Adaptive Suboptimal Output-Feedback Control for Linear Systems Using Integral Reinforcement Learning

Lemei M. Zhu, Hamidreza Modares, Gan Oon Peen, Frank L. Lewis, *Fellow, IEEE*, and Baozeng Yue

***Abstract*— Reinforcement learning (RL) techniques have been successfully used to find optimal state-feedback controllers for continuous-time (CT) systems. However, in most real-world control applications, it is not practical to measure the system states and it is desirable to design output-feedback controllers. This paper develops an online learning algorithm based on the integral RL (IRL) technique to find a suboptimal output-feedback controller for partially unknown CT linear systems. The proposed IRL-based algorithm solves an IRL Bellman equation in each iteration online in real time to evaluate an output-feedback policy and updates the output-feedback gain using the information given by the evaluated policy. The knowledge of the system drift dynamics is not required by the proposed method. An adaptive observer is used to provide the knowledge of the full states for the IRL Bellman equation during learning. However, the observer is not needed after the learning process is finished. The convergence of the proposed algorithm to a suboptimal output-feedback solution and the performance of the proposed method are verified through simulation on two real-world applications, namely, the X–Y table and the F-16 aircraft.**

***Index Terms*— Integral reinforcement learning (IRL), linear continuous-time (CT) systems, optimal control, output feedback.**

I. Introduction

In THIS paper, a new partially model-free algorithm based on policy iteration (PI) is presented for finding a suboptimal output-feedback controller for continuous-time (CT) linear time-invariant systems. Optimal controllers for linear systems are usually designed offline by solving the algebraic Riccati equation (ARE), which requires complete knowledge of the system dynamics [1]. For real-world control applications, however, designing online controllers in real time without requiring complete knowledge of the system dynamics is often desirable. Although classical adaptive control design methods [2] are capable of handling modeling uncertainties, they are generally far from optimal solutions.

Recently, reinforcement learning (RL) [3]–[6] has been used as a method to find optimal state-feedback controllers for both linear [7]–[11] and nonlinear systems [12]–[17]. The optimal solution is obtained by learning online the solution to the Hamilton–Jacobi–Bellman (HJB) equation [1]. A survey of RL-based feedback control design methods is found in [18]–[20]. Most of the available RL algorithms for approximating the HJB solution are based on the PI technique [3]. Instead of trying a direct approach to solve the HJB equation, the PI technique describes a class of iterative algorithms consisting of two steps, namely, policy evaluation and policy improvement. In the policy evaluation step, the performance of a given control policy is evaluated, and in the policy improvement step, an improved policy is obtained using the information given by the evaluated policy. These two steps are repeated until the policy improvement step no longer changes the present policy, and hence the convergence to the optimal controller is achieved. For linear systems, the HJB equation reduces to an ARE. Kleinman [21] presented an offline PI algorithm with guaranteed convergence to the optimal state-feedback solution for linear CT systems. The algorithm requires complete knowledge of the system dynamics to solve the ARE. To obviate the requirement of the drift dynamics of the system, Vrabie *et al.* [11], [22] presented a new formulation of the PI algorithm, called the integral RL (IRL) technique, for linear CT systems. The IRL solves in an online fashion the optimal control problem for CT systems using only partial knowledge about the system dynamics. Later, Lee *et al.* [8] and Jiang and Jiang [9] used the IRL idea to present online model-free RL algorithms for completely unknown CT linear systems. For discrete-time systems, Q-learning [10], [23], [24] (called action-dependent heuristic dynamic programming by Werbos [25], [26]), model-free approximate dynamic programming (ADP) [27], and dual model-free ADP algorithms [28], [29] are presented to solve optimal state-feedback control problems efficiently and without knowing system dynamics.

While RL algorithms are efficiently used to solve the optimal state-feedback problems, there exist no results on developing a learning algorithm based on the RL techniques to

Manuscript received September 19, 2013; revised January 4, 2014 and February 27, 2014; accepted May 5, 2014. Date of publication June 3, 2014; date of current version December 15, 2014. Manuscript received in final form May 7, 2014. This work was supported in part by the National Science Foundation under Grant ECCS-1128050 and Grant IIS-1208623, in part by the Office of Naval Research under Grant N00014-13-1-0562, in part by the European Office of Aerospace Research and Development, Air Force Office of Scientific Research, under Grant 13-3055, in part by the National Natural Science Foundation of China under Grant 61120106011, and in part by the China Education Ministry Project 111 under Grant B08015. Recommended by Associate Editor A. Alessandri.

L. Zhu is with the Department of Basic, North China Institute of Science and Technology, Hebei 101601, China (e-mail: zhulemei@hotmail.com).

H. Modares is with the Arlington Research Institute, University of Texas, Fort Worth, TX 76118 USA (e-mail: modares@uta.edu).

G. O. Peen is with the Singapore Institute of Manufacturing Technology, Singapore 638075 (e-mail: opgan@simtech.a-star.edu.sg).

F. L. Lewis is with the Arlington Research Institute, University of Texas, Fort Worth, TX 76118 USA, and also with the State Key Laboratory of Synthetical Process Automation, Northeastern University, Shenyang 110036, China (e-mail: lewis@uta.edu).

B. Yue is with the Department of Mechanics, School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: bzyue@bit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2014.2322778

find an output-feedback controller for uncertain CT systems. However, it is not easy to measure the full states of the systems in practical situations, and it is desirable to control the system based on the directly observed output of the system. The static output-feedback problem is one of the most researched problems in the control society. The use of static output feedback allows flexibility and simplicity of implementation, and it is of extreme importance in practical controller design applications, including flight control, manufacturing robotics, and elsewhere where it is desired that the controller have certain prespecified desirable structure, for example, feedback only from certain available sensors.

The problem of stabilizing linear CT systems using static output feedback has attracted much attention in the literature. Necessary and sufficient conditions for the existence of a stabilizing static output-feedback controller are provided in [30]. The optimal static output-feedback control design is also considered by some researchers. It is required to solve two AREs simultaneously to find a suboptimal output-feedback control solution [1]. Because of the difficulty of solving these AREs, even for completely known dynamical systems, to our knowledge, no static output-feedback design method is presented using RL techniques. Note that although in [7] an RL-based output-feedback method is designed, however, it is a dynamic output-feedback design method, and it is presented for discrete-time systems, not for CT systems. Lewis and Vamvoudakis [7] expressed the system states in terms of the delayed values of the inputs and outputs to design an optimal dynamic output-feedback controller for discrete-time systems.

