Finite-time Extended State Observer Based Adaptive Optimal Control of Continuous-time Linear Systems

Qinghua Zhang\textsuperscript{a}, Yong Lin\textsuperscript{b} and Yangdong Wen\textsuperscript{c}

School of Electrical Engineering and Automation, Hefei University of Technology, Hefei, Anhui, 230009, China
\textsuperscript{a}zhangqinhua0807@163.com, \textsuperscript{b}linyong@hfut.edu, \textsuperscript{c}wenyd@hfut.edu

Abstract. In this paper, a novel strategy iterative method based on finite-time extended state observer is proposed to find the online adaptive optimal controller for a class of linear continuous-time systems with completely unknown system dynamics and unmeasurable state variables. In this method, finite-time state observer estimates the system state variables by using the input and output information of the system and the policy iterative method uses the estimated state variables and system input information to iteratively solve the algebraic Riccati equation until the adaptive optimal controller is obtained. It is worth noting that the method proposed in this paper only uses the input and output information of the system, without requiring the apriori knowledge of the system matrices and the state variables. A numerical simulation example will be provided to demonstrate the effectiveness of the method.

1. Introduction
Dynamic programming (DP) \cite{1} as an optimal control problem for nonlinear dynamic systems, is widely used in space technology, industrial control and economic management. The key of DP is Bellman principle of optimality, which transforms the multilevel decision-making problem into a series of single decision-making problems. If the system is linear and the cost function has the quadratic form with respect to the state and control input, the optimal control can be expressed as a linear feedback of the states, where the feedback gain matrix is obtained by solving a standard Riccati equation. However, it is very difficult to solve Riccati equation. In addition, DP method has obvious weaknesses: the amount of computation increases exponentially with the dimension of $x$ and $u$ increasing, (i.e., curses of dimensionality \cite{2}).

In order to overcome these weaknesses, the framework of adaptive dynamic programming (ADP) is proposed, in which the idea is to estimate the cost function by using an approximate structure of function (such as polynomial, neural network, etc.) and to solve DP problem forward-in-time. Due to the great potential in solving the optimal control problem of complex nonlinear systems, ADP scheme has received a widespread attention in recent years, such as adaptive dynamic programming for security of networked control systems with actuator saturation \cite{3}, model-free adaptive control for unknown nonlinear zero-Sum differential game \cite{4}, leader-to-formation stability of multi-agent systems \cite{5}, dual iterative adaptive dynamic programming for a class of discrete-time nonlinear systems with time-delays \cite{6} and so on.
However, partial state variables of the system are often unmeasurable in industrial control. In order to solve the problem, [8-9] propose the output feedback adaptive optimal control method. But this control strategy can only get the suboptimal solution of the controller. [10] Uses neural network (NN) observer to estimate the state of the system. But NN has some shortcomings such as heavy calculation burden and the difficult selection of initial value.

Thus, online adaptive optimal control for a class of linear continuous-time systems with completely unknown system dynamics and unmeasurable state variables is still an open problem. This paper presents a novel strategy iterative method based on finite-time extended state observer to find the online adaptive optimal controller for a class of linear continuous-time systems. This method employs the strategy iterative technique to iteratively solve the algebraic Riccati equation. And the strategy iterative technique only uses system input information and the estimated state variables by finite-time state observer, without requiring the a priori knowledge of the system matrices and state variables.

2. Problem formulation

Consider a class of continuous-time linear systems described by:

\[
\begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx
\end{align*}
\]

The purpose of optimal control is to find a linear quadratic regulator in the form of

\[
u = -Kx
\]

which minimizes the following performance index

\[
J(x, u) = \int_{0}^{\infty} (x'Qx + u'Ru)dt
\]

Where, \(x_0\) is the initial state of the system; \(R = R^T > 0\); \(Q = Q^T > 0\). with \((A, Q^{1/2})\) observable.

