Systems & Control Letters 154 (2021) 104983

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

Model-based and model-free designs for an extended continuous-time LQR with exogenous inputs

Sayak Mukherjee [a,*], He Bai [b], Aranya Chakrabortty [c]

[a] Optimization and Control Group, Pacific Northwest National Laboratory, USA
[b] Mechanical and Aerospace Engineering Department, Oklahoma State University, USA
[c] Department of Electrical and Computer Engineering, North Carolina State University, USA

ARTICLE INFO

Article history:
Received 1 January 2021
Received in revised form 8 April 2021
Accepted 3 June 2021
Available online 26 June 2021

Keywords:
Linear quadratic regulator
Exogenous inputs
Matrix differential equations
Reinforcement learning

ABSTRACT

We present an extended linear quadratic regulator (LQR) design for continuous-time linear time invariant (LTI) systems in the presence of exogenous inputs with a novel feedback control structure. We first propose a model-based solution with cost minimization guarantees for states and inputs using dynamic programming (DP) that out-performs classical LQR with exogenous inputs. The control law consists of a combination of the optimal state feedback and an additional *optimal* term which is dependent on the exogenous inputs. The control gains for the two components are obtained by solving a set of matrix differential equations. We provide these solutions for both finite horizons and steady state cases. In the second part of the paper, we formulate a reinforcement learning (RL) based algorithm which does not need any model information except the input matrix, and can compute approximate steady-state extended LQR gains using measurements of the states, the control inputs, and the exogenous inputs. Both model-based and data-driven optimal control algorithms are tested with a numerical example under different exogenous inputs showcasing the effectiveness of the designs.

Published by Elsevier B.V.

1. Introduction

We consider the classical linear quadratic regulator (LQR) design problem for continuous-time LTI systems in the presence of external input signals. Physical processes in practice are often influenced by extraneous time-varying inputs and cannot be manipulated using a controller, referred to as "exogenous inputs" [1–3]. They can represent coupling variables for one part of a dynamic model with other parts, or can simply be external disturbances. Classically, various disturbance rejection schemes such as active disturbance rejection control [4–6], disturbance observer based control [7,8], or a combination of disturbance estimation and cancellation based control [9] have been developed to handle such scenarios, and guarantee desired control performances despite the disturbance. In [10], a generalized H_2 framework is discussed with full information about the disturbance with the Gaussian noise assumption. Linear quadratic Gaussian (LQG) is also a classical variant. The specific problem of LQR with exogenous inputs has been studied in papers such as [11,12], where the control input is designed with a component that cancels out the effect of the disturbance. However, these methods do not guarantee minimization of the total state and control cost when accommodating the external inputs in the control. We, on the contrary, provided an explicit proof of the optimality when designing an extended LQR control. The designs in [13,14] present exogenous input rejection approaches for output regulation via LQR by imposing a certain structure on the state matrix. Moreover, our control follows an optimal feedback structure with the form of linear feedback of state, exogenous input and its derivative for any generic dynamic systems. A relevant result on this topic is [3], but that design is for a discrete-time system. Its extension to continuous time is quite non-trivial and brings out newer insights and solution approaches, as will be shown in the forthcoming sections.

Thereafter, this article will formulate a model-free variant which will be able to learn an approximate steady-state feedback solutions. Reinforcement learning (RL), originally proposed in [15], is used for this purpose. In recent years, several papers such as [16–21] have used RL for LQR control using a variety of solution techniques such as adaptive dynamic programming (ADP), robust ADP, actor–critic methods, Q-learning, model reduction based RL, zeroth-order optimization based policy gradients etc. However, these RL techniques do not consider the system to be coupled with any exogenous inputs. An overview of recent results can be found in [22]. There are more variants of data-driven control research such as data-dependent linear matrix inequalities in [23], finite horizon LQ and tracking control without models using extremum seeking [24–26], learning safety certificates from

* Corresponding author.
E-mail addresses: sayak.mukherjee@pnnl.gov (S. Mukherjee), he.bai@okstate.edu (H. Bai), achakra2@ncsu.edu (A. Chakrabortty).

https://doi.org/10.1016/j.sysconle.2021.104983
0167-6911/Published by Elsevier B.V.

data [27], various distributed control designs [28–30], stabilizing control designs via policy iterations [31], to name a few. [32] discusses connections between direct and indirect control approaches. As mentioned for ADP/RL [16–18], other research works such as extremum-seeking based learning control [24–26], or other variants focus on computing the linear quadratic policies without exogenous inputs. Relevant results on robust learning control designs such as [33–35], on the other hand, consider systems with disturbances, however, only the standard LQR state feedback gains are learned, and therefore do not guarantee net cost minimization in such settings. We, in this article, alleviate such limitations by using an extended LQR framework in the data-driven setting. We use the results derived in the model-based setting, and then formulate a trajectory relationship consisting of state, controls and exogenous input measurements, thereby aiding in solving the problem in a model-free way.

The first contribution of the paper is to present an extended LQR design where the control input comprises of the optimal state feedback and an additional *optimal* term that minimizes the impact of exogenous inputs for a continuous-time LTI (CTLTI) system with guaranteed quadratic net-cost minimization. The problem is formulated in terms of a set of matrix differential equations for computing the feedback gains using the fundamental results from dynamic programming (DP). Results are tested with different exogenous inputs for a third-order CTLTI system. The second contribution is to extend the design using a model-free and measurement-driven approach. We develop an ADP-based algorithm that can learn steady-state gains for both optimal state feedback and disturbance attenuation components of the control signal, and thereby providing guaranteed net state and control cost minimization. The algorithm is model-free in the sense that the state and disturbance input matrices are not needed to run the algorithm; the control input matrix, however, must be known. Numerical examples show the effectiveness of the RL-based extended LQR design.

The rest of the paper is organized as follows. In Section 2, the model-based LQR design in presence of exogenous input is formulated, followed by matrix differential equations with finite-horizon and steady-state approximated solutions. The RL-based algorithm is developed in Section 3. Numerical simulations validating the two designs are presented in Section 4. Section 5 concludes the paper with future research directions.

Notations: The following notations will be used all throughout the paper. $\mathbb{RH}_\infty$ is the set of all proper, real and rational stable transfer matrices; $\otimes$ denotes Kronecker product; $\mathbf{1_n}$ denotes a column vector of size n with all ones; $\mathcal{N}(A)$ denotes null-space of the matrix A; A_t denotes the time-varying matrix A indexed by the time variable t; $vec(.)$ is the standard vectorization operation.