In this paper, we introduce the use of the IRL approach to develop an online solution for suboptimal output-feedback control of partially unknown CT systems. Necessary and sufficient conditions for the existence of the solution to the output feedback are presented. An offline PI algorithm is given to find a suboptimal output-feedback solution that satisfies the given necessary and sufficient conditions in convergence. This offline algorithm iterates on a Lyapunov equation to find a suboptimal output-feedback controller. The solution to the Lyapunov equation requires complete knowledge of the system dynamics. To develop an online partially model-free PI algorithm, the IRL idea is used to replace the Lyapunov function with the IRL Bellman equation, which does not involve the system dynamics. Since the IRL needs the knowledge of the system states, an adaptive observer for CT linear systems is designed to measure the full states during learning. After the learning is finished, the system states are no longer needed, and therefore, the observer is not required. Simulation results on an X–Y table system and an F-16 aircraft verify the suitability of the proposed methods.

II. IRL

In this section, the IRL technique [11] is presented. The IRL is an online learning method to find the solution to the optimal state-feedback control problems for CT systems without using knowledge regarding the internal system dynamics. This method will be used in Section IV to find a suboptimal output-feedback solution online in real time.

Consider the linear CT system

$$\dot{x} = A\,x + B\,u \qquad y = C\,x \tag{1}$$

where

$x \in \mathbb{R}^{n\times 1}$	system state vector;
$y \in \mathbb{R}^{p\times 1}$	system output;
$u \in \mathbb{R}^{m\times 1}$	control input;
$A \in \mathbb{R}^{n\times n}$	drift dynamics of the system;
$B \in \mathbb{R}^{n\times m}$	input matrix.

It is assumed that C has full rank, the pair (A, B) is controllable, and the pair (A, C) is observable.

Assume in this section that the full knowledge of the system states are available for measurement and one can control the system through feedback of the states. That is, assume that in the linear system (1), we have $y = x$. Given the state-feedback control policy $u = -K_x x$, the value function can be written as

$$V(x(t)) = \int_t^\infty x^T(\mathcal{T})\left(Q + K_x^T R K_x\right)x(\mathcal{T})d\mathcal{T} = x^T(t)P\,x(t) \tag{2}$$

where $Q = Q^T \geq 0$ and $R = R^T > 0$ are the state and control performance weights, respectively.

The solution to the optimal control problem, determined by Bellman's optimality principle, is given by [1]

$$K_x = R^{-1}B^T P \tag{3}$$

where P is the unique positive definite solution for the ARE

$$A^T P + PA + Q - PBR^{-1}B^T P = 0. \tag{4}$$

To find the optimal state-feedback solution, first, the ARE (4) is solved for P and then the optimal solution is given using the ARE solution P in (3).

To find the solution to the ARE in an online manner using only partial knowledge about the system dynamics, the IRL technique can be used. In the IRL, a Bellman equation is introduced for evaluating a fixed control policy, which does not require any knowledge of the system dynamic. Note that for any time t and time interval $T > 0$, the value function (2) satisfies

$$V(x(t)) = \int_t^{t+T} x^T(\mathcal{T})\left(Q + K_x^T R K_x\right)x(\mathcal{T})d\mathcal{T} + V(x(t+T)). \tag{5}$$

Based on (5) and using (2), and denoting $x(t)$ with x_t, one has

$$x_t^T P x_t = \int_t^{t+T} x_{\mathcal{T}}^T\left(Q + K_x^T R K_x\right)x_{\mathcal{T}}d\mathcal{T} + x_{t+T}^T P x_{t+T}. \tag{6}$$

This equation is known as the IRL Bellman equation and can be used to evaluate a fixed control policy $u = -K_x x$, i.e., to obtain the matrix P in (2) for control gain K_x.

Using the IRL Bellman equation to evaluate a policy and a control update law in the form of (3) to find an improved policy, the following iterative IRL algorithm is

introduced in [11] to find the optimal state-feedback solution online in real time.

Algorithm 1 (Online IRL for Finding the Optimal State-Feedback Control): Start with an initial stabilizing control gain K_x^1 and then until convergence perform the following two steps.

1) (Policy evaluation) given a control input gain K_x^i, find the P^i using the IRL Bellman equation as

$$x_t^T P^i x_t = \int_t^{t+T} x_{\mathcal{T}}^T \left(Q + \left(K_x^i\right)^T R \left(K_x^i\right)\right) x_{\mathcal{T}} d\mathcal{T} + x_{t+T}^T P^i x_{t+T}. \tag{7}$$

2) (Policy improvement) update the control policy using

$$K_x^{i+1} = R^{-1} B^T P^i. \tag{8}$$

This is a PI algorithm that solves the ARE (4) online using data measured along the system trajectories in real time. Note that, it does not involve the plant matrix A.

III. Output Feedback for CT Linear Systems

In this section, the output-feedback control problem for CT systems is discussed. Necessary and sufficient conditions for the existence of the output-feedback solution and the relationship between possible solutions and global optimal output-feedback solution are discussed. An offline PI algorithm is then presented to find a suboptimal output-feedback solution. This algorithm is a basis for presenting our online IRL algorithm in the next section.

In contrast to the previous section, it is assumed here that the full knowledge of the system states is not available for measurement and the aim is to design a static output-feedback controller for the system given by (1). In the static output-feedback control design, the control input is calculated directly from the multiplication of output measurements by constant feedback gains, formed as

$$u = -Ky \tag{9}$$

where K is an $m \times p$ constant output-feedback gain matrix to be determined by the design procedure. The aim of the output feedback is to find the control gain K to stabilize the system dynamics (1).

It is first shown that the Lyapunov equation related to an output-feedback gain K, which can be used to evaluate the value function corresponding to K, is given by

$$A_c^T P + P A_c + Q + C^T K^T R K C = 0 \tag{10}$$

with $A_c = A - BKC$. To show this, given the output-feedback control policy as $u = -Ky = -KCx$ and using (2), the corresponding value function can be written as

$$V(x(t)) = \int_t^{\infty} x^T (Q + C^T K^T R K C)\, x\, d\mathcal{T} = x^T(t) P x(t). \tag{11}$$

By taking the derivative of (11), one has

$$\dot{x}^T P x + x^T P \dot{x} + x^T (Q + C^T K^T R K C) x = 0. \tag{12}$$

Substituting $\dot{x}$ from (1) into (12) gives

$$x^T ((A - BKC)^T P + P(A - BKC)^T + Q + C^T K^T R K C) x = 0. \tag{13}$$

This expression must hold for any x. As a result

$$(A - BKC)^T P + P(A - BKC)^T + Q + C^T K^T R K C = 0 \tag{14}$$

which is identical to (10). Therefore, the Lyapunov equation (10) can be used to evaluate the value function related to the output-feedback gain K.

In the following, necessary and sufficient conditions for the existence of the solution to the output-feedback problem are given. This is inspired by the work of [31] that presented necessary and sufficient conditions for the H_∞ output-feedback problem. Slightly different conditions for the existence of the solution to the output-feedback problem are given in [30].

