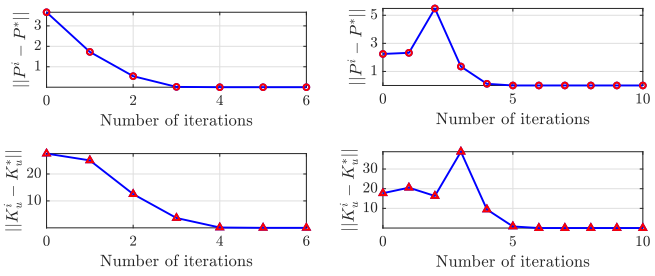


Fig. 3: Two spring-mass system

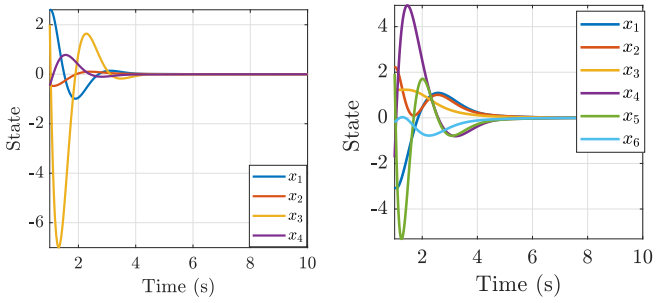
TABLE I: Comparison of matrix P and feedback gain K_u for N-LQR, R-ARE and Off-policy R-LQR PI

System	Method	P	K_u
Single spring	N-LQR (4)	$\begin{bmatrix} 0.2633 & 0.1390 & 0.0809 & 0.3427 \\ 0.1390 & 0.7605 & 0.0191 & 0.4045 \\ 0.0809 & 0.0191 & 0.0402 & 0.0899 \\ 0.3427 & 0.4045 & 0.0899 & 0.5609 \end{bmatrix}$	$\begin{bmatrix} 8.0902 & 1.9098 & 4.0225 & 8.9945 \end{bmatrix}$
	R-ARE (9)	$\begin{bmatrix} 1.2882 & 0.0366 & 0.2553 & 1.8243 \\ 0.0366 & 1.3576 & -0.0127 & 0.9714 \\ 0.2553 & -0.0127 & 0.0748 & 0.3313 \\ 1.8243 & 0.9714 & 0.3313 & 3.8020 \end{bmatrix}$	$\begin{bmatrix} 25.5316 & -1.2656 & 7.4831 & 33.1313 \end{bmatrix}$
	Off-policy R-LQR PI (ε=0.0001)	$\begin{bmatrix} 1.2882 & 0.0366 & 0.2553 & 1.8243 \\ 0.0366 & 1.3576 & -0.0127 & 0.9714 \\ 0.2553 & -0.0127 & 0.0748 & 0.3313 \\ 1.8243 & 0.9714 & 0.3313 & 3.8020 \end{bmatrix}$	$\begin{bmatrix} 25.5316 & -1.2656 & 7.4831 & 33.1313 \end{bmatrix}$
Two spring	N-LQR (4)	$\begin{bmatrix} 0.018 & 0.004 & -0.004 & 0.005 & 0.013 & 0.015 \\ 0.004 & 0.201 & 0.151 & -0.004 & 0.064 & 0.295 \\ -0.004 & 0.151 & 0.749 & -0.002 & 0.023 & 0.401 \\ 0.005 & -0.004 & -0.002 & 0.010 & 0.002 & -0.003 \\ 0.013 & 0.064 & 0.023 & 0.002 & 0.036 & 0.089 \\ 0.015 & 0.295 & 0.401 & -0.003 & 0.089 & 0.519 \end{bmatrix}$	$\begin{bmatrix} 0.518 & -0.423 & -0.206 & 1.0 & 0.185 & -0.341 \\ 1.27 & 6.375 & 2.35 & 0.185 & 3.567 & 8.95 \end{bmatrix}$
	R-ARE (9)	$\begin{bmatrix} 0.356 & -0.029 & -0.040 & 0.082 & 0.012 & 0.399 \\ -0.029 & 1.522 & -0.002 & 0.037 & 0.277 & 2.062 \\ -0.040 & -0.002 & 1.379 & -0.023 & -0.019 & 0.895 \\ 0.082 & 0.037 & -0.023 & 0.042 & 0.006 & 0.127 \\ 0.012 & 0.277 & -0.019 & 0.006 & 0.079 & 0.369 \\ 0.399 & 2.062 & 0.895 & 0.127 & 0.369 & 4.598 \end{bmatrix}$	$\begin{bmatrix} 8.224 & 3.701 & -2.327 & 4.156 & 0.609 & 12.724 \\ 1.171 & 27.672 & -1.924 & 0.609 & 7.855 & 36.911 \end{bmatrix}$
	Off-policy R-LQR PI (ε=0.0001)	$\begin{bmatrix} 0.356 & -0.029 & -0.040 & 0.082 & 0.012 & 0.399 \\ -0.029 & 1.522 & -0.002 & 0.037 & 0.277 & 2.062 \\ -0.040 & -0.002 & 1.379 & -0.023 & -0.019 & 0.895 \\ 0.082 & 0.037 & -0.023 & 0.042 & 0.006 & 0.127 \\ 0.012 & 0.277 & -0.019 & 0.006 & 0.079 & 0.369 \\ 0.399 & 2.062 & 0.895 & 0.127 & 0.369 & 4.599 \end{bmatrix}$	$\begin{bmatrix} 8.224 & 3.701 & -2.327 & 4.156 & 0.609 & 12.725 \\ 1.171 & 27.673 & -1.925 & 0.609 & 7.855 & 36.912 \end{bmatrix}$



