Extended adaptive optimal control of linear systems with unknown dynamics using adaptive dynamic programming

Minggang Gan | Jingang Zhao | Chi Zhang

State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing, China

Correspondence
Jingang Zhao, State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China.
Email: zhaojingang521@126.com

Funding information
International Graduate Exchange Program of Beijing Institute of Technology; National Natural Science Foundation of China, Grant/Award Number: 61673065

Abstract
The extended infinite horizon optimal control problem of continuous time linear systems with unknown dynamics is investigated in this paper. This optimal control problem can be solved using the corresponding extended algebraic Riccati equation. A new policy iteration algorithm is proposed to approximate the solution of the extended algebraic Riccati equation when the system dynamics are known. The convergence of the proposed algorithm is proved. Based on the proposed policy iteration algorithm, an online adaptive dynamic programming (ADP) algorithm is developed to find the solution to the extended infinite horizon optimal control problem of unknown continuous time linear systems. The convergence of the online ADP algorithm is analyzed. Finally, two simulation examples are given to demonstrate the effectiveness of the developed online ADP algorithm.

KEYWORDS
adaptive dynamic programming, continuous time, linear systems, optimal control, unknown dynamics

1 INTRODUCTION
The purpose of an optimal control problem is to find an optimal control law that minimizes a predefined performance index function [1]. The optimal control problem has received significant attention in past decades because it is not only academically challenging, but also of practical significance [2,3]. This paper will mainly study the optimal control problem of continuous time linear systems with unknown dynamics [4,5].

According to traditional optimal control theory [1], the solution to the infinite horizon optimal control problem of continuous time linear systems can be found by solving the corresponding algebraic Riccati equation. When system dynamics are known, an iterative technique has been proposed to solve the corresponding algebraic Riccati equation [6]. However, it is generally difficult or impossible to model the systems accurately in practical applications, such as motion systems [7], helicopter systems [8], and wastewater treatment processes [9]. Therefore, the system dynamics are usually completely unknown or partially unknown [10–12]. The corresponding algebraic Riccati equation is thus extremely difficult to solve without the precise knowledge of system dynamics. To overcome this dilemma, the model-free optimal control method has been of considerable interest to the control systems community in recent years [13–15]. ADP is a bio-logically inspired approximate intelligent method and can handle optimization problems with model uncertainty or model unknown [16–24]. For continuous time linear systems, the authors of [25] propose a policy iteration-based ADP method to solve the optimal control for partially...
unknown continuous time linear systems. In [26], the authors solve the continuous time linear systems with completely unknown system dynamics using an online policy iteration ADP method.

However, the performance index function in existing results is the quadratic function of system states and control inputs. They do not contain the product of system states and control inputs. In practical applications, such as mechatronic vehicle suspension systems [27] and vehicle dynamic systems [28], the product of system states and control inputs has an important effect on its optimal control design. Therefore, the extended infinite horizon optimal control problem of continuous time linear systems with unknown system dynamics is investigated in this paper. The extended infinite horizon optimal control problem is that the performance index function contains not only the quadratic function of system states and control inputs, but also the product of them. Utilizing the advantage of ADP in model-free optimization control, this paper will develop an online ADP algorithm to solve the extended infinite horizon optimal control problem of continuous time linear systems with unknown system dynamics.

The main contributions of this paper are described as follows. First, to the best of our knowledge, a new policy iteration algorithm to solve extended algebraic Riccati equation is presented when system dynamics are known. Second, based on the proposed new policy iterative algorithm, we develop an online ADP algorithm to solve extended infinite horizon optimal control problem of continuous time linear systems. Third, the developed online ADP algorithm does not require prior knowledge of system dynamics. It utilizes online measured inputs and states data to achieve optimal control of the unknown system. This will provide a new idea to solve some practical problems, such as mechatronic vehicle suspension systems.

The remaining sections are outlined as follows. Section 2 presents the background and the problem formulation. In Section 3, when system dynamics are known, an iterative technique for solving the extended algebraic Riccati equation is shown. Section 4 presents an online ADP algorithm to solve the extended algebraic Riccati equation without requiring the prior knowledge of the system dynamics. Two simulation examples are given to demonstrate the effectiveness of the developed online ADP algorithm. Section 5 concludes this paper and gives directions for future research.