Theorem 1: The system defined by (1) is output-feedback stabilizable if and only if:

1) (A, B) is stabilizable and (A, C) is detectable;
2) there exist matrices K and L such that

$$KC = R^{-1}(B^T P + L) \tag{15}$$

holds, where $P^T = P > 0$ is the solution of

$$A^T P + PA + Q - PBR^{-1}B^T P + L^T R^{-1} L = 0. \tag{16}$$

Proof (Necessity): Suppose that there exists an output-feedback gain K that stabilizes the system dynamic. Therefore, since $A - BKC$ is stable, then (A, B) is stabilizable because $A - BG$ is stable for $G = KC$ and (A, C) is detectable because $A - LC$ is stable for $L = BK$. Hence, the necessity of the first condition is shown. On the other hand, since $A - BKC$ is stable, there exists a unique symmetric nonnegative definite matrix P such that the following Lyapunov equation holds:

$$(A - BKC)^T P + P(A - BKC) + Q + C^T K^T R K C = 0. \tag{17}$$

Substituting KC from (15) into (17) gives (16). This shows the necessity of the second condition.

Sufficiency: Suppose that conditions 1 and 2 in the statement of Theorem 1 hold. Consider the Lyapunov function $V(x) = x^T P x$. The derivative of the Lyapunov function along the trajectories of the closed-loop system is given by

$$\dot{V}(x) = x^T [(A - BKC)^T P + P(A - BKC)] x. \tag{18}$$

Using (15) in (18), one has

$$\dot{V}(x) = x^T [A^T P + PA - PBR^{-1}B^T P - PBR^{-1}B^T P - L^T R^{-1} B^T P - PBR^{-1} L] x. \tag{19}$$

Using (16) in (19) yields

$$\begin{aligned} \dot{V}(x) &= x^T [-Q - L^T R^{-1} L - PBR^{-1}B^T P - L^T R^{-1} B^T P - PR^{-1} BL] x \\ &= x^T [-Q - (B^T P + L)^T R^{-1} (B^T P + L)] x \\ &= x^T [-Q - (KC)^T R (KC)] x < 0. \end{aligned} \tag{20}$$

Therefore, the closed-loop system is asymptotically stable. This completes the proof.

Note that, substituting K from (15) into the Lyapunov equation (10) gives (16). This shows the equivalence of (10) and (16) for evaluating the value function of a given K in the form of (15).

In the following Theorem 2, the relationship between the output-feedback solution given by (15) and (16), and the global optimal output-feedback solution is given.

Theorem 2: Consider the system (1) with the control input given by (9). If the control gain K satisfies (15) and (16) with $L = 0$, then K is the global optimal output-feedback gain.

Proof: Consider the value function (11). Then, for a stabilizing control policy one has

$$\begin{aligned} V(x(0),u) &= \int_0^\infty [x^T(Q+(KC)^T R\,KC)\,x]\,d\mathcal{T} \\ &\quad + \int_0^\infty \frac{d}{dt}(V(x))\,d\mathcal{T} + V(x(0)) \\ &= \int_0^\infty [x^T(Q+(KC)^T R\,KC)\,x]\,d\mathcal{T} \\ &\quad + \int_0^\infty x^T((A-BKC)^T P \\ &\quad + P(A-BKC))x\,d\mathcal{T} + V(x(0)) \\ &= \int_0^\infty x^T H(K,P)\,x\,d\mathcal{T} + V(x(0)) \end{aligned} \tag{21}$$

where $H(K,P)$ is defined as

$$H(K,P) = Q+(KC)^T RKC+(A-BKC)^T P+P(A-BKC). \tag{22}$$

After doing some manipulations, $H(K,P)$ becomes

$$\begin{aligned} H(K,P) = {} & Q + A^T P + PA - PBR^{-1}B^T P + L^T R^{-1} L \\ & +(-R^{-1}(B^T P + L) + KC)^T R \\ & \times(-R^{-1}(B^T P + L) + KC) \\ & +2(-R^{-1}(B^T P + L) + KC)^T L. \end{aligned} \tag{23}$$

Based on (15), define $K^*C = R^{-1}(B^T P + L)$, where P is the solution to the ARE (16). Then, using (16), the first row of the right-hand side of (23) is 0 and (23) becomes

$$\begin{aligned} H(K,P) = {} & (KC - K^*C)^T R\,(KC - K^*C) \\ & +2\,(KC - K^*C)^T L. \end{aligned} \tag{24}$$

Using (24) in (21) yields

$$\begin{aligned} V(x(0),u) = {} & \int_0^\infty x^T((KC - K^*C)^T R \\ & \times(KC-K^*C) + 2\,(KC-K^*C)^T L\,)x\,d\mathcal{T} \\ & +V(x(0)). \end{aligned} \tag{25}$$

Minimizing V with respect to the gain K yields

$$KC - K^*C = R^{-1}L. \tag{26}$$

Using $K^*C = R^{-1}(B^T P + L)$ in (26) gives the global optimal gain K as

$$KC = R^{-1}B^T P. \tag{27}$$

Substituting this gain in the Lyapunov equation (22) gives

$$A^T P + PA + Q - PBR^{-1}B^T P = 0. \tag{28}$$

Equations (27) and (28), which give the global optimal gain, are equivalent to (15) and (16) with $L = 0$ and this completes the proof. □

The output-feedback problem might not have a global optimal solution, that is, there might be no K to satisfy (27) and (28). Comparing (27) and (28) with (15) and (16), one can conclude that the solution to (15) and (16) gives a suboptimal output-feedback control input gain K. This relationship motivates proposing an offline algorithm by iterating on a Lyapunov equation until a matrix L is found that satisfies the necessary and sufficient conditions given by (15) and (16). The matrix L is completely arbitrary and it can be chosen to get a feasible control gain. Matrix L shows in some sense the difference between the global optimal state-feedback gain and the proposed output-feedback gain.

Note that, it was shown that the Lyapunov equation (10) can be used to evaluate an output-feedback control policy given in the form of (15). Using (10) to evaluate a policy and (15) as a control update law, an offline iterative PI algorithm for finding the solution to the output-feedback control is given as follows.

Algorithm 2 (Offline PI Solution for Output-Feedback Control).

1) Start with an admissible control policy K^0 and $L = 0$.
2) (Policy evaluation) given a control input gain K^i, find the P^i using the equation

$$\begin{aligned} (A - BK^iC)^T P^i + P^i(A - BK^iC) + Q \\ +C^T(K^i)^T R(K^i)C = 0. \end{aligned} \tag{29}$$

3) (Policy improvement) update the control policy and the matrix L using

$$\begin{aligned} K^{i+1} &= R^{-1}(B^T P^i + L^i)C^T(CC^T)^{-1} \\ L^{i+1} &= RK^{i+1}C - B^T P^i. \end{aligned} \tag{30}$$

The above PI algorithm formulates an offline solution for the output-feedback control, which needs the dynamics of the system. If it converges, and then it satisfies the necessary and sufficient conditions given by (15) and (16). The reformulation of finding a suboptimal output feedback to a PI algorithm in Algorithm 2 enables us to develop an IRL algorithm for finding a suboptimal output-feedback policy without knowing knowledge of the system drift dynamics. Next, we propose an online solution for learning the output-feedback control solution for partially unknown dynamics, i.e., the drift dynamic matrix A is unknown.