2. Model-based extended continuous-time LQR

We consider a continuous-time linear time-invariant system as

$$\dot{x} = Ax + Bu + Dw,\ x(0) = x_0, \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, and $w \in \mathbb{R}^p$ is the exogenous input which we assumed to be measurable. Motivated from [3], we would like the optimal feedback control to contain terms involving optimal state feedback, and additional *optimal* feedback depending upon exogenous inputs and their derivative information, and thereby, formulate a linear quadratic objective involving costs associated with states, controls and the exogenous inputs as:

$$J = x^T(T)Q_f x(T) + \int_0^T \left(x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau) + w(\tau)^T S_1 w(\tau)\right) d\tau \tag{2}$$

where $Q \succeq 0$, $R \succ 0$, $S_1 \succeq 0$, $Q_f \succeq 0$ are design matrices, and $T > 0$ is the final time instant until which the control in evaluated. We include $w(t)$ in the cost to keep the formulation much general, i.e., when we consider coupled dynamic systems with multiple components, the exogenous inputs for a certain component can depend on state variables of the same component or other components in a complex manner. For example, in the power grid dynamics, the terminal bus voltages of synchronous generators act as an exogenous input for the corresponding synchronous generator dynamics [36]. However, the bus voltages themselves are dependent on the other grid dynamic states in a complex manner. Please note that minimization of this cost function will produce the optimal control $u(t)$; however, instead of making $u(t)$ only a function of the state $x(t)$, we add an additional optimal feedback term that is governed by $w(t)$ and its derivative for the continuous-time system, which is a distinct contribution of this article, and later on, we will show that this framework will also help us to formulate reinforcement learning based solutions. The control law considers $w(t)$ information in a feed-forward manner, however the gain computation framework with net cost minimization guarantee is a novel solution provided in this paper. We first discuss the finite-horizon solution, and then provide an approximate steady-state formulation which will be used for the reinforcement learning (RL) algorithm presented in the next section. The following theorem presents a set of matrix differential equations (MDEs) that need to be solved to compute the optimal control.

Theorem 1. *The optimal control for the CTLTI system (1) with linear feedback controls depending on $x(t)$ and $w(t)$ and its first order derivatives considering the objective (2) is given by:*

$$u(t) = -K_{1t}x(t) - K_{2t}v(t),\ v(t) = [w(t)^T,\ 1]^T, \tag{3}$$

$$K_{1t} = R^{-1}B^T P_t,\ K_{2t} = \frac{1}{2}R^{-1}B^T P_{2t}, \tag{4}$$

where P_t and P_{2t} are computed as

$$-\dot{P}_t = Q + P_t A + A^T P_t - P_t B R^{-1} B^T P_t, \tag{5}$$

$$-\dot{P}_{2t} = A^T P_{2t} + P_{2t} M_t - P_t B R^{-1} B^T P_{2t} + 2P_t D_1, \tag{6}$$

$$D_1 = [D\ \ 0_{n\times 1}],\ M_t = \begin{bmatrix} 0_{p\times p} & \frac{dw}{dt} \\ 0_{1\times p} & 0 \end{bmatrix}. \tag{7}$$

Proof. We leverage dynamic programming (DP) to prove this theorem. The major motivation to use dynamic programming here is to later extend the methodology for the model independent RL designs where we use *adaptive dynamic programming*, thereby, keeping both the model-based and model-free formulations in a similar underlying framework. Denote $v(t) = [w(t)^T,\ 1]^T$, $D_1 = [D\ \ 0_{n\times 1}]$, and write the plant model as

$$\dot{x} = Ax + Bu + D_1 v. \tag{8}$$

Defining $S = \begin{bmatrix} S_1 & 0 \\ 0 & 0 \end{bmatrix}$ we can write $w^T S_1 w = v^T S v$. Subsequently, for $0 \le t \le T$ we define the value function

$$V_t(x, v) = \min_u \int_t^T (x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau) + v(\tau)^T S v(\tau))\, d\tau + x(T)^T Q_f x(T). \tag{9}$$

Next, consider any very small time interval $[t, t+h]$ where $h > 0$ is a small number. Over this interval, we can write

$$v(t+h) = \begin{bmatrix} w(t) + w(t+h) - w(t) \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} I_p & w(t+h) - w(t) \\ 0_{1\times p} & 1 \end{bmatrix}}_{L_t} \begin{bmatrix} w(t) \\ 1 \end{bmatrix} = L_t v(t), \tag{10}$$

We assume that the initial values during this interval are $x(t) = x_1$, $v(t) = v_1$, and the control input applied during the interval is a constant, i.e., $u = u_1$ during $[t, t+h]$, and we let the states $x(t)$, and exogenous input $w(t)$ evolve over the small time-interval. Cost incurred during $[t, t+h]$ is then given and approximated as

$$\begin{aligned}\mathcal{C}_1 &= \int_t^{t+h} \left(x(\tau)^T Q x(\tau) + u(\tau)^T R u(\tau) + v(\tau)^T S v(\tau)\right)\, d\tau \\ &= h(x_1^T Q x_1 + u_1^T R u_1 + v_1^T S v_1). \end{aligned} \tag{11}$$

The value function V_t is quadratic based on the nature of the cost function, i.e., $V_t(x_1, v_1) = x_1^T P_t x_1 + v_1^T P_{1t} v_1 + x_1^T P_{2t} v_1$. Note that we need to consider the cross-terms between x_1 and v_1 in the value function due to the nature of the cost function. The value of the state at time $t+h$ is given by

$$x_2 = x_1 + h(Ax_1 + Bu_1 + Dv_1). \tag{12}$$

On the other hand, the value of the exogenous input at time instant $t+h$ is given as

$$\begin{aligned} &v_2 = v_1 + N_1 h,\ N_1 = \frac{v_2 - v_1}{h} = \frac{(L_1 - I_{p+1})v_1}{h} = M_1 v_1, \\ &M_1 = \frac{(L_1 - I_{p+1})}{h} = \begin{bmatrix} 0_{p\times p} & \frac{w(t+h)-w(t)}{h} \\ 0_{1\times p} & 0 \end{bmatrix}, \end{aligned} \tag{13}$$

which gives $v_2 = v_1 + hM_1 v_1$. Therefore, the minimum cost-to-go from $t+h$ can be written as

$$\begin{aligned} &\mathcal{C}_2 = V_{t+h} = \mathcal{C}_2^1 + \mathcal{C}_2^2 + \mathcal{C}_2^3 = \\ &(x_1 + h(Ax_1 + Bu_1 + D_1 v_1))^T P_{t+h}(x_1 + h(Ax_1 + Bu_1 + D_1 v_1)) \\ &+\ v_2^T P_{1(t+h)} v_2 + (x_1 + h(Ax_1 + Bu_1 + D_1 v_1))^T P_{2(t+h)} v_2. \end{aligned} \tag{14}$$

Neglecting higher-order terms we expand the terms in $\mathcal{C}_2$. Using $P_{t+h} = P_t + h\dot{P}_t$ the first term in $\mathcal{C}_2$ can be expanded as

$$\begin{aligned} &\mathcal{C}_2^1 = x_1^T P_t x_1 + h[(Ax_1 + Bw + D_1 v_1)^T P_t x_1 + \\ &x_1^T P_t (Ax_1 + Bw + D_1 v_1) + x_1^T \dot{P}_t x_1]. \end{aligned} \tag{15}$$

Similarly we have

$$\begin{aligned} \mathcal{C}_2^2 &= (v_1 + M_1 v_1 h)^T (P_{1t} + h\dot{P}_{1t})(v_1 + M_1 v_1 h) \\ &= v_1^T P_{1t} v_1 + h[v_1^T M_1^T P_{1t} v_1 + v_1^T P_{1t} M_1 v_1 + v_1^T \dot{P}_{1t} v_1], \end{aligned} \tag{16}$$

$$\begin{aligned} &\mathcal{C}_2^3 = x_1^T P_{2t} v_1 + h[(Ax_1 + Bu_1 + D_1 v_1)^T P_{2t} v_1 + \\ &x_1^T P_{2t} M_1 v_1 + x_1^T \dot{P}_{2t} v_1]. \end{aligned} \tag{17}$$