Assuming that matrix A, B are known and \((A, Q^{1/2})\) is observable. The linear optimal controller can be found by solving the following Riccati equation.

\[
A^TP + PA + Q - PBR^{-1}B^TP = 0
\]

Under the above assumption, a unique symmetric positive definite matrix \(P^*\) can be obtained by solving (4). We bring \(P^*\) into the following formula.

\[
K^* = R^{-1}B^TP^*
\]

Then, \(K^*\) is the optimal feedback gain vector of (2).

It can be seen from (4) that it is necessary to know the information of the system matrix A and B, when using DP theory to solve the linear optimal control problem. And since (4) is nonlinear in \(P\), it is usually difficult to directly solve \(P^*\) from (4). In order to simplify the computation, some numerical approximation algorithms have been proposed. Among all the different approximation algorithms, the Kleinman algorithm [7] is widely applied. And it is reviewed as follows.

Theorem 1 [7] Let \(K_0 \in \mathbb{R}^{1 \times n}\) be any stabilizing feedback gain matrix, and iteratively calculate the following steps for \(k = 0, 1, ...\)

1. Solve for the real symmetric positive definite solution \(P_k\) of the Lyapunov equation
(A - BK_k) P_k + P_k (A - BK_k) + Q + K_k^T R K_k = 0 \quad (6)

(2) Update the feedback gain matrix by

\[ K_{k+1} = R^T B^T P_k \] \quad (7)

Then, the following properties hold:

\[ A - BK_k \text{ is Hurwize, } P^* \leq P_{k+1} \leq P_k, \lim_{k \to \infty} K_k = K^*, \lim_{k \to \infty} P_k = P^*. \]

In theorem 1, the approximate optimal state feedback matrix is obtained by iteratively solving (6) and (7). It is worth noting that it is easy to directly solve \( P_k \) from (6), since (6) is linear in \( P_k \).

In order to solve \( P_k \) without the information of matrix A, a strategy iteration \([11]\) method is proposed by using (8) instead of (6).

\[
\begin{align*}
x^T(t) P_k (x(t) - x^T(t + \delta t) P_k x(t + \delta t)) \\
= \int_0^t x^T(\tau) Q x(\tau) + u_k^T R u_k d\tau
\end{align*}
\]

\[ K_k = R^T B^T P_{k+1} \] \quad (9)

where \( u_k = -K_k x \) is the system control input on the time interval \([t, t + \delta t]\). The algorithm solves \( P_k \) from (8) without relying on matrix A. However, it can be seen from (9) that the algorithm still needs the knowledge of matrix B. For the purpose of solving \( P_k \) without the knowledge of A or B, a new online iterative algorithm \([13]\) is developed. We rewrite the original system (2) as:

\[ \dot{x} = A_k x + B(K_k x + u) \] \quad (10)

where \( A_k = A - BK_k \). Then, an online strategy iteration formula is given for iteratively solving \( P_k \) and \( K_{k+1} \).

\[
\begin{align*}
x^T(t + \delta t) P_k x(t + \delta t) - x^T(t) P_k x(t) \\
= \int_0^t [x^T(A_k^T P_k x) + x^T(B K_k x + u)^T B^T P_k x] d\tau
\end{align*}
\]

\[
= -\int_0^t x^T Q_k x d\tau + 2\int_0^t x^T (K_k x + u)^T R K_{k+1} x d\tau
\]

where \( Q_k = Q + K_k^T R K_k \). Notice that in (11), based on (6), the term \(-x^T Q_k x\), which depends on the unknown matrices \( A \) and \( B \), replaces \( x^T (A_k^T P_k x + P_k A_k x) \). Also, based on (7), the term \( B^T P_k \) involving \( B \) is replaced by \( R K_{k+1} \). It can be seen from the above-mentioned analytical result that the algorithm does not depend on the system model, but still needs all state variables of the system.