IV. Online IRL Solution to the Output-Feedback Control Solution

In this section, an online algorithm based on the IRL technique is given to learn a suboptimal output-feedback solution

for partially unknown systems. The IRL is used to develop an online learning algorithm to find an output-feedback control without knowing the drift dynamics, i.e., the matrix A in (1). Since the IRL needs the system states, an observer is used to provide the system states information during learning. After the learning is finished and once the optimal output-feedback gain is found, the system states are not needed and the observer is no longer required.

A. Proposed IRL-Based Output-Feedback Algorithm

The Lyapunov equation (29) requires complete knowledge of the system dynamics to evaluate a control policy. To obviate the requirement of complete knowledge of the system dynamics, the IRL idea can be used [11]. In this technique, an IRL Bellman equation is used, instead of the Lyapunov equation, to evaluate a policy without requiring any knowledge of the system dynamics.

Lemma 1: Assuming that $A_i = A - BK^iC$ is stable, solving for P^i in the following Bellman equation (31) is equivalent to finding the solution of the Lyapunov equation (29):

$$x_t^T P^i x_t = \int_t^{t+T} x_{\mathcal{T}}^T (Q + C^T (K^i)^T R (K^i) C) x_{\mathcal{T}} d\mathcal{T} + x_{t+T}^T P^i x_{t+T}. \tag{31}$$

Proof: See [11] for the proof. □

Using the IRL Bellman equation (31) to evaluate a control policy and update law in the form of (30) to find an improved policy, the following iterative IRL online algorithm is introduced to solve the output-feedback problem.

Algorithm 3 (Online IRL Solution for Output-Feedback Control).

1) Start with an initial stabilizing control gain K^1 and $L = 0$, and then until convergence perform the following two steps.
2) (Policy evaluation) given a control input gain K_i, find the P_i using the equation

$$x_t^T P^i x_t = \int_t^{t+T} x_{\mathcal{T}}^T (Q + (K^i)^T R (K^i)) x_{\mathcal{T}} d\mathcal{T} + x_{t+T}^T P^i x_{t+T}. \tag{32}$$

3) (Policy improvement) update the control policy using

$$K^{i+1} = R^{-1}(B^T P^i + L^i) C^T (CC^T)^{-1}$$
$$L^{i+1} = RK^{i+1}C - B^T P^i. \tag{33}$$

This is a PI algorithm that uses the RL idea. It does not need the system matrix A in (1). It solves the coupled equations (15) and (16) online using data measured along the system trajectories in real time. However, it requires knowledge of the full states x. Therefore, an observer will be used during learning.

Remark 1: Note that the proposed Algorithm 3 requires measurement of all the states during learning. After finding the optimal output-feedback gain K, the control input $u = -Ky$ can be used to control the system and therefore the system states are no longer needed.

B. Observer

In this subsection, an adaptive observer for the linear CT system (1) with unknown matrix A is introduced to estimate the states, which are required during learning using the online IRL Algorithm 3.

Considering the system (1), by adding and subtracting $A_m x$, we have

$$\dot{x} = A_m x + \bar{A} x + Bu \quad y = Cx \tag{34}$$

where A_m is a Hurwitz matrix, the pair (A_m, C) is observable and $\bar{A} = A - A_m$. Now, the state observer for (1) is given by

$$\dot{\hat{x}} = A_m \hat{x} + \hat{A}\sigma(\hat{x}) + Bu + L(y - C\hat{x})$$
$$\hat{y} = C\hat{x} \tag{35}$$

where

- $[\hat{x}$ and $\hat{y}]$ state and output vectors of the observer, respectively;
- $[\sigma(\hat{x})]$ activation functions that can be chosen as $\hat{x}$ in linear systems;
- $[\hat{A}]$ estimate of the unknown matrix $\bar{A}$;
- $[L]$ observer gain that is selected such that $A - LC$ is a Hurwitz matrix.

Inspired by [32] and [33], which was done for nonlinear systems, the following update law is used for $\hat{A}$:

$$\dot{\hat{A}} = -\eta(\tilde{y}^T C A_m)^T \sigma(\hat{x})^T - \rho \|\tilde{y}\| \hat{A} \tag{36}$$

where $\tilde{y} = y - \hat{y}$, η is the learning rate, and ρ is a small positive number.

The boundness of the state estimation error $\tilde{x} = x - \hat{x}$ is shown in [32].

Remark 2: Note that it was shown in [32] that the state estimation of the observer converges to the true state. However, it was not guaranteed that the system parameters $\hat{A}$ converge to their true values. The primary goal of adaptive observers is to estimate the true states of a plant. Identification of unknown parameters is of secondary interest and is achieved, provided that a certain signal vector, referred to as the regressor, is persistently exciting. In this paper, since the knowledge of system drift dynamics A is not needed in Algorithm 3 and only the states of the system are required, the state estimation without parameter identification is considered, so that the state is estimated regardless of persistence of excitation condition.

Remark 3: There are some parameters that can be adjusted in the proposed algorithm, i.e., the reinforcement interval T and the observer parameters η and ρ. Vrabie *et al.* [11] showed that although the value of the T does not affect in any way the convergence property of the online algorithm, it is necessary to choose a big enough value for it to find a solution to the policy evaluation step (32) using the least squares method. For the observer parameters, it was discussed in [32] that larger learning rates can lead to faster convergence but extra care should be taken to avoid overshoot. It was also discussed that the Hurwitz matrix A_m has an effect on convergence as well

as accuracy of the state estimation and some consideration for choosing this matrix was given.

Remark 4: In the proposed method, both observer and Algorithm 3 are implemented simultaneously to get the output-feedback solution. Once Algorithm 3 converges, the observer is no longer needed. Like most adaptive control methods in the literature, we assume here that the system is time invariant. However, it can still be extended for slowly time-varying systems. To this end, if the system performance deteriorates after the initial learning phase, one can start another learning phase by rerunning both observer and Algorithm 3 simultaneously.

Remark 5: The proposed PI Algorithm 3 requires an initial admissible policy. If one knows that the system to be control is itself stable, which is true for many cases, then the initial control gain can be chosen as $K = 0$ and the admissibility of the initial policy is guaranteed without requiring any knowledge of A. Otherwise, the initial admissible policy can be obtained using some knowledge of A and using a robust control method, such as H_∞ control [11]. Note that the learning process does not require any knowledge of A. Moreover, Algorithm 3 is a PI algorithm and the IRL value iteration can be used to avoid the need for an initial admissible policy.

V. Simulation Results

In this section, the performance of the proposed online algorithm for finding a suboptimal output-feedback policy is evaluated for an X–Y table and an aircraft system. It is shown that the output-feedback control policy obtained by the proposed online Algorithm 3 is very close to the output-feedback control policy obtained by the offline Algorithm 2. Moreover, the results of the proposed method are compared with the results of the well-known offline method of Moerder and Calise [34].