**Notation:** In this paper, $R$ denotes the sets of real numbers; $\| \cdot \|$ denotes the Euclidean norm for vectors; $\otimes$ denotes the Kronecker product; $\text{vec}(S) = [s_1^T\ s_2^T\ \ldots\ \ s_m^T]^T$, where $S \in R^{m \times n}$, $s_i \in R^n$ ($s = 1, 2, \ldots, m$) represents the columns of $S$; $I_{n \times n}$ denotes $n \times n$ identify matrix; $\varepsilon$ denotes small positive number.

## 2 Problem Formulation and Preliminaries

Consider the following continuous time linear time-invariant system

$$\dot{x}(t) = Ax(t) + Bu(t)$$ (1)

where $x(t) \in R^{n \times 1}$ is the system state variables, $u(t) \in R^{m \times 1}$ is the system control input variables; system drift dynamics matrix $A \in R^{n \times n}$ and input matrix $B \in R^{n \times m}$ are unknown constant matrices with appropriate dimensions. For simplicity, $x(t), u(t)$ can be abbreviated as $x, u$.

Different from the existing research, in this paper, the performance index function is defined as follows

$$J = \int_{0}^{\infty} (x^TQx + 2x^TNu + u^TRu) dt$$ (2)

where the conditions that $Q = Q^T \succeq 0, R = R^T > 0$ and $Q - NR^{-1}N^T \succeq 0$ are satisfied.

**Remark 1.** Note that Equation (2) contains the product of system state variables $x$ and control input variables $u$. This is the main difference with the existing research results.

**Assumption 1.** The pair $(A, B)$ is stabilizable.

**Assumption 2.** $(Q - NR^{-1}N^T, A - BR^{-1}N^T)$ is observable.

**Problem 1. Extended infinite horizon optimal control problem**

For the continuous time linear optimal control problem, our objective is to find an optimal state feedback law in the following form

$$u = -Kx(t)$$ (3)

which minimizes the performance index function (2). Where $K$ denotes the state feedback gain matrix.

Based on the traditional optimal control theory [1], when $A$ and $B$ are precisely known, the solution to this optimal control problem can be found by solving the following extended algebraic Riccati equation

$$A^T P + PA - (PB + N)R^{-1}(B^T P + N^T) + Q = 0$$ (4)

Based on the conditions and assumptions mentioned above, $P'$ is the unique positive definite symmetric solution of the extended algebraic Riccati Equation (4). Then,
the optimal state feedback gain matrix $K^*$ in (3) can thus be given by

$$K^* = R^{-1}(B^TP^* + N^T)$$  \hspace{1cm} (5)

In addition, $P^*$ has the following property

$$J(x_0; u^*) = \min_u J(x_0; u) = x_0^TP^*x_0$$  \hspace{1cm} (6)

It is easy to see from the third item on the left side of Equation (4) that (4) is nonlinear in $P$. Consequently, it is generally difficult to obtain $P^*$ by solving (4), especially for high-dimensional matrices.

When Remark 1 is not satisfied, that is $N = 0$, [6] has given an efficient algorithm to numerically approximate the solution of (4). Inspired by the efficient algorithm in reference [6], we will develop an iterative technique to numerically approximate the solution of the extended algebraic Riccati Equation (4) in this paper.

3 A NEW POLICY ITERATION ALGORITHM FOR SOLVING THE EXTENDED ALGEBRAIC RICCATI EQUATION

In this section, we firstly give the new policy iteration algorithm by the following theorem. Next, we prove the correctness of the theorem.

**Theorem 1.** $K_0 \in \mathbb{R}^{m \times n}$ is chosen such that the matrix $A_0 = A - BK_0$ is Hurwitz, and let $P_k$ be the unique positive definite symmetric solution of the linear algebraic equation

$$A_k^TP_k + P_kA_k + Q + K_k^TRK_k - K_k^TN^T - NK_k = 0$$  \hspace{1cm} (7)

where $K_k$, $A_k$ with $k = 0, 1, 2, 3, \ldots$ are given recursively by

$$K_{k+1} = R^{-1}(B^TP_k + N^T)$$

$$A_k = A - BK_k$$  \hspace{1cm} (8)

Then, the following properties hold

(i) $A_k = A - BK_k$ is Hurwitz;
(ii) $P^* \leq P_{k+1} \leq P_k$;
(iii) $\lim_{k \to \infty} K_k = K^*$, $\lim_{k \to \infty} P_k = P^*$.

