AIMS Mathematics, 11(4): 9284–9302.
DOI:10.3934/math.2026383
Received: 18 January 2026
Revised: 18 March 2026
Accepted: 31 March 2026
Published: 07 April 2026

https://www.aimspress.com/journal/Math

Research article

Inverse optimal output tracking for continuous-time linear systems based on inverse reinforcement learning

Yi Mo[1], Jingling Zhao[2], Kunyu Xiang[1] and Dengguo Xu[2,*]

[1] Industrial Internet School, Guangxi Vocational and Technical Institute of Industry, Nanning 530001, China

[2] School of Automation, Guangxi University of Science and Technology, Liuzhou 545006, China

* **Correspondence:** Email: xudengguo@gxust.edu.cn.

Abstract: In this study, we address a value function reconstruction problem in optimal output tracking of continuous-time linear time-invariant systems. The objective is to determine the state weight matrix within the value function that results in the specified optimal control law. First, by augmenting the system state and the desired tracking dynamic variable, the optimal tracking problem with a discount value function is turned into a linear regulator problem. Inverse optimal output tracking is thus simplified as an inverse optimal control of the augmented system. Second, a model-based inverse reinforcement learning algorithm is suggested to calculate the state weight matrix in the augmented value function. This algorithm updates the cost matrix via gradient descent and calculates the weight matrix through inverse optimal control. After continuous iterations, the weight matrix converges to a steady state. Third, astringency of the algorithm is rigorously analyzed, and the stability of the corresponding system is confirmed. Finally, the proposed algorithm's effectiveness is confirmed through simulation, illustrating that the system output asymptotically tracks a predetermined reference trajectory.

Keywords: inverse optimal output tracking; continuous-time linear systems; inverse reinforcement learning; inverse optimal control
Mathematics Subject Classification: 49N05, 49N45, 93C05

1. Introduction

The tracking control problem has been of wide concerne to engineers and the academic community. From the perspective of optimal control, in engineering applications, engineers and technicians should consider reducing energy consumption and costs to optimize the performance of systems [1]. However, in practice, the objective function of an optimal tracking system is often unknown. Inferring the

objective function for a tracking problem based on a known optimal control law is referred to as the inverse optimal tracking problem. Therefore, in this study, we address the inverse optimal output tracking (IOOT) problem for continuous-time linear systems using an inverse reinforcement learning (RL) algorithm.

Optimal control aims to find a control policy to optimize a given value function. RL is a powerful tool for optimal control because it can solve model-free problems. RL algorithms for the design of optimal control include two iterative methods: policy iteration [2] and value iteration [3, 4]. Policy iteration requires an initial stable control law, whereas value iteration operates without this prerequisite. Inverse optimal control (IOC), as introduced by Bellman and Kalaba, addresses the inverse problem in optimal control [5]. Their goal was to identify the criterion function based on a given optimal control law and descriptive equations. Subsequently, some IOC results for control systems were formally introduced by Kalman [6], mainly concentrating on identifying all performance functions linked to optimal stable feedback control for a given differential dynamic system. On this groundbreaking study, IOC has been widely explored and developed in discrete-time systems [7], nonlinear systems [8], nonquadratic cost functions [9], and beyond. The advancement of IOC theory was illustrated across various practical fields, including the optimized control of aircraft systems [10], robotics control [11], and electric power systems [12].

Inverse optimal control is generally developed offline when the system dynamics are completely known. Building on the principles of RL and IOC, inverse RL was initially introduced in [13] with the goal of reconstructing the reward function from observed optimum behaviors. The ideology is extremely significant in apprenticeship or imitation learning [14], serving as a foundation for the exploration of various inverse RL methods, such as Bayesian inverse RL [15], adversarial inverse RL [16], and deep inverse RL [17]. In the past few years, inverse RL has been applied to some new research fields such as trajectory tracking [18], autonomous driving [19], and energy management [20]. In addition, the adaptive optimal control, achieved by integrating optimal control with adaptive control in [21], employs an adaptive optimization framework to address the challenge of estimating unknown parameters in nonlinear systems. Moreover, in IOC theory and inverse dynamic game theory, inverse RL plays an indispensable role in identifying unknown objective value functions. In reconstructing unknown value functions, inverse RL primarily consists of model-based algorithms [22–24], partially model-free algorithms [25], and completely model-free algorithms [26, 27]. The results have been extended to inverse dynamic game systems [28–30] and inverse Q-learning [31–33].