3. Finite-time extended state observer based on line optimal control

For a class of linear continuous-time systems with completely unknown system dynamics and unmeasurable state variables, we propose a new online learning method. The structure of the algorithm is shown in Fig. 1, where \( t_0 \) and \( K_0 \) are the parameters to be selected.
3.1. State Observation

Let $K_0 \in \mathbb{R}^{1 \times n}$ be any stabilizing feedback gain matrix, and select a system input $u_1 = -K_0 \hat{x}, u = u_1$. Then, the state variables of the system are estimated by the following finite-time extended state observer [11-12].

$$
\begin{align*}
\dot{z}_1 &= v_1, v_1 = -\lambda_1 \eta^{T+1} \left| z_1 - x_1 \right|^{\frac{1}{T+1}} \text{sign}(z_1 - x_1) + z_2; \\
\dot{z}_2 &= v_2, v_2 = -\lambda_2 \eta^{T+2} \left| z_2 - v_1 \right|^{\frac{1}{T+2}} \text{sign}(z_2 - v_1) + z_3; \\
&\vdots \\
\dot{z}_i &= v_i, v_i = -\lambda_i \eta^{T+i-1} \left| z_i - v_{i-1} \right|^{\frac{1}{T+i-1}} \text{sign}(z_i - v_{i-1}) + z_{i+1}; \\
&\vdots \\
\dot{z}_n &= v_n + b_n u, v_n = -\lambda_n \eta^{T+n} \left| z_n - v_{n-1} \right|^{\frac{1}{T+n}} \text{sign}(z_n - v_{n-1}) + z_{n+1}; \\
\dot{z}_{n+1} &= v_{n+1}, v_{n+1} = -\lambda_{n+1} \eta \text{sign}(z_{n+1} - v_n).
\end{align*}
$$

(12)

where, $\lambda_i, i = 1, 2, ..., n+1$ are the coefficients of observer; $z_i, i = 1, 2, ..., n$ are the real-time robust estimations of $x_i, i = 1, 2, ..., n$ (i.e. $\hat{x}_i$ is the robust estimation value of $d(i.e. \hat{d} = z_{n+1}$). The finite-time state observer shown in (11) has the following features:

1. There exists a time constant $T$ such that $\hat{x}_i(t) = x_i(t), (i = 1, 2, ..., n + 1)$ for $t \geq T$;
2. The time constant $T$ can be estimated. The specific proof of finite time convergence for finite-time extended state observer is given in section 4.

3.2. Switching Process

Based on the features of the finite-time state observer, we manually select a time constant $t_0 \geq T$ and make the following switching strategy.

**Figure 1.** Finite-time extended state observer based online optimal control.
\begin{equation}
\begin{aligned}
\text{if } t \leq t_0 \\
u(t) = u_1(t), \\
\text{else} \\
u(t) = u_2(t).
\end{aligned}
\end{equation}

where \( u_1 \) and \( u_2 \) are shown in Fig. 1.

3.3. Online Optimal Control

We first consider the following control strategy:

\begin{equation}
u_2 = -K_k \hat{x} + \sigma
\end{equation}

where \( \sigma \) is a time-varying artificial exploration noise for online learning. Then a policy iteration formula is given.

\begin{equation}
\hat{x}^T(t + \delta t) P_k \hat{x}(t + \delta t) - \hat{x}^T(t) P_k \hat{x}(t) \\
= - \int_{t}^{t + \delta t} \hat{x}(\tau) Q_k \hat{x}(\tau) d\tau \\
+ 2 \int_{t}^{t + \delta t} (\sigma^T R K_{k+1} \hat{x}(\tau)) d\tau
\end{equation}

where \( Q_k = Q + K_k^T R K_k \). We call (15) the \textit{online policy iteration} equation, for it relies on the knowledge of estimated value of state and the control policy being applied, instead of the system knowledge and system state information. In order to iteratively solve \( P_k \) and \( K_{k+1} \) by using (15), the specific steps are as follows.