**Proof.** Suppose that $u_k = -K_kx = -R^{-1}B^TP_kx$ is an arbitrary feedback law. If $u_k$ is applied to the system (1), the resulting cost (2) can be written as

$$J(x_0; u_k) = x_0^TP_kx_0$$  \hspace{1cm} (10)

where $P_k$ is defined as the cost matrix associated with the feedback gain matrix $K_k$ and is given as follow

$$P_k = \int_0^\infty e^{(A-BK_k)t}(Q + 2NK_k + K_k^TRK_k)e^{(A-BK_k)\tau}dt$$  \hspace{1cm} (11)

where $P_k$ is finite if and only if the closed loop system matrix $A - BK_k$ has eigenvalues with negative real parts. In this case, $P_k$ is the unique positive definite symmetric solution of the following linear equation [6]

$$(A - BK_k)^TP_k + P_k(A - BK_k) + Q + K_k^TRK_k - K_k^TN^T - NK_k = 0$$  \hspace{1cm} (12)

Without loss of generality, we assume that the system terminal time is $T$ (where we may have $T = \infty$ in this case). Then, Equation (4) becomes the following finite horizon associated Riccati equation

$$\dot{P}_k = -A_k^TP_k - P_kA_k - Q - K_k^TRK_k + K_k^TN^T + NK_k$$  \hspace{1cm} (13)

Substituting (8) into (13) and making some mathematical transformation, we can obtain the following linear differential equation

$$\dot{P}_k = -A_k^TP_k - P_kA_k - Q - K_k^TRK_k + K_k^TN^T + NK_k$$  \hspace{1cm} (14)

Thus, the cost matrix $P_k$ satisfies the linear differential Equation (14).

Now, assume that we have a control law $u_{k+1} = -K_{k+1}x$. The cost matrix corresponding to the control law is $P_{k+1}$. Then, based on the previous discussion, $P_{k+1}$ satisfies the linear differential equation

$$\dot{P}_{k+1} = -A_{k+1}^TP_{k+1} - P_{k+1}A_{k+1} - Q - K_{k+1}^TRK_{k+1} + K_{k+1}^TN^T + NK_{k+1}$$  \hspace{1cm} (15)

Using the notation $A_k = A - BK_k$, $A_{k+1} = A - BK_{k+1}$, and writing $A_k = A_{k+1} - B(K_k - K_{k+1})$. Then substituting $A_k = A_{k+1} - B(K_k - K_{k+1})$ into (15) yields

$$\dot{P}_k = -A_k^TP_k - P_kA_{k+1} - Q - K_k^TRK_k + (K_k - K_{k+1})^TB^TP_k + P_kB(K_k - K_{k+1}) + K_k^TN^T + NK_k$$  \hspace{1cm} (16)

Define $\delta P = P_k - P_{k+1}$, then $\delta \dot{P} = \dot{P}_k - \dot{P}_{k+1}$. Accordingly, subtracting Equation (16) from Equation (15) yields
$$\delta \dot{P} = -A_k^T \delta P - \delta PA_k + K_k^T R K_k + K_k^T R K_{k+1}$$

$$\begin{equation}
(K_k - K_{k+1})^T B^T P_k + P_k B (K_k - K_{k+1}) +
(K_k^T R K_k + K_k^T R K_{k+1}) N^T + N (K_k - K_{k+1})
\end{equation}$$

(17)

Now, we add and subtract the term $(K_k - K_{k+1})^T R K_k + K_k^T R K_{k+1}$ from the right side of Equation (17) to obtain

$$\delta \dot{P} = -A_k^T \delta P - \delta P A_k - (K_k - K_{k+1})^T R (K_k - K_{k+1}) +
(K_k - K_{k+1})^T (B_k^T P_k - R K_{k+1} + N^T) +
(P_k B - K_k^T R + N)(K_k - K_{k+1})$$

(18)