In recent years, based on inverse RL, the IOC problem of objective value function reconstruction has been widely studied. The latest research efforts have proposed inverse RL algorithms for inverse optimal control of linear systems and nonlinear systems. On the other hand, the optimal output tracking problem is prevalent in engineering applications and represents a challenging problem in control system design, due to the difficulty of designing an optimal controller that ensures accurate tracking of an unstable reference signal [34]. One approach to tackle the challenge is to employ a discount value function, which reduces the optimal output tracking to solving a linear regulator problem for an augmented system [35]. Moreover, in practical control system design, the objective function for optimal tracking is usually unknown. However, to the best of our knowledge, the literature on inverse optimal tracking for reconstructing such objective functions remains scarce. The inherent difficulties in solving optimal tracking problems make their inverse counterparts particularly challenging. It should be noted that inverse optimal output tracking is a reference-tracking-driven inverse problem,

whereas IOC is driven by state regulation. Given this context, this paper studies the IOOT problem for continuous-time linear systems. The state weight matrix in the objective function of the optimal tracking system is iteratively derived using the inverse RL principle.

The primary contributions of this work are encapsulated by two aspects below:

1) The IOOT model of the continuous-time linear system is established, and the inverse optimal tracking problem is turned into an IOC design for the constructed augmenting linear system. At present, there is almost no published literature on the inverse optimal tracking problem. This work establishes an inverse RL-driven methodology for inverse optimal output tracking control, which presents a previously unexplored technical approach.

2) We propose a model-based inverse RL algorithm to address the inverse optimal output tracking in continuous-time linear systems. The state weight matrix in the objective function is computed iteratively. Furthermore, we establish astringency and stability results to offer rigorous theoretical analysis, thereby ensuring the algorithm's effectiveness. In the design of control systems based on inverse RL, current literature only focuses on identifying objective function in the regulator problem. In [36], the author introduced an inverse RL algorithm grounded in output feedback IOC for continuous-time linear systems. This paper presents an inverse RL algorithm aimed at IOOT, thereby extending the current inverse optimal output feedback control findings to encompass inverse optimal tracking scenarios.

This paper is organized as follows: Section 2 addresses the formulation of the inverse optimal output tracking problem for continuous-time linear time-invariant systems. In Section 3, we introduce a model-based inverse RL algorithm designed for inverse optimal output tracking. Section 4 delves into an analysis of the proposed algorithm, confirming its convergence and stability. Sections 5 and 6 are dedicated to presenting simulation results and drawing conclusions, respectively.

2. Problem formulation

Consider the following continuous-time linear system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), x(0) = x_0, \\ y &= Cx(t), \end{aligned} \tag{2.1}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ indicate the system state and control input, respectively. $x(0) = x_0$ denotes the initial value of System(2.1). $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are known matrices, and $C \in \mathbb{R}^{r\times n}$ is an output matrix.

Define an infinite time value function as

$$J(x, y_d, u, t) = \frac{1}{2}\int_t^\infty \left[(Cx - y_d)^T Q(Cx - y_d) + u^T Ru\right] \cdot e^{-\gamma(\tau - t)} \mathrm{d}\tau, \tag{2.2}$$

where $\gamma > 0$ is a discount factor, y_d is a desired reference tracking signal, and Q and R are positive semi-definite state weights positive definite input weights, respectively.