A. X–Y Table

The X–Y table is a high-acceleration positioning mechanism, platform for wire bonding, or die bonding machines in the semiconductor industry. A simplified model of controller system, including the driver, the plant, and the sensor, is considered here [35]. In this paper, only the x-axis control problem is discussed.

The approximated model is given as follows [35]:

$$G_p = \frac{X(S)}{U(S)} = \frac{b}{s(s+a)} \tag{37}$$

where the input is the current and the output is the displacement. The nominal values of the parameters are chosen as $a = 11$ and $b = 1$, and they are considered to be unknown.

The control block diagram is shown in Fig. 1, where r is a reference step input and $u(t)$ is the current. An integrator has been added in the feed-forward path to achieve zero steady-state error. To suppress the noise in the velocity measurements, a first-order low-pass filter with cutoff frequency of 200 Hz is utilized. The parameter $\mathcal{T}$ in this diagram is equal to 0.005.

The performance output that should track the reference command r is the position x, and the tracking error is $e = r - x$. The system state is $X = [x \quad \dot{x} \quad \varepsilon \quad \dot{x}_F]^T$ and

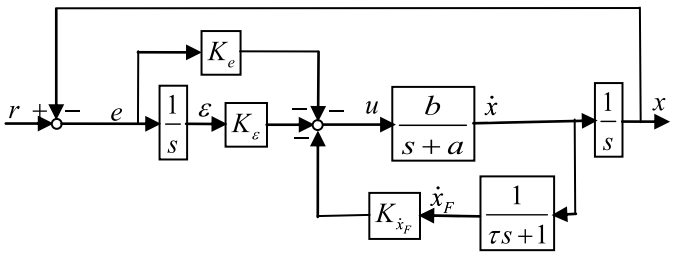


Fig. 1. Control block diagram for the X–Y table system.

the measured output is $y = [\varepsilon \quad e \quad \dot{x}_F]^T$, with $\varepsilon(t)$ the integrator output and $\dot{x}_F$ the filtered measurements of the velocity.

The linearized X–Y table dynamics about the nominal condition is augmented to include the filter and proportional plus integral compensator dynamics. The result is

$$\dot{X} = AX + Bu + Gr \quad y = CX + Fr \tag{38}$$

with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -a & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & \frac{1}{\mathcal{T}} & 0 & -\frac{1}{\mathcal{T}} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ b \\ 0 \\ 0 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$F = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

The output-feedback control input is

$$u = -Ky = -[\, K_\varepsilon \quad K_e \quad K_{\dot{x}_F} \,] \begin{bmatrix} \varepsilon \\ e \\ \dot{x}_F \end{bmatrix}. \tag{39}$$

By supposing $r = 0$, this system becomes a stabilization problem. Note that for the case, the reference trajectory is not zero, one use the method presented in [36]. Next, the offline Algorithm 2 and the online Algorithm 3 are used to find a suboptimal output-feedback control for this problem. The engineering judgment in optimal output-feedback design appears in the selection of Q and R. The cost function weighting matrices are chosen as $R = 1$ and

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}.$$

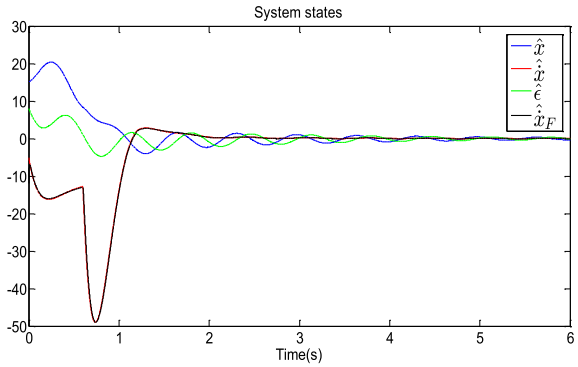


Fig. 2. States trajectories during the learning for X–Y table system.

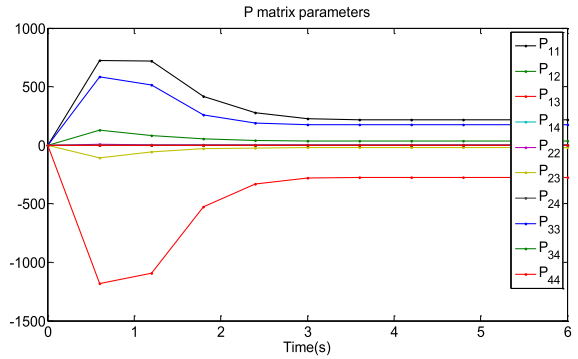


Fig. 3. Evolution of the parameters of the P matrix during the learning for X–Y table system.

Since we want the integral of the error ε goes to zero faster, a larger value in the Q matrix is chosen for this state. The initial control input in Algorithms 2 and 3 is chosen as $u = -K_0 y$ with $K_0 = [-1 \quad -10 \quad 0]$.

The final gain K obtained by Algorithm 2 is

$$K = \begin{bmatrix} -10.2470 & -16.8947 & 0.0234 \end{bmatrix}$$

and the P matrix is

$$P = \begin{bmatrix} 216.8111 & 16.8947 & -137.7146 & -0.0020 \\ 16.8947 & 2.4162 & -10.2470 & 0.0234 \\ -137.7146 & -10.2470 & 173.1187 & 0.0012 \\ -0.0020 & 0.0234 & 0.0012 & 0.0250 \end{bmatrix}.$$

Algorithm 2 needs the dynamic information of A. Next, we use the partial model-free Algorithm 3, which does not need to know the drift matrix A and is implemented online.

To implement this algorithm, the system state is initialized in $x_0 = [20 \quad -10 \quad 10 \quad -20]$ and the initial observer state is initialized in the origin. The simulation was conducted using the data obtained from the system at every 0.02 s. The least-square is solved in the real time after 25 number of data points are collected along with a single state trajectory for solving (32) for the kernel matrix P, and thus, the controller was updated every 0.5 s.

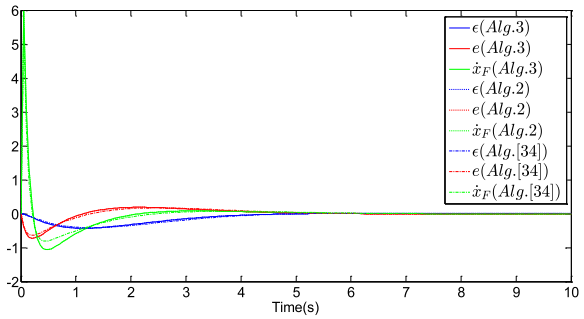


Fig. 4. Comparing the performance of Algorithms 2 and 3 with Moerder and Calise algorithm [34] for X-Y table system.