Then, the solution of Equation (18) with the boundary condition $\delta P = 0$ is given by [29].

$$\begin{equation}
\delta P = \int_0^T e^{(A - BK_{k+1})^T} [K_k - K_{k+1}]^T R (K_k - K_{k+1}) -
(K_k - K_{k+1})^T (B_k^T P_k - R K_{k+1} + N^T) -
(P_k B - K_k^T R + N)(K_k - K_{k+1})] e^{(A - BK_{k+1})^T} dt
\end{equation}$$

(19)

where $T$ can be equal to infinity, $\delta P = P_k - P_{k+1}$. Then, based on Equation (8) and (19), we can obtain

$$\begin{equation}
\delta P = \int_0^T e^{(A - BK_{k+1})^T} \{[K_k - K_{k+1}]^T R (K_k - K_{k+1})] +
\delta P - \delta P A_k \}
\end{equation}$$

(20)

Based on Equation (11), the cost matrix $P_0$ associated with $K_0$ is given as follow

$$P_0(t) = \int_t^\infty e^{\xi(T-\tau)} (Q + 2 N K_0 + K_0^T R K_0) e^{\xi(\tau-\tau)} d\tau$$

(21)

Let $\xi = t_1 - t + \tau$, where $t_1$ is arbitrary, we can obtain

$$P_0(t) = \int_t^\infty e^{\xi(t_1-\tau)} (Q + 2 N K_0 + K_0^T R K_0) e^{\xi(\tau-\tau)} d\tau = P_0(t_1)$$

(22)

It can be seen from Equation (21) and (22) that $P_0(t)$ is a constant matrix independent of $t$.

Since $K_0 \in R_k$ is chosen such that the matrix $A_0 = A - BK_0$ is Hurwitz, then $\|P_0(t)\| < \infty$ and $P_0(t)$ satisfies (7) or (14) with $k = 0$. Now let $K_1 = R^{-1}(B^T P_0 + N^T)$, based on (20), we can obtain $P_k \leq P_0$. Hence $P_1(t)$ is also bounded below and therefore has finite norm. Thus, $A_1 = A - BK_1$ is Hurwitz, and $P_1(t)$ satisfies (7) or (14) with $k = 1$. Similarly, repeating the above argument for $k = 2, 3, 4 \ldots \ldots$, we can obtain the desired results (i) and (ii).

According to the theorem on monotonic convergence of positive operators in [30] there exists $\lim_{k \to \infty} P_k = P_\infty$. Thus, by taking the limit of (7) as $k \to \infty$, we can obtain

$$A_k^* P_\infty + P_\infty A_k + Q + K_k^T R K_k - K_k^T N^T - N K_\infty = 0$$

(23)

Since $P'$ is the unique positive definite symmetric solution of (23), $P_\infty = P'$. Based on (8), there exists $\lim_{k \to \infty} K_k = K^*$. Thus, we can obtain the desired result (iii).

The proof of Theorem 1 is completed.

Remark 2. The linear algebraic Equation (7) is linear in $P_k$, one can iteratively solving $P_k$ by (7) and recursively updating $K_k$ by (8) when $A$ and $B$ are known. Then, the solution of the nonlinear Equation (4) is thus numerically approximated.

Next, based on Theorem 1, we will give an offline policy iteration algorithm to solve the optimal control Problem 1.

**Algorithm 1. Offline policy iteration algorithm for solving the optimal control Problem 1:**

**Step 1:** Given a stabilising feedback gain matrix $K_0$;

**Step 2:** Solve $P_k$ by the following equation

$$\begin{equation}
(A - BK_k) P_k + P_k (A - BK_k) + Q + K_k^T R K_k -
K_k^T N^T - N K_k = 0
\end{equation}$$

(24)

**Step 3:** Solve $K_{k+1}$ by

$$K_{k+1} = R^{-1}(B^T P_k + N^T)$$

(25)

**Step 4:** Let $k \leftarrow k + 1$, if $\|P_{k+1} - P_k\| \leq \varepsilon$ for $k \geq 0$, go to Step 5; else continue Step 2 and Step 3. Where $\varepsilon$ is a predefined small threshold.