Therefore, the total cost is given by the sum of the stage cost incurred during the interval $[t, t+h]$, and the cost-to-go from $t+h$ as

$$\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2. \tag{18}$$

To find the stable optimal control u_1, the net-cost will be minimized with respect to u_1. Differentiating $\mathcal{C}$ with respect to u_1 we get

$$2u_1^T R + v_1^T P_{2t} B + 2x_1^T P_t B = 0, \tag{19}$$

$$u_1 = -\underbrace{R^{-1}B^T P_t}_{K_{1t}} x_1 - \underbrace{\frac{1}{2} R^{-1} B^T P_{2t}}_{K_{2t}} v_1. \tag{20}$$

Eq. (20) shows that the optimal control is composed of not only the state feedback term, but also an additional feedback arising from the exogenous input. It will later turn out that P_{2t} depends on the derivative of the exogenous inputs. Following the principle of optimality, the Hamilton–Jacobi (HJ) equation guides us to write

$$V_t = \min_{u_1}(\mathcal{C}_1 + V_{t+h}), \tag{21}$$

$$x_1^T P_t x_1 + v_1^T P_{1t} v_1 + x_1^T P_{2t} v_1 = \min_{u_1}(\mathcal{C}_1 + V_{t+h}). \tag{22}$$

Equating the above equation we compactly write

$$h\left[\sum_{i=1}^{12} \mathcal{E}_i\right] = 0, \tag{23}$$

where explicit expressions of the terms in $\mathcal{E}_i,\ i = 1, \ldots, 12$ are listed in Appendix. As the terms are scalar we have $x_1^T P_t D_1 v_1 + v_1^T D_1^T P_t x_1 = 2x_1^T P_t D_1 v_1$. After a few algebraic computations we can write

$$\sum_{i=1}^{12} \mathcal{E}_i = x_1^T Q_{xx} x_1 + v_1^T Q_{vv} v_1 + x_1^T Q_{xv} v_1 = 0, \tag{24}$$

where,

$$Q_{xx} = Q + P_t A + A^T P_t - P_t B R^{-1} B^T P_t + \dot{P}_t, \tag{25}$$

$$Q_{vv} = S + M_1^T P_{1t} + P_{1t} M_1 + \dot{P}_{1t} + D_1^T P_{2t}, \tag{26}$$

$$Q_{xv} = A^T P_{2t} + P_{2t} M_1 - P_t B R^{-1} B^T P_{2t} + 2P_t D_1 + \dot{P}_{2t}. \tag{27}$$

As the sum of these three quadratic terms is zero, we get the following three matrix differential equations that need to be solved over the finite time-horizon $[0,\ T]$:

$$-\dot{P}_t = Q + P_t A + A^T P_t - P_t B R^{-1} B^T P_t, \tag{28}$$

$$-\dot{P}_{1t} = S + M_1^T P_{1t} + P_{1t} M_1 + D_1^T P_{2t}, \tag{29}$$

$$-\dot{P}_{2t} = A^T P_{2t} + P_{2t} M_1 - P_t B R^{-1} B^T P_{2t} + 2P_t D_1. \tag{30}$$

These three matrix differential equations form the core of this design problem, supplementing the conventional differential Riccati equations (DRE) in LQR theory. The matrix M_1 depends on the rate of change of the disturbance variable w as shown in (13). In the continuous-time setting, we consider the time-step h to go to zero in the limit. Subsequently, we recompute the matrix M_1 as $h \to 0$, and denote it as M_t where,

$$M_t = \begin{bmatrix} 0_{p\times p} & \lim_{h\to 0} \frac{w(t+h)-w(t)}{h} = \frac{dw}{dt} \\ 0_{1\times p} & 0 \end{bmatrix}, \tag{31}$$

using which we obtain the final matrix differential equations stated in Theorem 1. This concludes the derivation of the MDEs with the final conditions $P(T) = Q_f$ and zero matrices of appropriate dimensions for the other variables. The MDEs of P and P_2 are needed to find the gains $K_{1t} = R^{-1}B^T P_t$, and $K_{2t} = \frac{1}{2}R^{-1}B^T P_{2t}$, which are implemented as $u(t) = -K_{1t}x(t) - K_{2t}v(t)$. □

Theorem 1 provides a finite-horizon solution for LQR using the model information and information about the exogenous input. Here, we proposed additional control terms that are dependent on the novel MDE (6). Solving the MDE, we can compute the gains for the terms that use $w(t)$ and its first order derivatives. This new set of coupled MDEs results in an extended LQR version which provably improves the cost minimization compared to only a standard LQR feedback gain implementation. Please note that we start with the optimization objective in a more generalized setting considering a quadratic cost term associated with the exogenous inputs, however, the solution of the MDE associated with P_{1t} do not contribute to the feedback control gains. The MDE computation associated with P_t and P_{2t} requires measurement of the exogenous input and its derivative at all time t. However, derivative information of exogenous inputs may be difficult to

obtain in practice. Considering this fact, we next provide an approximate steady-state solution of the same LQR problem that does not depend on the derivative of w, and try to implement the results of Theorem 1 with limited information of the exogenous inputs. The exogenous input can still be of any nature. Whenever, the exogenous inputs are persistently exciting over long time-horizon, the cost associated with the feedback control is computed with finite time-integral with the steady-state control gains implemented. In the simulation section, one such example is given. It will be shown shortly that this formulation will help us to learn the gains with unknown state matrices using RL/ADP. For this we first partition P_{2t} as

$$P_{2t} = [P_{2at} \;\; P_{2bt}], \tag{32}$$

where $P_{2at} \in \mathbb{R}^{n\times p}$ and $P_{2bt} \in \mathbb{R}^{n\times 1}$. Using (32), we can write the MDE of P_{2a} as

$$-\dot{P}_{2at} = A^T P_{2at} - P_t B R^{-1} B^T P_{2at} + 2P_t D. \tag{33}$$

P_{2bt}, on the other hand, depends on $\dot{w}(t)$ due to the structure of M_t, and therefore, does not converge if the exogenous input is persistently time-varying. In order to provide an approximate steady-state solution, we neglect the contribution arising from P_{2b}, and as P_{2a} converges to a constant matrix, an steady state constant gain K_2 can be computed. This will be shown shortly. Please note that P_{2a} is dependent on the exogenous disturbance matrix D, which is constant, and therefore, we can expect steady-state solution for the gains associated with the exogenous input feedback. Subsequently, the steady state solution in the sense of fixed control gains, is given in the following corollary.