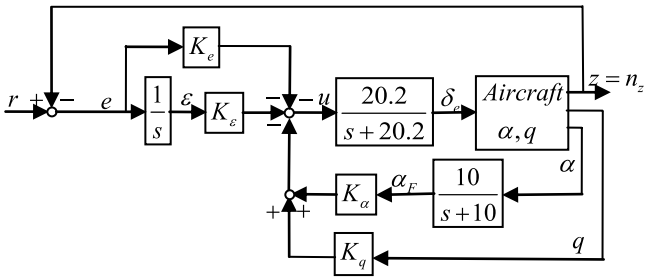


Fig. 5. G command system.

The gain K and the matrix P obtained online using Algorithm 3 are

$$K = [-10.2476 \ -16.9070 \ 0.0241]$$

$$P = \begin{bmatrix} 217.8261 & 16.9060 & -137.8182 & -0.0033 \\ 16.9060 & 2.4169 & -10.2486 & 0.0231 \\ -137.8182 & -10.2486 & 173.2079 & 0.0027 \\ -0.0033 & 0.0231 & 0.0027 & 0.0256 \end{bmatrix}.$$

In Fig. 2, the system state trajectories during learning are presented. Fig. 3 shows the convergence of the P matrix parameters using Algorithm 3.

Comparing the output-feedback gains obtained by Algorithms 2 and 3, it is clear that the results obtained by the online Algorithm 3 without knowing the system drift dynamics is close to the results obtained by the offline Algorithm 2.

We now compare the results of the proposed method with those obtained using the well-known offline method of Moerder and Calise [34]. Using this algorithm, the following output-feedback gain K is obtained:

$$K = [-10.0870 \quad -16.5732 \quad 1.0539].$$

Comparing the output-feedback gain obtained using the offline method of Moerder and Calise [34] and those obtained using Algorithms 2 and 3, it is obvious that their first and second elements of the output-feedback gains are closed together and their last elements are slightly different. Fig. 4 compares the performance of the measured output $y = [\varepsilon \quad e \quad \dot{x}_F]^T$ for these methods for a specific initial condition. This figure shows that the performance of the offline method of Moerder and Calise [34] is very close to the performance of Algorithms 2 and 3 for ε and e and is slightly better than Algorithms 2 and 3 for $\dot{x}_F$. This confirms that like the method of Moerder and

Calise [34], Algorithms 2 and 3 converge to a suboptimal solution. However, unlike the method of Moerder and Calise [34], the proposed Algorithm 3 does not require the knowledge of the system drift dynamics.

B. F-16 Normal Acceleration Regulator Design

The proposed adaptive suboptimal output-feedback control is now applied to the problem of designing an output-feedback normal acceleration regulator for an F-16 aircraft [37]. The control system is shown in Fig. 5, where n_z is the normal acceleration, q is the pitch rate, α is the angle of attack, r is the reference input in g, which is set to zero in here, and the control input $u(t)$ is the elevator actuator angle. An integrator output ε has been added in the feed-forward path to achieve zero steady-state error. To suppress the noise in the α measurements, a low-pass filter with the cutoff frequency of 10 rad/s is utilized to provide filtered measurements α_F of the angle of attack.

The state and the measured outputs are $X = [\alpha \quad q \quad \delta_e \quad \alpha_F \quad \varepsilon]^T$ and $y = [\alpha_F \quad q \quad e \quad \varepsilon]^T$ with δ_e the elevator actuator.

The linearized F-16 dynamics about the nominal flight condition described in [37] with short-period approximation is given as

$$\dot{x} = Ax + Bu \quad y = Cx \tag{40}$$

with

$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 & 0 & 0 \\ 0.82225 & -1.07741 & -0.17555 & 0 & 0 \\ 0 & 0 & -20.2 & 0 & 0 \\ 10 & 0 & 0 & -10 & 0 \\ -16.26 & -0.9788 & 0.4852 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 20.2 & 0 & 0 \end{bmatrix}^T$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 57.2958 & 0 \\ 0 & 57.2958 & 0 & 0 & 0 \\ -16.26 & -0.9788 & 0.4852 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The factor of 57.2958 is added to convert the angles from radians to degrees.

The output-feedback control input is

$$u = -Ky = -[K_\alpha \quad K_q \quad K_e \quad K_\varepsilon]y. \tag{41}$$

Next, the offline Algorithm 2 and the online Algorithm 3 are used to find a suboptimal output-feedback control for this problem. The cost function weighting matrices are chosen as $R = 0.1$ and

$$Q = \begin{bmatrix} 264 & 16 & 1 & 0 & 0 \\ 16 & 60 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 350 \end{bmatrix}.$$

Note that, it is desired to obtain good stability of the angle of attack α, the pitch rate q, and the tracking error ε, so that, they should be weighted in Q matrix. We do not care about

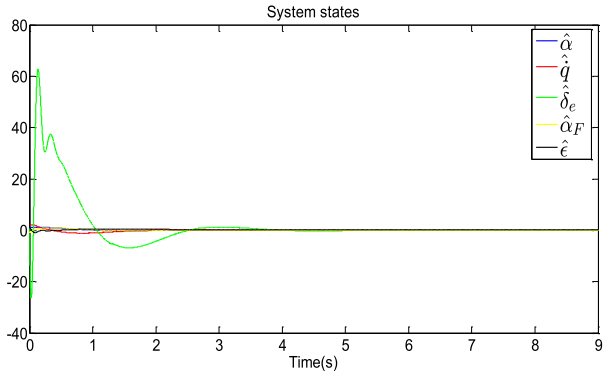


Fig. 6. States trajectories during the learning for F-16.

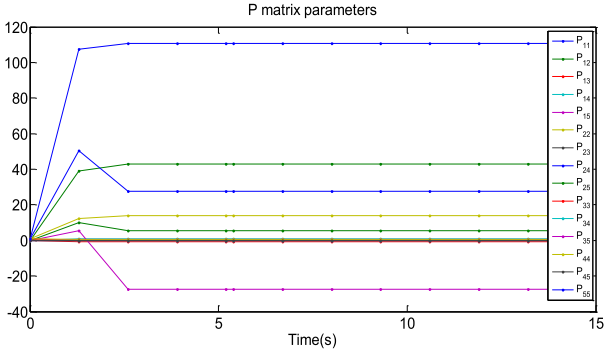


Fig. 7. Evolution of the parameters of the P matrix during the learning for F-16.

δ_e and α_F because, once other states are stable, they are also stable. Therefore, it is not necessary to weight them in the Q matrix. The initial control input in Algorithms 2 and 3 is chosen as $u = -K_0 y$ with $K_0 = [0 \quad 0 \quad 0 \quad 100]$. We now present the results of both algorithms.