**Step 5:** Use $u = -K_{k+1} x$ as the approximated optimal control input of the Problem 1.

**Algorithm 1.**

**Step 1:** Given a stabilising feedback gain matrix $K_0$;

**Step 2:** Solve $P_k$ by the following equation

$$\begin{equation}
(A - BK_k) P_k + P_k (A - BK_k) + Q + K_k^T R K_k -
K_k^T N^T - N K_k = 0
\end{equation}$$

(24)

**Step 3:** Solve $K_{k+1}$ by

$$K_{k+1} = R^{-1}(B^T P_k + N^T)$$

(25)

**Step 4:** Let $k \leftarrow k + 1$, if $\|P_{k+1} - P_k\| \leq \varepsilon$ for $k \geq 0$, go to Step 5; else continue Step 2 and Step 3. Where $\varepsilon$ is a predefined small threshold.

**Step 5:** Use $u = -K_{k+1} x$ as the approximated optimal control input of the Problem 1.

This algorithm is implemented offline and requires the precise knowledge of the system dynamics. However, in practice, it is often difficult to construct an accurate model or obtain the precise knowledge of the system dynamics. In the next, inspired by reference [11], we will develop a new online ADP algorithm to solve Problem 1 without requiring the prior knowledge of system dynamics.
4 | ADAPTIVE OPTIMAL CONTROL DESIGN WITHOUT SYSTEM DYNAMICS

In this section, based on the proposed new policy iteration algorithm above, we will present an online ADP algorithm that does not require $A$ and $B$.

Assume that $K_0$ is a known stabilizing feedback gain matrix. To solve Equation (24), we rewrite system (1) in the following form [11]

$$\dot{x} = A_k x + B (K_k x + u)\quad (26)$$

where $A_k = A - BK_k$.

Then, along the solutions of (26) by (7) and (8), one can obtain

$$x(t+\Delta t)^T P_k x(t+\Delta t) - x(t)^T P_k x(t) = \int_t^{t+\Delta t} [x^T(A_k^T P_k + P_k A_k) x + 2(u + K_k x)^T B P_k x] d\tau$$

$$\leq -\int_t^{t+\Delta t} x^T Q_k x d\tau + 2\int_t^{t+\Delta t} (u + K_k x)^T R K_{k+1} x d\tau - 2\int_t^{t+\Delta t} (u + K_k x)^T N T x d\tau\quad (27)$$

where $Q_k = Q + K_k^T R K_k - K_k^T N T - N K_k$.

**Remark 3.** It is noteworthy that in Equation (27), one can replace the system matrices with the states and inputs data measured online. Thus, we can obtain $P_k$ and $K_{k+1}$ from Equation (27) without requiring the precise knowledge of $A$ and $B$.

Consequently, under a given stabilizing feedback gain matrix $K_k$, to find $(P_k, K_{k+1})$ with the unknown system matrices, we introduce the Kronecker product

$$x^T Q_k x = (x^T \otimes x^T) vec(Q_k)(K_k x + u)^T R K_{k+1} x$$

$$= \left[(x^T \otimes x^T)(I_n \otimes (K_k)^T R) + (x^T \otimes u^T)(I_n \otimes R)\right] vec(K_{k+1})$$

$$(u + K_k x)^T N T x = [x^T \otimes u^T + (x^T \otimes x^T)(I_n^T \otimes K_k^T)] vec(N T)$$

Furthermore, define the following matrices, $\xi_{xx} \in \mathbb{R}^{l x n^2}$, $\gamma_{xx} \in \mathbb{R}^{l x n^2}$, $\gamma_{xu} \in \mathbb{R}^{l x mn}$, $I$ is a positive integer

$$\xi_{xx} = \left[x \otimes x \left| I_{l_1}^T \right. \quad \ldots \quad x \otimes x \left| I_{l_t}^T \right.\right]^T$$

$$\gamma_{xx} = \left[\int_{l_i}^{t+\Delta t} x \otimes x d\tau, \ldots, \int_{l_i}^{t+\Delta t} x \otimes x d\tau\right]^T$$

$$\gamma_{xu} = \left[\int_{l_i}^{t+\Delta t} x \otimes u d\tau, \ldots, \int_{l_i}^{t+\Delta t} x \otimes u d\tau\right]^T$$

where $0 \leq l_1 < l_2 < \ldots < l_t$.