First, in order to simplify the calculation without losing any information, we use the Kronecker product to convert \( \hat{x}^T P_k \hat{x} \) and \( \sigma^T R K_{k+1} \hat{x} \) into the following form [28].

\begin{equation}
\begin{aligned}
\mathbb{S}^T \mathbb{P}_{\mathcal{K}} \mathbb{S} = (\mathbb{S}^T \otimes \mathbb{S}) \text{vec}(P_k), \\
\sigma^T R K_{k+1} \mathbb{S} = (\mathbb{S}^T \otimes (\sigma^T R)) \text{vec}(K_{k+1}).
\end{aligned}
\end{equation}

Then, by specifying \( t = t_{k,1}, t_{k,2}, \ldots, t_{k,l_k} \) with \( 0 \leq t_{k,i} + \delta t \leq t_{k+1,1} \) and \( t_{k,i} + \delta t \leq t_{k+1,1} \) for all \( k = 0,1, \ldots \) and \( i = 1,2, \ldots, l_k \), (16) can be used to generate a series of equations as follows.

\begin{equation}
\Theta_k \begin{bmatrix}
\text{vec}(P_k) \\
\text{vec}(K_{k+1})
\end{bmatrix} = \Xi_k
\end{equation}

If \( \Theta_k \) has full column rank, \( \text{vec}(P_k) \) and \( \text{vec}(K_{k+1}) \) can be directly solved by the following equation.

\begin{equation}
\begin{bmatrix}
\text{vec}(P_k) \\
\text{vec}(K_{k+1})
\end{bmatrix} = (\Theta_k^T \Theta_k)^{-1} \Theta_k^T \Xi_k
\end{equation}
Assumption 2: For each \( k = 0, 1, ..., \) there exists a sufficiently large integer \( l_k > 0 \) to make the following rank condition hold.

\[
\text{rank} \left( \Theta_k \right) = \frac{n(n+1)}{2} + n
\]  

(19)

Lemma 1: Under Assumption 2, there is a unique pair \( (P_k, K_{k+1}) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1} \) satisfying (19) with \( P_k = P_k^T \).

Finally, the strategy iteration algorithm shown above is summarized in Fig. 2.

In Figure 2, \( \varepsilon \) is the stopping threshold, which determines the accuracy of iterative learning.

4. Numerical simulation

Consider the following linear constant system [27]:

\[
\begin{align*}
\mathbf{x} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_1 & -a_2 & -a_3 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ a_1 \end{bmatrix} u \\
\mathbf{y} &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}
\end{align*}
\]  

(20)

with initial state vector \( \mathbf{x} = [1, -1, 15]^T \). Where \( a_1, a_2, a_3, b \) is the uncertain parameters of linear system. Only for simulation purpose, we set \( a_1 = 0.1, a_2 = 0.5, a_3 = 0.7, b = 1 \). For the sake of simplicity of the experiment, it is assumed that (20) is open-loop stable. The design objective is to find
a linear optimal control law to minimize the following cost function $J(x_0, u) = \int_0^\infty (|x|^2 + u^2) \, dt$. The controller parameters designed in this paper are set as follows: 1) initial state feedback gain matrix: $K_0 = [0, 0, 0]$; 2) parameters of the finite time extended state observer: $\lambda_1 = 30$, $\lambda_2 = 20$, $\lambda_3 = 15$, $\lambda_4 = 0.01$, $b_0 = 0.8$, $\eta = 5$; 3) switching time: $t_0 = 2$; 4) policy iteration algorithm parameters: $\delta t = 0.11$, $t_{k+1, k} - t_{k, k} = t_{k, k+1} - t_{k, k} = 0.11$, $c = 0.01$, $\sigma = 2\sin(3\pi t)$. For comparison purposes, the optimal cost matrix $P^*$ and the optimal gain matrix $K^*$, which are directly calculated by the Riccati equation, are given below.

$$P^* = \begin{bmatrix} 2.36 & 2.24 & 0.91 \\ 2.24 & 4.24 & 1.90 \\ 0.91 & 1.90 & 1.60 \end{bmatrix} \quad (21)$$

$$K^* = [0.91 \ 1.90 \ 1.60]$$

Then, the simulation results are as follows.
1) The states trajectories are plotted in Figure 3.