Inverse optimal output tracking aims to find the weight matrix Q in the value function (2.2) given input weight R, ensuring that the tracking control law is optimal. Inverse optimal tracking is the inverse problem of optimal tracking.

Assumption 1. *Assume that the desired reference trajectory y_d is formed by the following dynamical system:*

$$\dot{y}_d = Fy_d,\ y_d(0) = y_{d0}, \tag{2.3}$$

where F is a real matrix, and y_{d0} is the initial value.

Assumption 2. *In System* (2.1)*, we assume that the output dimension r is at least equal to the state dimension n, meaning $r \geq n$.*

Remark 1. *The systems with an output dimension that is equal to or greater than the state dimension are frequently encountered in practical applications, such as liquid level control systems [37] and motor control systems [38].*

From the initial value problem (2.3), the desired reference trajectory can be expressed as

$$y_d(t) = e^{Ft}y_d(0) \equiv L_1(t)\, y_{d0}. \tag{2.4}$$

Referring to [35], assuming that the admissible control strategy for the tracking problem is expressed as

$$u = K_s x + K' y_d, \tag{2.5}$$

then the solution to System (2.1) is represented as

$$x(t) = e^{(A+BK)t}x_0 + \left(\int_0^t e^{(A+BK)(t-\tau)}BK' e^{F\tau}\mathrm{d}\tau\right)y_{d0} \equiv L_2(t)\, x_0 + L_3(t)\, y_{d0}. \tag{2.6}$$

According to the results in [35], the following lemma holds.

Lemma 1. *Consider the optimal output tracking problem with the state equation, the objective function, and the dynamical reference trajectory defined by* (2.1)–(2.3)*, respectively. Consequently, the value function* (2.2) *for the control policy* (2.5) *can be expressed in the following standard form:*

$$J(x(t), y_d(t)) = V(x(t), y_d(t)) = \frac{1}{2}\left[x^T(t)\, y_d^T(t)\right] P \left[x^T(t)\, y_d^T(t)\right]^T, \tag{2.7}$$

where the matrix P is positive definite.

Now, we introduce the following augmenting state:

$$\theta(t) = \left[x^T(t) \quad y_d^T(t)\right]^T. \tag{2.8}$$

By combining (2.1) and (2.3), the augmented continuous-time linear system can be constructed as

$$\dot{\theta} = T\theta + B_1 u, \tag{2.9}$$

where $T = \begin{bmatrix} A & 0 \\ 0 & F \end{bmatrix}$, $B_1 = \begin{bmatrix} B \\ 0 \end{bmatrix}$. By introducing the augmented state, we can reformulate the value function (2.2) into the following augmented form:

$$J(\theta, u, t) = \frac{1}{2}\int_t^\infty \left[\theta^T \bar{Q}\theta + u^T R u\right] \cdot e^{-\gamma(\tau - t)}\mathrm{d}\tau, \tag{2.10}$$

where $\bar{Q} = C_1^T Q C_1$, $C_1 = [C \quad -I]$, and I is an identity matrix.

According to Lemma 1, it follows that

$$\frac{1}{2}\int_t^{\infty}\left[\theta^T\bar{Q}\theta + u^T R u\right]\cdot e^{-\gamma(\tau-t)}\mathrm{d}\tau = \frac{1}{2}\theta^T P\theta. \tag{2.11}$$

By differentiating both sides of (2.11) regarding the trajectory defined by (2.9), we obtain the Bellman equation

$$0 = (T\theta + B_1 u)^T P\theta + \theta^T P(T\theta + B_1 u) - \gamma\theta^T P\theta + u^T R u + \theta^T\bar{Q}\theta. \tag{2.12}$$

Consider that the admissible control strategy (2.5) can be simplified as

$$u = K\theta, \tag{2.13}$$

where $K = \left[K_s \quad K'\right]$. Substituting (2.13) into (2.12) yields

$$(T + B_1 K)^T P + P(T + B_1 K) + \bar{Q} - \gamma P + (K)^T R K = 0. \tag{2.14}$$

Based on (2.12), the function is defined as

$$H(\theta, u, P) = (T\theta + B_1 u)^T P\theta + \theta^T P(T\theta + B_1 u) - \gamma\theta^T P\theta + \theta^T\bar{Q}\theta + u^T R u, \tag{2.15}$$

which is a Hamiltonian function.