Then, $\Phi_k \in \mathbb{R}^{l x (n^2 + mn)}$ and $\Psi_k \in \mathbb{R}^l$ are defined as follows

$$\Phi_k = [\xi_{xx}, -2\gamma_{xx} (I_n \otimes (K_k)^T R) - 2\gamma_{xu} (I_n \otimes R)]$$

$$\Psi_k = -\gamma_{xx} vec(Q_k) - 2[\gamma_{xx} + \gamma_{xu} (I_n \otimes K_k^T)] vec(N T)$$

Therefore, Equation (27) can be rewritten in the form of a linear equation as follows

$$\Phi_k \begin{bmatrix} vec(P_k) \\ vec(K_{k+1}) \end{bmatrix} = \Psi_k \quad (28)$$

**Lemma 1.** $\Phi_t$ is a full column rank matrix, satisfying the following condition [16]

$$\exists l > 0, \text{rank}([\xi_{xx}, \gamma_{xx}]) = \frac{n(n+1)}{2} + mn \quad (29)$$

**Lemma 2 ([7]).** If $K_0$ is a initial stabilizing feedback control gain and Lemma 1 is satisfied, the sequences $(P_k)_{k=0}^\infty$ and $(K_k)_{k=0}^\infty$ obtained by solving (28) will respectively converge to optimal $P^*$ and $K^*$.

Now, we can give the online ADP algorithm for solving the optimal control Problem 1 with unknown system dynamics.

**Algorithm 2.** (Online model-free ADP algorithm)

**Step 1:** Initialization and online data collection: Given a stabilizing feedback gain matrix $K_0$ and applying $u_0 = -K_0 x + e$ as the control input $t = l_1 = 0$, where $e$ is the exploration noise. Compute $\xi_{xx}$, $\gamma_{xx}$ and $\gamma_{xu}$ until Lemma 1 is satisfied, let $k = 0$.

**Step 2:** Policy evaluation and improvement: Solve $P_k$ and $K_{k+1}$ from Equation (28).

**Step 3:** Let $k \leftarrow k + 1$, if $\|P_{k+1} - P_k\| \leq \varepsilon$ for $k \geq 0$, go to Step 4; else go to Step 2. Where $\varepsilon$ is a predefined small threshold.

**Step 4:** Use $u = -K_{k+1} x$ as the approximated optimal control input of the Problem 1.

**Remark 4.** If $A$ is Hurwitz, $K_0$ can be set as $K_0 = 0$.

5 | SIMULATION RESULTS

In this section, two simulation examples are provided to demonstrate the effectiveness of the developed online ADP algorithm.

**Example 1.** We consider a practical electromechanical vehicle suspension system described by the following two degrees of freedom of movement differential equations [27,31]

$$\begin{align*}
    m_i \ddot{x}_i &= -k_s (x_i - x_i) - c_s (\dot{x}_i - \dot{x}_i) + u \\
    m_i \ddot{x}_i &= k_s (x_i - x_i) + c_s (\dot{x}_i - \dot{x}_i) - k_l (x_i - x_0) - u
\end{align*}$$
where $m_s$ and $m_t$ are the mass of the 1/4 body and the 1/2 bogie, respectively. $k_s$ and $k_t$ are the lateral stiffness of the spring. $c_s$ is damping coefficients. $u$ is the control force. $x_0$, $x_t$ and $x_s$ represent the displacement.

Define $x_1 = x_s - x_t, x_2 = x_s, x_3 = x_t - x_0, x_4 = x_t, y_1 = x_2, y_2 = x_4, w = x_0$. Based on modern control theory, one can obtain

$$
\dot{x} = Ax + Bu + Gw
$$

where $x = [x_1, x_2, x_3, x_4]^T$ denotes the state variables, $y = [y_1, y_2]^T$ denotes the output variables,

$$
A = \begin{bmatrix}
0 & 1 & 0 & -1 \\
-k_s/m_s & -c_s/m_s & 0 & c_s/m_s \\
0 & 0 & 0 & 1 \\
k_s/m_t & c_s/m_t & -k_t/m_t & -c_t/m_t
\end{bmatrix}
$$