Corollary 1. *The steady-state approximate optimal control gains following the form from Theorem 1 for the system (1) with the objective (2) is given by,*

$$u(t) = -K_1 x(t) - K_2 w(t), \tag{34}$$

$$K_1 = R^{-1}B^T P, \;\; K_2 = \frac{1}{2} R^{-1} B^T P_{2a}, \tag{35}$$

where P and P_{2a} are obtained by solving

$$0 = Q + PA + A^T P - PBR^{-1}B^T P, \tag{36}$$

$$0 = A^T P_{2a} - PBR^{-1}B^T P_{2a} + 2PD. \tag{37}$$

Proof. The infinite horizon solution of P_t is standard in the optimal control literature. We will, therefore, show that the MDE (33) will converge to the solution obtained from the matrix equation (37). Subtracting (37) from (33), and denoting $\tilde{P}_{2at} = P_{2at} - P_{2a}$ we can write,

$$\begin{aligned} -\dot{\tilde{P}}_{2at} &= A^T(P_{2at} - P_{2a}) - [K_{1t}B^T P_{2at} - K_1 B^T P_{2a}] + \\ &\qquad 2(P_t - P)D, \qquad (38)\\ &= A^T \tilde{P}_{2at} - [K_{1t}B^T(P_{2at} - P_{2a}) + \\ &\qquad (K_{1t} - K_1)B^T P_{2a}] + 2\tilde{P}_t D, \qquad (39)\\ &= (A - BK_{1t})^T \tilde{P}_{2at} - (K_{1t} - K_1)B^T P_{2a} + 2\tilde{P}_t D. \qquad (40) \end{aligned}$$

As $t \to \infty$, K_{1t} and P_t converge to K_1 and P, respectively, resulting in the following dynamics

$$-\dot{\tilde{P}}_{2at} = (A - BK_1)^T \tilde{P}_{2at}. \tag{41}$$

As $A - BK_1 \in \mathbb{RH}_\infty$ i.e., Hurwitz, P_{2at} will converge to P_{2a}. □

Remark 1. It is interesting to note the difference in implementation between Theorem 1, and Corollary 1. The solution given by Theorem 1 is the exact time-varying optimal solution, whereas, in Corollary 1 we have approximated the solution by neglecting the contributions from the derivative of the exogenous inputs for the purpose of ease in implementation and also developing RL algorithm in the next section, however, as the solution in Corollary 1 contains the term $K_2 w(t)$, the time varying nature of the exogenous inputs will still influence the closed-loop trajectories. The solution in Corollary 1 will always be optimal with respect to standard LQR solution as the solution of Corollary 1 is the optimal solution of a representative optimal control problem with constant $w(t)$, and can therefore be generalized for time-varying $w(t)$.

This steady-state solution is used next to formulate the RL algorithm that can compute the feedback gains K_1 and K_2 from Corollary 1 without knowing the state matrix A and the exogenous input matrix D. However, we will need to know the control input matrix B for the algorithm developed in the next section.

3. Reinforcement learning control

Learning Problem: With unknown state matrix A, and exogenous input matrix D, *learn* the gains K_1, and K_2 corresponding to the approximate steady-state LQR solutions from Corollary 1 using the trajectory measurements of states, control, and exogenous inputs.

As we consider that we do not have any information about the model matrices A, D, it is theoretically difficult to formulate the learning algorithm with unmeasured exogenous inputs. However, this is not entirely a restrictive design choice as exogenous inputs in control systems may be measured with deployment of high-fidelity sensors. For example, in power systems, the terminal generator bus voltages act as exogenous inputs to generator states [36], and we can measure these bus voltages using phasor measurement units (PMUs) [37]. In another example, for small unmanned aerial systems, anemometers can be installed on drones to measure wind velocity as exogenous inputs [38].

The iterative solution of (36) is computed using Kleinman's algorithm [39]. We append another iterative equation based on (37) to develop model-dependent iterative equations for the coupled matrix equations (36)–(37).

Theorem 2. *Starting with a stabilizing K_{1_0}, i.e., $A - BK_{1_0}$ is Hurwitz, the optimal LQR controller is obtained by using the following steps.*
for *$k = 0, 1, \ldots,$*
1. Solve for P_k from

$$A_{ck}^T P_k + P_k A_{ck} + Q + K_{1k}^T R K_{1k} = 0, A_{ck} = A - BK_{1k}. \tag{42}$$

2. Update K_{1k} as,

$$K_{1(k+1)} = R^{-1}B^T P_k. \tag{43}$$

end
The solution of the above iterative solution will lead to the infinite horizon optimal solution K_1. Then solve for K_2 as:
3. Find P_{2a} by solving

$$A^T P_{2a} - K_{1k}^T B^T P_{2a} + 2P_k D = 0. \tag{44}$$

4. Update K_2 as,

$$K_2 = \frac{1}{2} R^{-1} B^T P_{2a}. \tag{45}$$

Proof. The iterative update and convergence of P_k and K_{1k} directly follow from the Kleinman's algorithm [39]. We, therefore, show that the solution of P_{2a} in (44) (a Sylvester equation) converges to the solution in (37) as $k \to \infty$. The sequence of P_{2ak} for $k = 0, 1, ..$ can be constructed by solving (44) as

$$A^T P_{2ak} - K_{1k}^T B^T P_{2ak} + 2P_k D = 0. \tag{46}$$

Next, considering (37) and (46), we show that P_{2ak} will converge to P_{2a} as $k \to \infty$. Subtracting (37) from (44), we have,

$$0 = A^T(P_{2ak} - P_{2a}) + (K_1 - K_{1k})B^T P_{2a} - K_{1k}^T B^T (P_{2ak} - P_{2a}). \quad (47)$$

Running (42)–(43) in Theorem 2 with $k \to \infty$, P_k and K_{1k} will converge to P and K_1, respectively. Therefore, we can write,

$$0 = A^T(\lim_{k\to\infty}(P_{2ak} - P_{2a})) - K_1^T B^T(\lim_{k\to\infty}(P_{2ak} - P_{2a})) \quad (48)$$

$$0 = (A - BK_1)^T(\lim_{k\to\infty} P_{2ak} - P_{2a}). \quad (49)$$

As $A - BK_1$ is Hurwitz, it follows that $(\lim_{k\to\infty} P_{2ak} - P_{2a}) \notin \mathcal{N}((A - BK_1)^T)$, and therefore P_{2ak} converges to P_{2a} as $k \to \infty$ following Theorem 2. This will also lead to convergence of K_2 directly. This concludes the proof. □

3.1. Formulation of the learning algorithm

The objective is to learn the feedback gains K_1 and K_2 as in Corollary 1 without knowing A and D. An exploration signal $u = u_0$ is used to persistently excite the system (1). This exploration signal, however, should not make the system state trajectories unbounded [17]. We recall the state dynamics (1) as

$$\dot{x} = Ax + Bu_0 + Dw, \quad (50)$$

$$= A_c x + B(K_1 x + u_0) + Dw, \quad (51)$$

where $A_c = A - BK_1$. Continuing with the following computations we can write

$$\frac{d}{dt}(x^T P_k x + x^T P_{2ak} w) = \dot{x}^T P_k x + x^T P_k \dot{x} + \dot{x}^T P_{2ak} w + x^T P_{2ak} \dot{w}. \quad (52)$$