Lemma 2. *The optimal control solution for the augmented system* (2.9) *and value function* (2.10) *is obtained from*

$$u = K^*\theta = -R^{-1}B_1^T P\theta, \tag{2.16}$$

where P is the positive definite symmetric solution to the augmented ARE

$$A_1^T P + PA_1 - PB_1 R^{-1}B_1^T P + \bar{Q} = 0 \tag{2.17}$$

with $A_1 = T - 0.5\gamma I$.

Proof. The existence of optimal control requires that the following stationarity condition holds:

$$\frac{\partial H}{\partial u} = B_1^T P\theta + Ru = 0, \tag{2.18}$$

which can derive (2.16). Putting (2.16) into (2.12) yields

$$(T\theta + B_1 K^*\theta)^T P\theta + \theta^T P(T\theta + B_1 K^*\theta) - \gamma\theta^T P\theta + \theta^T C_1^T Q C_1\theta + [K^*\theta]^T R K^*\theta = 0. \tag{2.19}$$

It follows that

$$T^T P + (K^*)^T B_1^T P + PT + PB_1 K^* - \gamma P + (K^*)^T R K^* + C_1^T Q C_1 = 0. \tag{2.20}$$

Substituting (2.16) into (2.20) yields

$$(T - 0.5\gamma I)^T P + P(T - 0.5\gamma I) - PB_1 R^{-1}B_1^T P + C_1^T Q C_1 = 0, \tag{2.21}$$

which is (2.17).

This completes the proof. □

The optimal tracking problem entails a known system and a desired signal, aiming to determine the optimal tracking law that minimizes a certain objective function. However, in practical applications, determining the weight matrix within the objective function is a challenging task. Transforming the optimal tracking problem into designing an optimal control law, this paper aims to convert the IOOT problem into designing an IOC law. Under the condition of a known input weight matrix in an objective function, we develop an inverse RL algorithm to compute the state weight matrix for the optimal tracking system. Specifically, given the input weight R, the goal of IOOT is to identify the state weight $\bar{Q}$ in (2.10) so that the prescribed tracking control law K^* becomes optimal concerning the associated objective function, while guaranteeing that the output accurately follows the desired tracking signal $y_d(t)$.

3. Model-based inverse RL algorithm

We design a model-based inverse RL algorithm for inverse optimal output tracking of a continuous-time linear time-invariant system in this section. Given the control law K_0, system dynamics (A, B), command generator system, and input weight R, the state weight $\bar{Q}$ in the performance metric is iteratively calculated.

3.1. Optimal control on a given value function

First, we must initialize $\bar{Q}^{(0)} = \bar{Q}^{(0)T}$, $R = R^T > 0$. Given the known dynamics A_1 and B_1, it is necessary to determine the initial cost matrix $P^{(0)}$ from the ARE

$$A_1^T P^{(0)} + P^{(0)} A_1 - P^{(0)} B_1 R^{-1} B_1^T P^{(0)} + \bar{Q}^{(0)} = 0. \tag{3.1}$$

Given our freedom to select $\bar{Q}^{(0)}$, we can choose it so that $\left(A_1, \sqrt{\bar{Q}^{(0)}}\right)$ is observable. It is important to observe that if an optimal solution exists, it is unique. The initial control law can be calculated as $K^{(0)} = -R^{-1} B_1^T P^{(0)}$.