$$
B = \begin{bmatrix}
0 & 1/m_s & 0 & -1/m_t
\end{bmatrix}^T
$$

$$
C = \begin{bmatrix}
-k_s/m_s & -c_s/m_s & 0 & c_s/m_s \\
k_s/m_t & c_s/m_t & -k_t/m_t & -c_t/m_t
\end{bmatrix}
$$

$$
D = [1/m_s - 1/m_t]^T, G = [00 - 10]^T
$$

In practical applications, $Gw$ has no effect on control force, so $Gw$ can be ignored [31]. In order to improve the smooth operation of the vehicle, while improving the dynamic characteristics of the bogie, we should reduce the lateral acceleration of the vehicle body $\ddot{x}_s$ and the lateral acceleration of the frame $\ddot{x}_t$ as possible. Thus, the performance index function is defined as follows

$$
J = \int_0^{\infty} (q_1 \cdot \ddot{x}_s^2 + q_2 \cdot \ddot{x}_t^2 + r \cdot u^2) \, dt
$$

$$
= \int_0^{\infty} (y^T q y + u^T ru) \, dt
$$

where $q = diag([q_1, q_3]). q_1, q_3$ and $r$ respectively denote weighting coefficients of vehicle body acceleration, bogie acceleration and control force.

Substituting $y = Cx + Du$ into the performance index function, one can obtain

$$
J = \int_0^{\infty} x^T Q x + 2x^T Nu + u^T Rudt
$$

where $Q = C^T q C, N = C^T D R = D^T q D + r$.

In the simulation, $m_s = 240kg, m_t = 36kg, k_s = 16000Ns/m, c_s = 1000Ns/m, k_t = 160000N/m, x_0 = [0 0 0.10]^T, q = diag([10^5 10^4]), r = 1$. The exploration noise is chosen as $e = 100 \sum_{i=1}^{100} \sin(w_i t), i = 1, 2, ..., 100, w_i$ are randomly selected from $[-500, 500]$. The system control input is the exploration noise during $t = 0$ to $t = 1s$. 

![FIGURE 1](image1.png) Convergence of $P_k$ to its optimal value $P^*$ [Color figure can be viewed at wileyonlinelibrary.com]

![FIGURE 2](image2.png) Convergence of $K_k$ to its optimal value $K^*$ [Color figure can be viewed at wileyonlinelibrary.com]

![FIGURE 3](image3.png) The system states trajectories [Color figure can be viewed at wileyonlinelibrary.com]

![FIGURE 4](image4.png) The trajectories of $||x||$ [Color figure can be viewed at wileyonlinelibrary.com]
When \( t \geq 1s \), \( u = -K^*x \) will be used as the system optimal control input.

Using the proposed Algorithm 2, \( P_k \) and \( K_k \) are respectively converge to \( P^* \) and \( K^* \) after six iterations. Figure 1 shows the convergence of \( P_k \). Figure 2 shows the convergence of \( K_k \).

The optimal value of \( P^* \) and \( K^* \) are given as follows

\[
P^* = P_6 = \begin{bmatrix}
0.5079 & 0.0330 & 0.0855 & 0.0036 \\
0.0330 & 0.0093 & -0.0309 & 0.0007 \\
0.0855 & -0.0309 & 5.8804 & 0.0309 \\
0.0036 & 0.0007 & 0.0309 & 0.0007
\end{bmatrix}
\]

\[
K^* = K_6 = 10^5 \begin{bmatrix}
-0.1483 & -0.0108 & 1.2756 & 0.0106
\end{bmatrix}
\]

Thus, the optimal control input of the system is

\[
u^* = -K^*x = 14830x_1 + 1080x_2 - 127560x_3 - 1060x_4.
\]

The system states trajectories are shown in Figure 3. The trajectories of \( \|x\| \) are shown in Figure 4. As can be seen from Figures 3 and 4, the system can be asymptotically stabilized to the origin.

The control input trajectory is shown in Figure 5. As can be seen from Figure 5, the control input is divided into two parts. From \( t = 0s \) to \( t = 1s \), the exploration noise is used as the control input. When \( t = 1s \), the control input updates for the optimal control input \( u^* = -K^*x = 14830x_1 + 1080x_2 - 127560x_3 - 1060x_4 \).