![Figure 3. Profile of the system states trajectories.](image)

2) After seven iterations, the approximate optimal matrices $P$ and $K$ are obtained as follows.

$$P_7 = \begin{bmatrix} 2.37 & 2.25 & 0.91 \\ 2.25 & 4.26 & 1.90 \\ 0.91 & 1.90 & 1.60 \end{bmatrix} \quad (22)$$

$$K_7 = [0.91 \ 1.90 \ 1.60]$$

3) The convergence of $K_k$ to its optimal values is illustrated in Fig. 4.
As shown in Fig. 3, the state observer can quickly track and estimate the state variables of the system, even though the state variables change rapidly. And in switching process, the state variables of the system are continuous and smooth because of the continuity of control strategy. Finally, the closed-loop system remains stable and the system state variables tend to zero. Moreover, as can be seen from Fig. 4, after a period of online learning, $K_k$ tends to its optimal value $K^*$. And specific numerical analysis can refer to (21) and (22).

5. Conclusion

In this paper, a computational policy iteration approach based on finite-time extended state observer has been proposed for finding online optimal controllers for a class of continuous-time linear systems with completely unknown system dynamics and unmeasurable state variables. In this method, finite-time extended state observer is used to estimate the state variables, and the policy iteration is employed to solve the algebraic Riccati equation iteratively with the knowledge of estimated value of state. Numerical simulation results demonstrated that the proposed method can still find the online optimal controller, although the system parameters are uncertain and the system state variables are not measurable.

References

[1] Bellman R E. Dynamic Programming [M]. Dover Publications, Incorporated, 2003.
[2] Bertsekas D P. Dynamic Programming & Optimal Control [J]. 2005, 47.
[3] Yang H, Li Y, Liu Z, et al. Adaptive dynamic programming for security of networked control systems with actuator saturation [J]. Information Sciences, 2018, 460.
[4] Zhong X, He H, Ding W, et al. Model-Free Adaptive Control for Unknown Nonlinear Zero-Sum Differential Game [J]. IEEE Transactions on Cybernetics, 2017, 48 (5): 1633-1646.
[5] Gao W, Jiang Z P, Lewis F L, et al. Leader-to-Formation Stability of Multi-Agent Systems: An Adaptive Optimal Control Approach [J]. IEEE Transactions on Automatic Control, 2018, PP (99): 1-1.
[6] Wei Q, Wang D, Zhang D. Dual iterative adaptive dynamic programming for a class of discrete-time nonlinear systems with time-delays [J]. Neural Computing & Applications, 2013, 23 (7-8): 1851-1863.
[7] Kleinman D. On an iterative technique for Riccati equation computations [J]. IEEE Trans.a.c, 1968, 13 (1): 114-115.
[8] Gao W, Jiang Y, Jiang Z P, et al. Output-feedback adaptive optimal control of interconnected
systems based on robust adaptive dynamic programming [J]. Automatica, 2016, 72 (8): 37-45.

[9] Gao W, Huang M, Jiang Z P, et al. Sampled-data-based adaptive optimal output-feedback control of a 2-degree-of-freedom helicopter [J]. IET Control Theory & Applications, 2016, 10 (12): 1440-1447.

[10] Liu D, Huang Y, Ding W, et al. Neural-network-observer-based optimal control for unknown nonlinear systems using adaptive dynamic programming [J]. International Journal of Control, 2013, 86 (9): 1554-1566.

[11] ArieLevant. Higher-order sliding modes, differentiation and output-feedback control [J]. International Journal of Control, 2003, 76 (9-10): 924-941.

[12] Tian Y P. Finite-time stability of cascaded time-varying systems [J]. International Journal of Control, 2007, 80 (4): 646-657.