(a) Single spring-mass system (b) Two spring-mass system

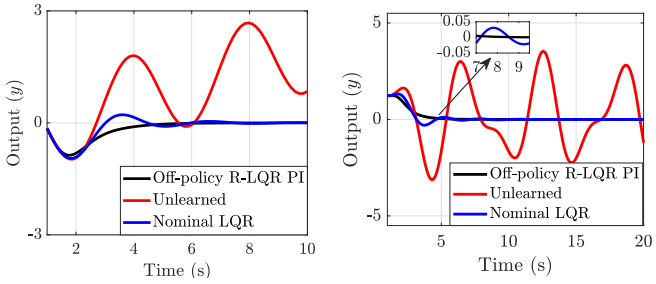
Fig. 4: Convergence of P and K_u for off-policy R-LQR PI



(a) Single spring-mass system (b) Two spring-mass system

Fig. 5: State trajectories with off-policy R-LQR PI

R-ARE (9) for both the examples are given in Table I. For simulating off-policy R-LQR PI algorithm, initial states $x_0 = [0\ 1\ 0\ 0]^\top$ and $x_0 = [0\ 0\ 1\ 0\ 0\ 0]^\top$ are chosen for example 1 and 2 respectively. Stabilizing behavior policy is applied till the end of data collection from $t = 0$ to 1s with injected excitation noise and the data is collected at intervals of 0.01s. The learned P and K_u matrices with off-policy R-LQR PI is given in Table I, which exactly converges to the optimal solution of R-ARE (9) and the convergence is shown in Fig. 4. The learned controller is applied to the system after 1s and the state trajectories are plotted in Fig. 5, that exactly converges to zero. The output response of unlearned system, off-policy R-LQR and N-LQR are shown in Fig. 6, which clearly shows that the off-policy R-LQR converges quickly and the transients are also minimized, which indicates the robustness against the uncertainties.



(a) Single spring-mass system (b) Two spring-mass system

Fig. 6: Comparison of output profile of N-LQR and off-policy R-LQR PI

V. CONCLUSIONS

Off-policy reinforcement learning algorithm is addressed for robust linear quadratic regulator (R-LQR) problem of continuous-time linear systems with parametric uncertainties using policy iteration (PI) framework. The proposed model-free off-policy R-LQR PI algorithm makes use of the data samples rather the system dynamics to solve the modified Riccati equation corresponding to R-LQR problem which is beneficial in situations where there are uncertainties in the plant, as the proposed method more effectively mitigates the impact of the system's uncertain dynamics. Numerical simulations are carried out to validate our algorithm for systems subjected to parametric uncertainties. Future extension of this work could involve applying the off-policy R-LQR PI algorithm to nonlinear systems with a more general type of uncertainty.

REFERENCES

[1] D. E. Kirk, *Optimal control theory: an introduction*. Courier Corporation, 2004.

[2] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.

[3] J. Douglas and M. Athans, "Robust linear quadratic designs with real parameter uncertainty," *IEEE Transactions on automatic control*, vol. 39, no. 1, pp. 107–111, 1994.

[4] J. S. Douglas, "Linear quadratic control for systems with structured uncertainty," Ph.D. dissertation, Massachusetts Institute of Technology, 1991.

[5] D. Bertsekas, *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[6] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[7] D. Liu, S. Xue, B. Zhao, B. Luo, and Q. Wei, "Adaptive dynamic programming for control: A survey and recent advances," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 142–160, 2020.