To further verify the validity of our algorithm, we apply the traditional PID control algorithm to Example 5. Let \( e(t) = -x_4(t), K_p = 1, K_i = 0, K_d = 0.1 \). The system states trajectories are shown in Figure 6 The control input trajectory is shown in Figure 7.

**Example 2.** We consider the following continuous time linear system:

\[
\dot{x} = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
-12 & -10 & -5
\end{bmatrix} x + \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} u
\]

where \( Q = \begin{bmatrix}
198 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix} \), \( R = \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} \). In the simulation, \( x_0 = [102 \ 5]^T \), the selected exploration noise is the same as Example 5. Along the state trajectories from \( t = 0s \) to \( t = 2s \), the states and inputs information are collected. Then, using the proposed Algorithm 2, \( P_k \) and \( K_k \) are respectively converge to \( P^* \) and \( K^* \) after four iterations. Figure 8 shows the convergence of \( P_k \). Figure 9 shows the convergence of \( K_k \). The optimal value of \( P^* \) and \( K^* \) are given as follows

\[
P^* = P_4 = \begin{bmatrix}
144.0662 & 51.4459 & 6.4932 \\
51.4459 & 28.8222 & 4.2791 \\
6.4932 & 4.2791 & 0.6752
\end{bmatrix}
\]
\[ K^* = K_d = [6.4932 \ 4.2791 \ 1.6752] \]

The system states trajectories are shown in Figure 10. The trajectories of \( ||x|| \) are shown in Figure 11. As can be seen from Figures 10 and 11, the system can be asymptotically stabilized to the origin.

The control input trajectory is shown in Figure 12. It can be seen from Figure 12 that the control input is divided into two parts. From \( t = 0 \) to \( t = 2 \), the exploration noise is used as the control input. When \( t = 2 \), the control input updates for the optimal control input \( u^* = -K^*x = -6.4932x_1 - 4.2791x_2 - 1.6752x_3 \). The control input gradually tends to 0 over time, which accords with engineering.

Similarly, to further verify the validity of our algorithm, we apply the traditional PID control algorithm to Example 5. In this example, let \( e(t) = -x_4(t) \). We conduct simulation experiments for two different sets of PID parameters, such as (1): \( K_p = 10, K_i = 0, K_d = 1 \); (2): \( K_p = 1, K_i = 0, K_d = 1 \).
0. \( K_p = 1 \). The system states trajectories corresponding to different sets of PID parameters are plotted in Figures 13 and 14.

Simulation experiments show that both the traditional PID control algorithm and our proposed algorithm can control the controlled objective well. But our proposed algorithm is in an optimal manner. And the performance of the PID algorithm is greatly affected by the PID parameters. In summary, all the above two simulation results verify the effectiveness of our proposed online model-free ADP algorithm.

6 | CONCLUSION

In this paper, we propose an iterative technique to solve an extended algebraic Riccati equation. According to this proposed technique, an online ADP algorithm to solve extended infinite horizon optimal control problem of continuous time linear systems is presented. The presented ADP algorithm does not require the prior knowledge of system dynamics \( A \) and \( B \). Optimal control design for practical mechatronic vehicle suspension systems and a general continuous-time linear system are given to verify the effectiveness of the online ADP algorithm. In future work, the proposed algorithm can be developed to solve optimal control for uncertain nonlinear systems and optimal tracking control problems.

ACKNOWLEDGEMENTS

National Natural Science Foundation of China 61673065; International Graduate Exchange Program of Beijing Institute of Technology.

ORCID

Jingang Zhao https://orcid.org/0000-0002-2583-8446

REFERENCES

18. K. Vamvoudakis and L. Frank, Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica 46 (2010), no. 5, 878–888.
22. Y. Liu, H. Zhang, Y. Luo, and J. Han, ADP based optimal tracking control for a class of linear discrete-time system with multiple delays, J. Franklin Inst. 353 (2016), no. 9, 2117–2136.

How to cite this article: Gan M, Zhao J, Zhang C. Extended adaptive optimal control of linear systems with unknown dynamics using adaptive dynamic programming. Asian J Control. 2019;1–10. https://doi.org/10.1002/asjc.2243