3.2. Parameter update under gradient descent

Our goal is to monitor the difference between the gain K in its current iteration and the known control gain K_0. Specifically, when examining the difference in the ith iteration, we refer to this difference as the function

$$s^{(i)} = K^{(i)} - K_0 = -R^{-1} B_1^T P^{(i)} - K_0. \tag{3.2}$$

Then, we define error function

$$E^{(i)}(P) = \mathrm{Tr}\left(s^{(i)T} s^{(i)}\right), \tag{3.3}$$

where $Tr(.)$ is the matrix trace. The error is a function concerning matrix P that needs to be minimized. Utilizing the gradient descent theory [39], it follows that

$$P^{(i+1)} = P^{(i)} - \alpha \frac{\partial E^{(i)}(P)}{\partial P} = P^{(i)} - \alpha \frac{\partial\, Tr\left(s^{(i)T} s^{(i)}\right)}{\partial P}, \tag{3.4}$$

where α is a positive learning rate.

Substituting $s^{(i)T}$ and $s^{(i)}$ into the product expansion, we obtain

$$\begin{aligned} s^{(i)T}s^{(i)} =& \left(-P^{(i)}B_1R^{-1} - K_0^T\right)\left(-R^{-1}B_1^TP^{(i)} - K_0\right) \\ =& P^{(i)}B_1R^{-1}R^{-1}B_1^TP^{(i)} + P^{(i)}B_1R^{-1}K_0 + K_0^TR^{-1}B_1^TP^{(i)} + K_0^TK_0. \end{aligned} \tag{3.5}$$

Then, using the linearity of the trace to calculate the trace, while ignoring the constant terms that are independent of P, we get

$$\mathrm{Tr}\left(s^{(i)T}s^{(i)}\right) = \mathrm{Tr}\left(P^{(i)}B_1^TR^{-1}R^{-1}B_1^TP^{(i)}\right) + \mathrm{Tr}\left(P^{(i)}B_1R^{-1}K_0\right) + \mathrm{Tr}\left(K_0^TR^{-1}B_1^TP^{(i)}\right) + \mathrm{Tr}\left(K_0^TK_0\right). \tag{3.6}$$

To find the extremum, taking the partial derivatives of the terms of Eq (3.6) with respect to $P^{(i)}$, we obtain

$$\begin{aligned} \frac{\partial E^{(i)}(P)}{\partial P^{(i)}} &= P^{(i)}B_1R^{-1}R^{-1}B_1^T + B_1R^{-1}R^{-1}B_1^TP^{(i)} + K_0^TR^{-1}B_1^T + B_1R^{-1}K_0 \\ &= -\left(K^{(i)} - K_0\right)^TR^{-1}B_1^T - B_1R^{-1}\left(K^{(i)} - K_0\right) \\ &= -s^{(i)T}R^{-1}B_1^T - B_1R^{-1}s^{(i)}. \end{aligned} \tag{3.7}$$

Note that the norm $\left\|s^{(i)}\right\|$ is bounded for each i if $P^{(0)}$ in $K^{(0)} = -R^{-1}B_1^TP^{(0)}$ is a solution to ARE (3.1). Therefore, given that $\left\|s^{(i)}\right\| > \left\|s^{(i+1)}\right\|$, for $i = 0, 1, \ldots$, we have

$$\left\|s^{(0)}\right\| > \left\|s^{(1)}\right\| > \cdots \geq 0, \tag{3.8}$$

where $\|\cdot\|$ denotes the 2-norm.

3.3. Updating via inverse optimal control

The final step involves updating $\bar{Q}^{(i)}$ using $P^{(i)}$ derived from the previous gradient descent step. By substituting the updated matrix $P^{(i)}$ into ARE (3.1), we can thus obtain

$$\bar{Q}^{(i)} = -A_1^TP^{(i)} - P^{(i)}A_1 + P^{(i)}B_1R^{-1}B_1^TP^{(i)}. \tag{3.9}$$