Expanding the first two terms in (52), and using (42) and (43) we get,

$$\begin{aligned}
&\dot{x}^T P_k x + x^T P_k \dot{x} \\
&= x^T[A_{ck}^T P_k + P_k A_{ck}]x + 2(K_{1k}x + u_0)^T B^T P_k x \\
&\quad + 2w^T D^T P_k x, \\
&= -x^T \bar{Q}_{1k} x + 2(K_{1k}x + u_0)^T R K_{1(k+1)} x + 2w^T D^T P_k x,
\end{aligned} \quad (53)$$

where $\bar{Q}_{1k} = Q + K_{1k}^T R K_{1k}$. Expanding the last two terms in (52), and using (44) and (45) we get,

$$\begin{aligned}
&\dot{x}^T P_{2ak} w + x^T P_{2ak} \dot{w} \\
&= (Ax + Bu_0 + Dw)^T P_{2ak} w + x^T P_{2ak} \dot{w}, \\
&= x^T(A^T P_{2ak}) w + x^T P_{2ak} \dot{w} + 2u_0^T R K_{2(k+1)} w \\
&\quad + w^T D^T P_{2ak} w, \\
&= x^T(-2P_k D + K_{1k}^T B^T P_{2ak}) w + x^T P_{2ak} \dot{w} + \\
&\quad 2u_0^T R K_{2(k+1)} w + w^T D^T P_{2ak} w,
\end{aligned} \quad (54)$$

Therefore, revisiting (52) we can write

$$\begin{aligned}
&\frac{d}{dt}(x^T P_k x + x^T P_{2ak} w) = -x^T \bar{Q}_{1k} x + \\
&2(K_{1k}x + u_0)^T R K_{1(k+1)} x + x^T K_{1k}^T B^T P_{2ak} w + x^T P_{2ak} \dot{w} + \\
&2u_0^T R K_{2(k+1)} w + w^T D^T P_{2ak} w.
\end{aligned} \quad (55)$$

Taking integrals on both sides of (55) we finally get Eq. (56) which is given in Box I. The RL algorithm can be constructed by formulating an iterative version of (56), which is given in Algorithm 1.

Algorithm 1 RL algorithm for extended LQR

1. *Data storage: Store* data (x, w and u_0) for interval $(t_1, t_2, \cdots, t_l)$, $t_i - t_{i-1} = T$. There exists a sufficiently large number of sampling intervals for each iteration step such that the rank condition from Remark 2 satisfies. Then *construct* the following matrices,

$$\delta_{xx} = \left[x \otimes x|_{t_1}^{t_1+T}, \quad \cdots \quad , x \otimes x|_{t_l}^{t_l+T}\right]^T, \quad (57)$$

$$\delta_{xw} = \left[x \otimes w|_{t_1}^{t_1+T}, \quad \cdots \quad , x \otimes w|_{t_l}^{t_l+T}\right]^T, \quad (58)$$

$$I_{xx} = \left[\int_{t_1}^{t_1+T}(x \otimes x)d\tau, \quad \cdots \quad , \int_{t_l}^{t_l+T}(x \otimes x)d\tau\right]^T, \quad (59)$$

$$I_{xu_0} = \left[\int_{t_1}^{t_1+T}(x \otimes u_0)d\tau, \quad \cdots \quad , \int_{t_l}^{t_l+T}(x \otimes u_0)d\tau\right]^T. \quad (60)$$

$$I_{wu_0} = \left[\int_{t_1}^{t_1+T}(w \otimes u_0)d\tau, \quad \cdots \quad , \int_{t_l}^{t_l+T}(w \otimes u_0)d\tau\right]^T. \quad (61)$$

$$I_{xw} = \left[\int_{t_1}^{t_1+T}(x \otimes w)d\tau, \quad \cdots \quad , \int_{t_l}^{t_l+T}(x \otimes w)d\tau\right]^T, \quad (62)$$

$$I_{x\dot{w}} = \left[\int_{t_1}^{t_1+T}(x \otimes \dot{w})d\tau, \quad \cdots \quad , \int_{t_l}^{t_l+T}(x \otimes \dot{w})d\tau\right]^T, \quad (63)$$

$$I_{ww} = \left[\int_{t_1}^{t_1+T}(w \otimes w)d\tau, \quad \cdots \quad , \int_{t_l}^{t_l+T}(w \otimes w)d\tau\right]^T. \quad (64)$$

2. *Controller update iteration :* Starting with a stabilizing K_{10}, *Compute* K_1 and K_2 iteratively $(k = 0, 1, \cdots)$ using the following iterative equation

$$\underbrace{\begin{bmatrix}\Theta_k^1 & \Theta_k^2 & \Theta_k^3 & \Theta_k^4 & \Theta_k^5\end{bmatrix}}_{\Theta_k} \begin{bmatrix} vec(P_k) \\ vec(P_{2ak}) \\ vec(K_{1(k+1)}) \\ vec(K_{2(k+1)}) \\ vec(D^T P_{2ak}) \end{bmatrix} = \underbrace{-I_{xx} vec(\bar{Q}_{1k})}_{\Phi_k}. \quad (65)$$

where $\Theta_k^1 = \delta_{xx}$, $\Theta_k^2 = \delta_{xw} - I_{xw}(I_p \otimes K_{1k}^T B^T) - I_{x\dot{w}}$, $\Theta_k^3 = -2I_{xx}(I_n \otimes K_{1k}^T R) - 2I_{xu_0}(I_n \otimes R)$, $\Theta_k^4 = -2I_{wu_0}(I_n \otimes R)$, $\Theta_k^5 = -I_{ww}$.

Terminate the loop when $|P_k - P_{k-1}| \& |P_{2ak} - P_{2a(k-1)}| < \varsigma$, where $\varsigma > 0$ is a small threshold.

3. *Applying K in the system:* Finally, apply $u = -K_1 x - K_2 w$, and remove u_0.

Remark 2 (*Convergence and Stability*)**.** Algorithm 1 is developed based on the iterative algorithm presented in Theorem 2. Therefore, the convergence to optimal and stable feedback gains following Algorithm 1 can be assured based on its equivalence with Theorem 2, which itself is based on the matrix differential equations obtained in Theorem 1. The dynamic programming based approach taken for deriving the matrix differential equations in Theorem 1 guarantees the stability and optimality of the feedback gain solution, the infinite horizon component of which eventually gets manifested as the ADP-based solutions from Algorithm 1.

Remark 3. To accurately compute the learning gain, the underlying notion of *persistency of excitation* associated with system identification and adaptive control is needed to be satisfied. For each $k = 0, 1, 2, \ldots$, it is assumed that there exists a sufficiently large integer $l_k > 0$ signifying large enough sampling intervals, such that $\text{rank}(\Theta_k) = n(n+1)/2 + np + mn + mp + p^2$, which is required to persistently excite the system, and compute unique solutions. This can be satisfied, for example, by utilizing data from at least twice as many sampling intervals as the number of unknowns. We will need the data samples to be rich in the dynamic system behavioral information, and therefore, the exploration signal should considerably excite the underlying unknown dynamics. We have used the sum of sinusoids of varying frequencies selected randomly as an exploration signal. However, the choice of the exploration signal may not be trivial, and different dynamical models with varying degree of complexity may need dedicated exploration signal design. In our experiments, the sum of sinusoidal excitation show sufficiently good performance.

$$x(t+T)^T P_k x(t+T) - x(t)^T P_k x(t) + x(t+T)^T P_{2ak} w(t+T) - x(t)^T P_{2ak} w(t) - \int_t^{t+T} (2(K_{1k}x + u_0)^T R K_{1(k+1)} x) d\tau -$$
$$\int_t^{t+T} (x^T K_{1k}^T B^T P_{2ak} w + x^T P_{2ak} \dot{w} + 2u_0^T R K_{2(k+1)} w + w^T D^T P_{2ak} w) d\tau = -\int_t^{t+T} x^T \bar{Q}_{1k} x d\tau. \quad (56)$$

Box I.