[8] D. Liu, Q. Wei, D. Wang, X. Yang, and H. Li, *Adaptive dynamic programming with applications in optimal control*. Springer, 2017.

[9] K. G. Vamvoudakis, D. Vrabie, and F. L. Lewis, "Online adaptive algorithm for optimal control with integral reinforcement learning," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 17, pp. 2686–2710, 2014.

[10] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2226–2236, 2011.

[11] C. Mu, Z. Ni, C. Sun, and H. He, "Data-driven tracking control with adaptive dynamic programming for a class of continuous-time nonlinear systems," *IEEE transactions on cybernetics*, vol. 47, no. 6, pp. 1460–1470, 2016.

[12] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Transactions on Automatic control*, vol. 59, no. 11, pp. 3051–3056, 2014.

[13] W. Gao and Z.-P. Jiang, "Adaptive dynamic programming and adaptive optimal output regulation of linear systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4164–4169, 2016.

[14] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica*, vol. 43, no. 3, pp. 473–481, 2007.

[15] B. Luo, Y. Yang, and D. Liu, "Policy iteration Q-learning for data-based two-player zero-sum game of linear discrete-time systems," *IEEE Transactions on Cybernetics*, vol. 51, no. 7, pp. 3630–3640, 2020.

[16] W. Wang, X. Chen, H. Fu, and M. Wu, "Data-driven adaptive dynamic programming for partially observable nonzero-sum games via Q-learning method," *International Journal of Systems Science*, vol. 50, no. 7, pp. 1338–1352, 2019.

[17] B. Zhao and Y. Li, "Model-free adaptive dynamic programming based near-optimal decentralized tracking control of reconfigurable manipulators," *International Journal of Control, Automation and Systems*, vol. 16, no. 2, pp. 478–490, 2018.

[18] X. Yang and H. He, "Adaptive dynamic programming for decentralized stabilization of uncertain nonlinear large-scale systems with mismatched interconnections," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 8, pp. 2870–2882, 2018.

[19] B. Luo, H.-N. Wu, and T. Huang, "Off-policy reinforcement learning for $\mathscr{H}_\infty$ control design," *IEEE transactions on cybernetics*, vol. 45, no. 1, pp. 65–76, 2014.

[20] Y. Wen, H. Zhang, H. Su, and H. Ren, "Optimal tracking control for non-zero-sum games of linear discrete-time systems via off-policy reinforcement learning," *Optimal Control Applications and Methods*, vol. 41, no. 4, pp. 1233–1250, 2020.

[21] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, 2012.

[22] B. Kiumarsi, F. L. Lewis, and Z.-P. Jiang, "$\mathscr{H}_\infty$ control of linear discrete-time systems: Off-policy reinforcement learning," *Automatica*, vol. 78, pp. 144–152, 2017.

[23] A. Mullachery and S. Chitraganti, "Off-policy reinforcement learning for optimal control of a two wheeled self balancing robot," in *Ninth Indian Control Conference*, 2023, pp. 383–388.

[24] J. Na, J. Zhao, G. Gao, and Z. Li, "Output-feedback robust control of uncertain systems via online data-driven learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2650–2662, 2020.

[25] J. Zhao and Q. Zeng, "Adaptive robust control for uncertain systems via data-driven learning," *Journal of Sensors*, vol. 2022, pp. 1–9, 2022.

[26] B. Pang, T. Bian, and Z.-P. Jiang, "Robust policy iteration for continuous-time linear quadratic regulation," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 504–511, 2021.

[27] Y. Yang, Z. Guo, H. Xiong, D.-W. Ding, Y. Yin, and D. C. Wunsch, "Data-driven robust control of discrete-time uncertain linear systems via off-policy reinforcement learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 12, pp. 3735–3747, 2019.

[28] A. Amirparast and S. Kamal Hosseini Sani, "Off-policy reinforcement learning algorithm for robust optimal control of uncertain nonlinear systems," *International Journal of Robust and Nonlinear Control*.

[29] A. Filiatrault, *Elements of earthquake engineering and structural dynamics*. Presses inter Polytechnique, 2013.

[30] D. Kleinman, "On an iterative technique for riccati equation computations," *IEEE Transactions on Automatic Control*, vol. 13, no. 1, pp. 114–115, 1968.

[31] Y. Yang, Y. Wan, J. Zhu, and F. L. Lewis, "$\mathscr{H}_\infty$ tracking control for linear discrete-time systems: model-free Q-learning designs," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 175–180, 2020.