The gain K obtained by Algorithm 2 is

$$K = [-0.0001 \quad -0.0526 \quad 4.5855 \quad 59.1708]$$

and the P matrix is

$$P = \begin{bmatrix} 119.3025 & 22.2239 & -0.3693 & 0.4397 & -15.0639 \\ 22.2339 & 13.8284 & -0.0372 & 0.0346 & 2.9356 \\ -0.3693 & -0.0372 & 0.0061 & -0.0001 & 0.2929 \\ 0.4397 & 0.0346 & -0.0001 & 0.5000 & 0.0064 \\ -15.0639 & 2.9356 & 0.2929 & 0.0064 & 28.2385 \end{bmatrix}.$$

Algorithm 2 needs the dynamic information of A. Next, we use the partial model-free Algorithm 3, which does not require the drift matrix A. To implement this algorithm, the system state is initialed in $x_0 = [1 \quad 2 \quad 1 \quad -1 \quad 1]$ and the initial observer state is initialized in the origin. The simulation was conducted using data obtained from the system at every 0.02 s. The least-square is solved in real-time after 50 number of data points are collected along with a single state trajectory for solving (32) for the kernel matrix P, and thus the controller was updated in every 1 s. The gain K and matrix P obtained

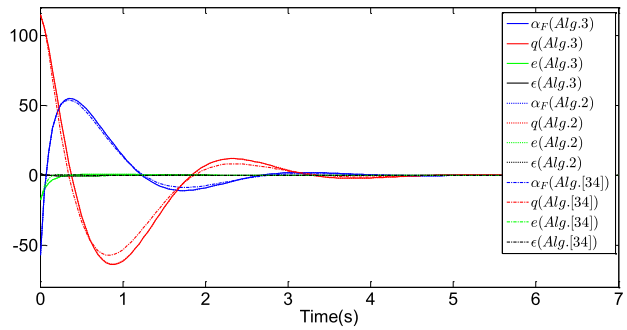


Fig. 8. Comparing the performance of Algorithms 2 and 3 with Moerder and Calise algorithm [34] for F-16.

online using Algorithm 3 are

$$K = [-0.0000 \quad -0.0428 \quad 4.5022 \quad 59.6871]$$

$$P = \begin{bmatrix} 110.8028 & 21.4047 & -0.3626 & 0.4380 & -13.7254 \\ 21.4047 & 13.8601 & -0.0340 & 0.0349 & 2.7539 \\ -0.3626 & -0.0340 & 0.0065 & -0.0001 & 0.2954 \\ 0.4380 & 0.0349 & -0.0001 & 0.5012 & 0.0067 \\ -13.7254 & 2.7539 & 0.2954 & 0.0067 & 27.4672 \end{bmatrix}.$$

In Fig. 6, the system state trajectories during learning are presented. Fig. 7 shows the convergence of P matrix parameters.

Comparing the output-feedback gains obtained by Algorithms 2 and 3, it is clear that the results obtained by the online Algorithm 3 without knowing the system internal dynamics is close to the results obtained by the offline Algorithm 2.

We now compare the results of the proposed method with those obtained using the well-known offline method of Moerder and Calise [34]. Using this algorithm, the following output-feedback gain K is obtained:

$$K = [-0.6925 \quad -0.2880 \quad 4.9601 \quad 63.1450].$$

Comparing the output-feedback gain obtained using the offline method of Moerder and Calise [34] and those obtained using Algorithms 1 and 2, it is obvious that their first and second elements of the output-feedback gains are closed together and their last elements are slightly different. Fig. 8 compares the performance of the measured output $y = [\alpha_F \quad q \quad e \quad \varepsilon]^T$ for these methods for a specific initial condition. This figure shows that the performance of the offline method of Moerder and Calise [34] is very close to the performance of Algorithms 2 and 3. This confirms that like the method of Moerder and Calise [34], Algorithms 2 and 3 converge to a suboptimal solution. However, unlike the method of Moerder and Calise [34], the proposed Algorithm 3 does not require the knowledge of the system drift dynamics.

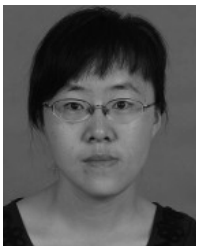
VI. Conclusion

An online IRL-based algorithm is presented to learn a suboptimal output-feedback control law for linear CT systems. The proposed method does not require the knowledge of the system drift dynamics. An adaptive observer is used to provide the knowledge of the system states during learning for the IRL algorithm. The proposed method is tested on two real-world control applications, including X–Y table and F-16 flight control.

References

[1] F. L. Lewis, D. Vrabie, and V. Syrmos, *Optimal Control*, 3rd ed. New York, NY, USA: Wiley, 2012.
[2] P. Ioann and B. Fidan, *Adaptive Control Tutorial*. Philadelphia, PA, USA: SIAM, 2006.
[3] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 1998.
[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control: Approximate Dynamic Programming*, 4th ed. Belmont, MA, USA: Athena Scientific, 2012.
[5] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses Of Dimensionality*. New York, NY, USA: Wiley, 2007.
[6] R. A. Howerd, *Dynamic Programming and Markov Processes*. Cambridge, MA, USA: MIT Press, 1960.
[7] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Trans. Syst. Man Cybern., B, Cybern.*, vol. 41, no. 1, pp. 14–25, Feb. 2011.
[8] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral Q-learning and explorized policy iteration for adaptive optimal control of continuous-time linear systems," *Automatica*, vol. 48, no. 11, pp. 2850–2859, Nov. 2012.
[9] Y. Jiang and Z. P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, Oct. 2012.
[10] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M. B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, Apr. 2014.
[11] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, Feb. 2009.
[12] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.
[13] H. N. Wu and B. Luo, "Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H_∞ control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1884–1895, Dec. 2012.
[14] K. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous infinite-time horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, May 2010.
[15] H. Modares, M. B. Naghibi-Sistani, and F. L. Lewis, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, Jan. 2014.
[16] X. Xu, Z. Hou, C. Lian, and H. He, "Online learning control using adaptive critic designs with sparse kernel machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 762–775, May 2013.
[17] T. Dierks and S. Jagannathan, "Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1118–1129, Jul. 2012.
[18] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst. Mag.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.
[19] F. Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.
[20] S. N. Balakrishnan, J. Ding, and F. L. Lewis, "Issues on stability of ADP feedback controllers for dynamical system," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 38, no. 4, pp. 913–917, Aug. 2008.
[21] D. Kleinman, "On an iterative technique for Riccati equation computations," *IEEE Trans. Autom. Control*, vol. 13, no. 1, pp. 114–115, Feb. 1968.
[22] D. Vrabie and F. L. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Netw.*, vol. 22, no. 3, pp. 237–246, Apr. 2009.