Remark 4. The learning algorithm requires $\dot{w}$ information to construct $I_{x\dot{w}}$. However, the designed controller only depends on w, as proved in Corollary 1. Therefore, we may learn the gains K_1 and K_2 in a controlled environment where $\dot{w}$ can be made available, and then implement them for other applications where $\dot{w}$ is not available. For this we use the fact that the approximate steady-state gain K_2 is dependent only on the exogenous input matrix D, and not on the exogenous input itself. The learning algorithm relieves the requirement of knowledge about D.

The algorithm compensates for the inadequacies of the robust ADP approaches [33,35] which compute only $u = -K_1 x$ in presence of the exogenous inputs, whereas the proposed extended method can learn a more optimized approximated feedback gain. We, next, validate our design by few numerical examples.

4. Numerical examples

Consider the following third-order CTLTI model:

$$A = \begin{bmatrix} -1.4 & 0.2 & -0.1 \\ -0.2 & -0.8 & -0.3 \\ 0.1 & -0.1 & -0.9 \end{bmatrix}, B = \begin{bmatrix} 0.1 & 0.8 \\ 1.1 & 0.3 \\ 0.9 & 0.5 \end{bmatrix}, D = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \quad (67)$$

We consider two different types of exogenous inputs:
E1. An exponentially decaying input, $w(t) = e^{-t}$, and
E2. A sinusoidal exogenous input, $w(t) = 1 + 0.1\sin(t)$.

4.1. Model-based control

We solve the MDEs given in Theorem 1 for a 15 s time window with 0.01 s time step. We consider $Q = 20I_3$, $R = I_2$, $S = 1$ and $Q_f = I_3$. The states are initialized at the origin. The DRE (5) leads to a convergent infinite horizon LQR solution. The feedback gain K_1 is found to be

$$K_1 = \begin{bmatrix} -0.353 & 3.107 & -2.009 \\ 2.795 & -0.247 & 1.829 \end{bmatrix}. \quad (68)$$

The iterations for computing P_2 are shown in Fig. 1 for both exogenous inputs E1 and E2. It can be seen from Fig. 1 that the matrix P_{2a} (first column of P_2) converges to a steady state solution, whereas the matrix P_{2b} (second column of P_2) shows a variation that is similar to the exogenous input which validates its dependency on the derivative of the exogenous inputs. The matrix P_{2a} converges to $[1.781, 1.888, -1.202]^T$ for both E1 and E2 as this part of P_2 depends only on the A, B and D. Please note that in Fig. 1, the integration has been performed backward in time. The plots are also vindicate our solution structure as given by Theorem 1, where the matrix P_{2bt} is dependent on the evolution of the exogenous input derivatives, and we can see from the figures for both the exogenous inputs, it follows such trajectories in its evolution.

The closed-loop state trajectories with the extended LQR control are plotted in Fig. 2. This figure shows a comparison between the classical LQR control and the proposed extended LQR control, from which it is clear that the latter results in better dynamic performance. The minimization of the net cost associated with

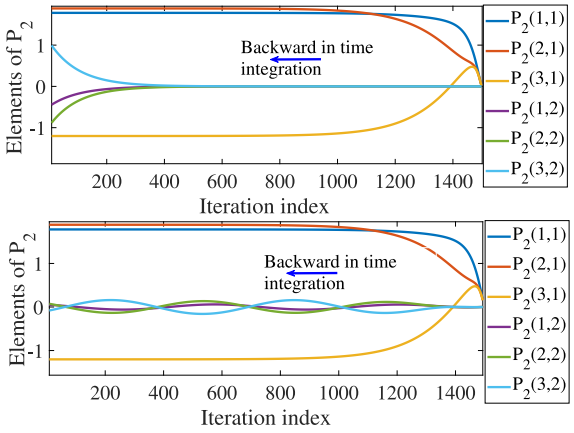

Fig. 1. Convergence of the elements of P_{2a}. Elements of P_{2b} varies based on the exogenous input (top panel for E1 and bottom panel for E2), the time integration of MDEs in Theorem 1 is performed backward in time starting at $t = 15$ s.

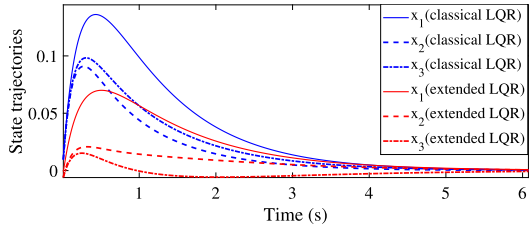

Fig. 2. Comparison of closed-loop state trajectories with classical LQR (blue) and proposed extended LQR (red) for the exogenous input E1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Comparison of net quadratic cost (state and control) between classical and extended LQR designs based on Theorem 1 for the simulation example.

	Classical LQR	Extended LQR
Exogenous input E1	1.0535	0.7162
Exogenous input E2	37.442	24.73

state deviations and the control efforts can be seen from Table 1 for both the exogenous inputs. Table 1 shows that the proposed extended LQR guarantees minimization of the net cost function, producing lower net cost than the classical LQR. Fig. 3 shows the net savings in the cost following the extended LQR design using Theorem 1. An interesting point to note is that the cost function also shows similar variation as the exogenous inputs.

When the exogenous input model is not known, one may implement the steady state solution given in Corollary 1 to find K_1 and K_2. The solution obtained by solving the matrix equations in Corollary 1 via an iterative approach as in Theorem 2 converges to steady state feedback gains. The feedback gain K_1 is same as in (68), and the gain $K_2 = [0.5869, 0.6953]^T$. The comparative costs

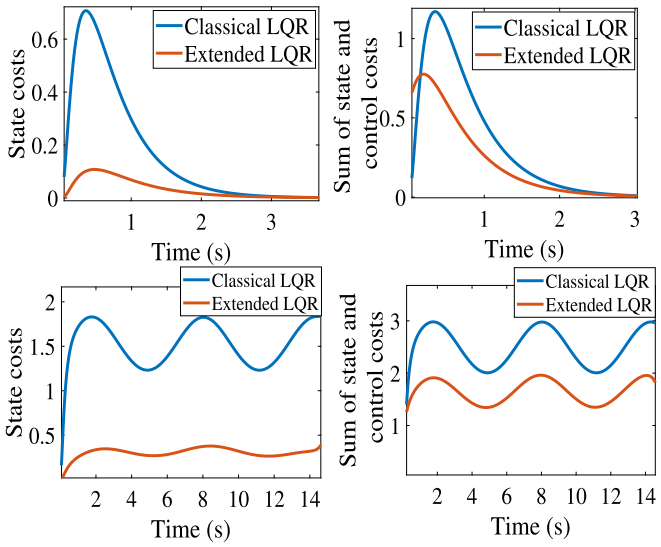

Fig. 3. Savings in the cost objective following the extended LQR design than the classical LQR.

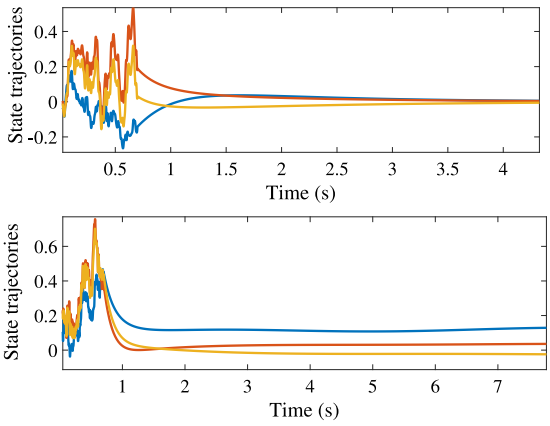

Fig. 4. Exploration (up till 0.7 s) and control implementation for the RL control design (Top panel is for E1 and the bottom panel is for E2).

Table 2
Comparison of net quadratic cost (state and control) between classical and extended LQR infinite horizon solutions based on Corollary 1 for the simulation example.

	Classical LQR	Extended LQR
Exogenous input E1	1.0535	0.7203
Exogenous input E2.	37.425	24.83

for the steady state solutions are tabulated in Table 2, showing effectiveness.

4.2. Reinforcement learning control

Finally, we test the RL Algorithm 1 for computing the infinite horizon feedback gains K_1 and K_2 corresponding to P and P_{2a}. The objective is to recover the gains corresponding to the model based analysis shown in the previous sub-section from the measurements. The system trajectories are gathered during the exploration with a time-step of 0.01 s. Here $n = 3, m = 2$ and $p = 1$. Therefore, following Remark 2, around 40 time samples are required to compute the unique optimal solutions. We perform the exploration for 0.7 s, from which the measurements of $x(t)$, $u_0(t)$, and $w(t)$ are stored as in Step 1 of

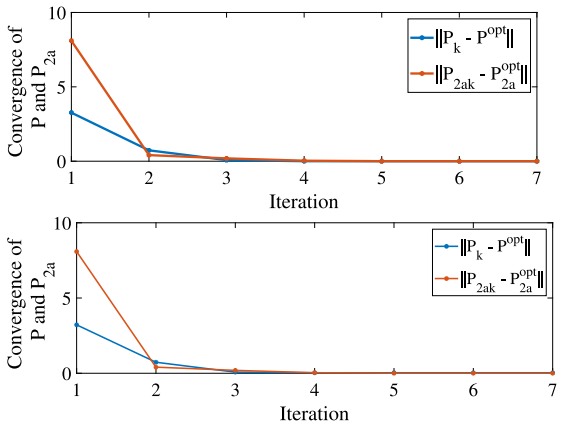

Fig. 5. Convergence of P and P_{2a} to the infinite horizon optimal solutions (Top panel is for E1 and the bottom panel is for E2).

Algorithm 1. The exploration has been performed using the sum of sinusoids with random frequencies, which makes sure that the system is being persistently excited. The exploration phase for both types of exogenous inputs is shown in Fig. 4. By gathering the data matrices, the learning algorithm compensates for the lack of knowledge about the state matrix A and the disturbance matrix D.

Thereafter, P and P_{2a} and the corresponding control gains are iteratively computed as shown in Fig. 5. The solutions converge to the ideal optimal infinite horizon solution within 4 iterations. Please note that these ideal optimal solutions correspond to Theorem 2 and Corollary 1, and are only used here for comparison purpose to substantiate our theoretical results. The optimal closed-loop performance is shown in Fig. 4, where the initial learning phase can be identified, and then the learned extended LQR controller is implemented. All of these experiments substantiate our theoretical developments, and show the substantial performance improvement of the extended controller over classical designs.

5. Conclusions and future work

This paper developed a novel extended LQR control design for continuous time LTI systems perturbed with exogenous inputs. The design guarantees the net state and control cost to be lower than that from the classical LQR design. Both model-based and model-free versions are reported. The model based design guarantees the net cost minimization using dynamic programming. The model-free design is based on reinforcement learning that can compute the steady-state LQR gains using measurements of states, control inputs, and the exogenous inputs. Convergence and stability guarantees of the RL algorithm are established. Numerical simulations are provided to illustrate the effectiveness of the design in both model-based and model-free settings.

Future work includes investigation of learning algorithms with limited knowledge of exogenous inputs. Research efforts can also be directed toward formulating the distributed version of the proposed extended learning-based LQR designs. Application of the algorithm to navigation of small aerial systems in windy environments, and designing wide-area damping controls for large-scale power systems to improve its dynamic performance following contingencies can provide practical examples for deployment of such techniques.

CRediT authorship contribution statement

Sayak Mukherjee: Conceptualization, Methodology, Software, Investigation, Validation, Writing - original draft. **He Bai:** Methodology, Validation, Investigation, Supervision, Writing - review & editing. **Aranya Chakrabortty:** Methodology, Resources, Supervision, Investigation, Writing - review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The work of first author (during his Ph.D.) and third author were partially supported by the National Science Foundation, United States grant EECS 1940866. The work of the second author was supported in part by the National Science Foundation, USA under award number 1925147.

Appendix. Terms $\mathcal{E}_i$, $i = 1, \ldots, 12$ in Theorem 1

$$\mathcal{E}_1 = x_1^T Q x_1, \tag{A.1}$$

$$\mathcal{E}_2 = u_1^T R u_1 = x_1^T P_t B R^{-1} B^T P_t z + x_1^T P_t B R^{-1} B^T P_{2t} v_1 + v_1^T P_{2t} B R^{-1} B^T P_t x_1 + v_1^T P_{2t} B R^{-1} B^T P_{2t} v_1, \tag{A.2}$$

$$\mathcal{E}_3 = v_1^T S v_1, \tag{A.3}$$

$$\mathcal{E}_4 = v_1^T M_1^T P_{1t} v_1, \tag{A.4}$$

$$\mathcal{E}_5 = v_1^T P_{1t} M_1 v_1, \tag{A.5}$$

$$\mathcal{E}_6 = v_1^T \dot{P}_t^1 v_1, \tag{A.6}$$

$$\mathcal{E}_7 = x_1^T A^T P_{2t} v_1 + v_1^T D_1^T P_{2t} v_1 - x_1^T P_t B R^{-1} B^T P_{2t} v 1 - v_1^T P_{2t} B R^{-1} B^T P_{2t} v_1, \tag{A.7}$$

$$\mathcal{E}_8 = x_1^T P_{2t} M_1 v_1, \tag{A.8}$$

$$\mathcal{E}_9 = x_1^T \dot{P}_t^2 v_1, \tag{A.9}$$

$$\mathcal{E}_{10} = x_1^T A^T P_t x_1 + v_1^T D_1^T P_t x_1 - x_1^T P_t B R^{-1} B^T P_t x_1 - v_1^T P_{2t} B R^{-1} B^T P_t x_1, \tag{A.10}$$

$$\mathcal{E}_{11} = x_1^T P_t A x_1 + x_1^T P_t D_1 v_1 - x_1^T P_t B R^{-1} B^T P_t x_1 - x_1^T P_t B R^{-1} B^T P_{2t} v 1, \tag{A.11}$$