[23] C. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Dept. Comput. Sci. & Eng., Cambridge Univ., Cambridge, U.K., 1989.
[24] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proc. Amer. Control Conf.*, Baltimore, MD, USA, Jun. 1994, pp. 3475–3476.
[25] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control*, D. A. White and D. A. Sofge, Eds. New York, NY, USA: Van Nostrand Reinhold, 1992.
[26] P. J. Werbos, "Neural networks for control and system identification," in *Proc. 28th IEEE Conf. Decision Control*, Dec. 1989, pp. 260–265.
[27] J. Si and Y. T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, May 2001.
[28] Z. Ni, H. He, and J. Wen, "Adaptive learning in tracking control based on the dual critic network design," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 6, pp. 913–928, Jun. 2013.
[29] H. Hea, Z. Ni, and J. Fu, "A three-network architecture for on-line learning and optimization based on adaptive dynamic programming," *Neurocomputing*, vol. 78, no. 1, pp. 3–13, Feb. 2012.
[30] V. Kucera and C. E. De Souzasa, "Necessary and sufficient condition for feedback stabilizability," *Automatica*, vol. 31, no. 9, pp. 1357–1359, Sep. 1995.
[31] J. Gadewadikar, F. L. Lewis, and M. Abu-Khalaf, "Necessary and sufficient conditions for H_∞ static output-feedback control," *J. Guid. Control Dyn.*, vol. 29, no. 4, pp. 915–920, Jul./Aug. 2006.
[32] F. Abdollahi, H. A. Talebi, and R. V. Patel, "A stable neural network observer with application to flexible-joint manipulators," in *Proc. 9th Int. Conf. Neural Inform. Process.*, 2006, pp. 1910–1914.
[33] F. Abdollahi, H. A. Talebi, and R. V. Patel, "A stable neural network based observer with application to flexible-joint manipulators," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 118–129, Jan. 2006.
[34] D. D. Moerder and A. J. Calise, "Convergence of a numerical algorithm for calculating optimal output feedback gains," *IEEE Trans. Autom. Control*, vol. 30, no. 9, pp. 900–903, Sep. 1985.
[35] H. Ding and J. Wu, "Point-to-point motion control for a high-acceleration positioning table via cascaded learning schemes," *IEEE Trans. Ind. Electron.*, vol. 54, no. 5, pp. 2735–2744, Oct. 2007.
[36] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Trans. Autom. Control*, to be published.
[37] B. L. Stevens and F. L. Frank, *Aircraft Control and Simulation*, 2nd ed. New York, NY, USA: Wiley, 2003.



Lemei M. Zhu received the B.S., M.S., and Ph.D. degrees from the Beijing Institute of Technology, Beijing, China, in 2001, 2005, and 2014, respectively.

She was with the University of Texas at Arlington, Arlington, TX, USA, as a Visiting Student, from 2012 to 2013. She is currently with the North China Institute of Science and Technology, Beijing. Her current research interests include adaptive control, nonlinear systems, and their applications to spacecraft control.

Hamidreza Modares received the B.S. degree from the University of Tehran, Tehran, Iran, in 2004, and the M.S. degree from the Shahrood University of Technology, Shahrud, Iran, in 2006. He is currently pursuing the Ph.D. degree with the University of Texas at Arlington, Arlington, TX, USA.

He joined the Shahrood University of Technology as a Faculty Lecturer from 2006 to 2009. His current research interests include optimal control, reinforcement learning, approximate dynamic programming, neural adaptive control, and pattern recognition.

Gan Oon Peen received the Ph.D. degree from the National University of Singapore, Singapore, in 1997.

He was a Post-Doctoral Researcher with the Centre for Process System Engineering, Imperial College London, London, U.K. He is a Senior Scientist and Group Manager with the Singapore Institute of Manufacturing Technology and a Technical Lead with the National RFID Centre, Singapore. He is also an Adjunct Associate Professor with Nanyang Technological University, Singapore. His current research interests include control, automation, and RFID data analytics.

Frank L. Lewis (S'70–M'81–SM'86–F'94) received the bachelor's degree in physics/electrical engineering and the M.S.E.E. degree from Rice University, Houston, TX, USA, the M.S. degree in aeronautical engineering from the University of West Florida, Pensacola, FL, USA, and the Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA, USA.

He is a U.K. Chartered Engineer, a UTA Distinguished Scholar Professor, a UTA Distinguished Teaching Professor, and the Moncrief-O'Donnell Chair with the University of Texas at Arlington Research Institute, Fort Worth, TX, USA, and a Qian Ren Thousand Talents Professor with Northeastern University, Shenyang, China. He is a Distinguished Visiting Professor with the Nanjing University of Science and Technology and a Project 111 Professor with Northeastern University. He is the Founding Member of the Board of Governors of the Mediterranean Control Association. He has authored numerous journal special issues, journal papers, and 14 books, including *Optimal Control, Aircraft Control, Optimal Estimation*, and *Robot Manipulator Control*, which are used as university textbooks worldwide, and holds six U.S. patents. His current research interests include feedback control, intelligent systems, cooperative control systems, and nonlinear systems.

Dr. Lewis is a member of the National Academy of Inventors, a fellow of the International Federation of Automatic Control, a fellow of the U.K. Institute of Measurement and Control, and a PE Texas. He was a recipient of the Fulbright Research Award, the NSF Research Initiation Grant, the ASEE Terman Award, the International Neural Network Society's Gabor Award, the U.K. Inst Measurement & Control Honeywell Field Engineering Medal, the IEEE Computational Intelligence Society's Neural Networks Pioneer Award, the Outstanding Service Award from Dallas IEEE Section, and the Texas Regents Outstanding Teaching Award in 2013. He was selected as an Engineer of the Year by Ft. Worth IEEE Section. He was listed in Ft. Worth Business Press Top 200 Leaders in Manufacturing.

Baozeng Yue received the bachelor's degree from Henan University, Kaifeng, China, and the master's degree from the Harbin Institute of Technology, Harbin, China, in 1981 and 1993, respectively, and the Ph.D. degree from Tsinghua University, Beijing, China, in 1998.

He was with Shanghai Jiao Tong University, Shanghai, China, as a Post-Doctoral Researcher from 1998 to 2000. In 2000, he became a Faculty Member with the Department of Mechanics at the Beijing Institute of Technology (BIT), Beijing. He was a Visiting Scholar, founded by the China Scholarship Council, with the Center for Applied Mechanics and Computer Science, Amsterdam, The Netherlands, in 2006. He was a Visiting Professor with the City University of Hong Kong, Hong Kong, in 2012. He was a Senior Visiting Scholar with the University of Texas at Arlington, Arlington, TX, USA, in 2013, founded by the China Scholarship Council. He is a Professor of the School of Aerospace Engineering at BIT. He has supervised seven Ph.D. and 15 M.S. theses, including two Ph.D. theses of foreign Ph.D. students. He has authored and co-authored more than 100 research papers, and has published one book on nonlinear liquid sloshing dynamics. His current research interests include dynamics and control, nonlinear dynamics, and spacecraft dynamics.

Dr. Yue is a member of the Committee for Dynamics and Control, Chinese Society of Theoretic and Applied Mechanics.