$$\mathcal{E}_{12} = x_1^T \dot{P}_t x_1. \tag{A.12}$$

References

[1] E. Menguy, J. Boimond, L. Hardouin, J. Ferrier, Just-in-time control of timed event graphs: update of reference input, presence of uncontrollable input, IEEE Trans. Automat. Control 45 (11) (2000) 2155–2159.
[2] J.-M. Yang, S.W. Kwak, Corrective control of asynchronous machines with uncontrollable inputs: application to single-event-upset error counters, IET Control Theory Appl. 4 (11) (2010) 2454–2462.
[3] Abhinav Kumar Singh, Bikash C. Pal, An extended linear quadratic regulator for LTI systems with exogenous inputs, Automatica 76 (2017) 10–16.
[4] J. Han, From PID to active disturbance rejection control, IEEE Trans. Ind. Electron. 56 (3) (2009) 900–906.
[5] B.-Z. Guo, Z.-L. Zhao, On the convergence of an extended state observer for nonlinear systems with uncertainty, Systems Control Lett. 60 (2011) 420–430.
[6] B.-Z. Guo, F.-F. Jin, The active disturbance rejection and sliding mode control approach to the stabilization of the euler-bernoulli beam equation with boundary input disturbance, Automatica 49 (2013) 2911–2918.
[7] Wen-Hua Chen, Disturbance observer based control for nonlinear systems, IEEE/ASME Trans. Mechatronics 9 (4) (2004) 706–710.
[8] W. Chen, J. Yang, L. Guo, S. Li, Disturbance-observer-based control and related methods—An overview, IEEE Trans. Ind. Electron. 63 (2) (2016) 1083–1095.
[9] A. Chakrabortty, M. Arcak, Robust stabilization and performance recovery of nonlinear systems with unmodeled dynamics, IEEE Trans. Automat. Control 54 (6) (2009) 1351–1356.
[10] Mario Rotea, The generalized H_2 control problem, Automatica 29 (1993) 373–385.
[11] Ka Cheok, Nan Loh, A ball balancing demonstration of optimal and disturbance-accomodating control, IEEE Control Syst. Mag. 7 (1) (1987) 54–57.
[12] C. Johnson, Accomodation of external disturbances in linear regulator and servomechanism problems, IEEE Trans. Automat. Control 16 (6) (1971) 635–644.
[13] B. Gao, J. Hong, S. Yu, H. Chen, Linear-quadratic output regulator with disturbance rejection: Application to vehicle launch control, in: 2017 American Control Conference (ACC), 2017, pp. 1960–1965.
[14] Bingzhao Gao, Jinlong Hong, Ting Qu, Shuyou Yu, Hong Chen, An output regulator with rejection of time-varying disturbance: Experimental validation on clutch slip control, IEEE Trans. Control Syst. Technol. 28 (3) (2019) 1158–1167.
[15] R.S. Sutton, A.G. Barto, Reinforcement learning - An introduction, MIT press, Cambridge, 1998, 1998.
[16] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, F.L. Lewis, Adaptive optimal control for continuous-time linear systems based on policy iteration, Automatica 45 (2009) 477–484.
[17] Y. Jiang, Z.-P. Jiang, Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics, Automatica 48 (2012) 2699–2704.
[18] K.G. Vamvoudakis, Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach, Systems Control Lett. 100 (2017) 14–20.
[19] B. Kiumarsi, K.G. Vamvoudakis, H. Modares, F.L. Lewis, Optimal and autonomous control using reinforcement learning: A survey, IEEE Trans. Neural Netw. Learn. Syst. (2018).
[20] Sayak Mukherjee, He Bai, Aranya Chakrabortty, Reduced-dimensional reinforcement learning control using singular perturbation approximations, Automatica 126 (2021) 109451.
[21] Maryam Fazel, Rong Ge, Sham Kakade, Mehran Mesbahi, Global convergence of policy gradient methods for the linear quadratic regulator, in: International Conference on Machine Learning, PMLR, 2018, pp. 1467–1476.
[22] Zhong-Ping Jiang, Tao Bian, Weinan Gao, et al., Learning-Based Control: A Tutorial and Some Recent Results, Now Publishers, 2020.
[23] Claudio De Persis, Pietro Tesi, Formulas for data-driven control: Stabilization, optimality, and robustness, IEEE Trans. Automat. Control 65 (3) (2019) 909–924.
[24] Paul Frihauf, Miroslav Krstic, Tamer Başar, Finite-horizon LQ control for unknown discrete-time linear systems via extremum seeking, Eur. J. Control 19 (5) (2013) 399–407.
[25] Sei Zhen Khong, Dragan Nešić, Miroslav Krstić, Iterative learning control based on extremum seeking, Automatica 66 (2016) 238–245.
[26] Shu-Jun Liu, Miroslav Krstic, Tamer Başar, Batch-to-batch finite-horizon LQ control for unknown discrete-time linear systems via stochastic extremum seeking, IEEE Trans. Automat. Control 62 (8) (2016) 4116–4123.
[27] Nicholas M. Boffi, Stephen Tu, Nikolai Matni, Jean-Jacques E. Slotine, Vikas Sindhwani, Learning stability certificates from data, 2020, arxiv preprint arXiv:2008.05952.
[28] Sayak Mukherjee, Thanh Long Vu, Reinforcement learning of structured control for linear systems with unknown state matrix, 2020, arxiv preprint arXiv:2011.01128.
[29] Salar Fattahi, Nikolai Matni, Somayeh Sojoudi, Efficient learning of distributed linear-quadratic control policies, SIAM J. Control Optim. 58 (5) (2020) 2927–2951.
[30] Wentao Tang, Prodromos Daoutidis, Distributed adaptive dynamic programming for data-driven optimal control, Systems Control Lett. 120 (2018) 36–43.
[31] Andrew Lamperski, Computing stabilizing linear controllers via policy iteration, in: 2020 59th IEEE Conference on Decision and Control (CDC), IEEE, 2020, pp. 1902–1907.
[32] Florian Dörfler, Jeremy Coulson, Ivan Markovsky, Bridging direct & indirect data-driven control formulations via regularizations and relaxations, 2021, arxiv preprint arXiv:2101.01273.

[33] Y. Jiang, Z.P. Jiang, Robust adaptive dynamic programming, in: F. Lewis, D. Liu (Eds.), Reinforcement Learning and Approximate Dynamic Programming for Feedback Control, IEEE Press, 2013.
[34] Biao Luo, Huai-Ning Wu, Tingwen Huang, Off-policy reinforcement learning for H_∞ control design, IEEE Trans. Cybern. 45 (1) (2014) 65–76.
[35] Sayak Mukherjee, He Bai, Aranya Chakrabortty, On robust model-free reduced-dimensional reinforcement learning control for singularly perturbed systems, in: 2020 American Control Conference (ACC), IEEE, 2020, pp. 3914–3919.
[36] Peter W. Sauer, Mangalore Anantha Pai, Power System Dynamics and Stability, Vol. 101, Wiley Online Library, 1998.
[37] Sayak Mukherjee, Aranya Chakrabortty, He Bai, Atena Darvishi, Bruce Fardanesh, Scalable designs for reinforcement learning-based wide-area damping control, IEEE Trans. Smart Grid (2021).
[38] Paolo Bruschi, Massimo Piotto, F. Dell'Agnello, J. Ware, N. Roy, Wind speed and direction detection by means of solid-state anemometers embedded on small quadcopters, Procedia Eng. 168 (2016) 802–805.
[39] D. Kleinman, On an iterative technique for Riccati equation computations, IEEE Trans. Automat. Control 13 (1) (1968) 114–115.