We continue repeating the above steps until $0 \leq E^{(i)} < \epsilon$, where $\epsilon \in \mathbb{R}^+$ represents a predefined accuracy. The combination of the resulting $\bar{Q}^*$, the initialized R, and the given (A_1, B_1) ensures that the value function (2.2) achieves optimality. Hence, we get a new ARE and the control gain

$$A_1^TP^* + P^*A_1 - P^*B_1R^{-1}B_1^TP^* + \bar{Q}^* = 0, \tag{3.10}$$

$$K^* = -R^{-1}B_1^TP^* = K_0, \tag{3.11}$$

where the matrix P^* is a positive solution to ARE (3.10), $\bar{Q}^* = C_1^TQ^*C_1$, $\bar{Q}^*$ represents the optimal weight matrix, and K^* denotes the optimal control gain.

Through continuous iterative calculation, $P^{(i)}$ converges to P^*, and $\bar{Q}^{(i)}$ converges to $\bar{Q}^*$. Thus, the inverse optimal tracking weight Q^* can be derived from the converged weight $\bar{Q}^*$ via $Q^* = \left(C_1C_1^T\right)^{-1}C_1\bar{Q}^*C_1^T\left(C_1C_1^T\right)^{-1}$.

To summarize, we consolidate the entire procedure in Algorithm 1.

Algorithm 1 Model-based inverse RL algorithm

(1). Known (A_1, B_1), given $R = R^T > 0$, discount factor γ and known control gain K_0. Initialize $\bar{Q}^{(0)} = \bar{Q}^{(0)T} > 0$, obtain $P^{(0)}$ by solving ARE and set $i = 0$.

(2). Compute

$$K^{(i)} = -R^{-1}B_1^T P^{(i)}, \tag{3.12}$$

and

$$s^{(i)} = K^{(i)} - K_0. \tag{3.13}$$

(3). Update positive definite cost matrix as

$$P^{(i+1)} = P^{(i)} + \alpha\left(s^{(i)T}R^{-1}B_1^T + B_1R^{-1}s^{(i)}\right). \tag{3.14}$$

(4). Update the reward weighting matrix as

$$\bar{Q}^{(i+1)} = -A_1^T P^{(i+1)} - P^{(i+1)}A_1 + P^{(i+1)}B_1R^{-1}B_1^T P^{(i+1)}. \tag{3.15}$$

(5). Increment i by (1). Continue with steps (2) and (3) until $E^{(i)}(P) = \mathrm{Tr}\left(s^{(i)T}s^{(i)}\right) < \epsilon$, where $\epsilon > 0$ is a small constant.

(6). Continuously repeating iterative calculations such that $P^{(i)}$ converges to P^* and $\bar{Q}^{(i)}$ converges to $\bar{Q}^*$. Solve for Q^* from $\bar{Q}^*$ by the formula

$$Q^* = \left(C_1C_1^T\right)^{-1}C_1\bar{Q}^*C_1^T\left(C_1C_1^T\right)^{-1}. \tag{3.16}$$

Remark 2. *In Algorithm 1, the derivation of optimal tracking weight matrix Q^* from the augmented weight matrix $\bar{Q}^*$ requires the invertibility condition of C^TC. Assumption 2 is therefore introduced to guarantee this fundamental requirement, which represents a typical scenario in practical control systems.*

4. Analysis of the algorithm

This section provides the convergence results of the proposed algorithm and confirms the stability of the corresponding closed-loop system.

Theorem 1. *The state reward weight in* (3.15) *and positive definite cost* (3.14)*, as determined by Algorithm 1, converge to $\bar{Q}^*$ and P^*, respectively, resulting in K_0, as shown in* (3.11).

Proof. For a given stabilizing $\bar{Q}^{(i)} \geq 0$, the positive matrix $P^{(i)}$ obtained through (3.9) is the unique solution to the ARE. Based on (3.7), the following function is introduced:

$$\phi^{(i)}(s) = \frac{\partial E^{(i)}}{\partial P^{(i)}} = -(s^{(i)T}R^{-1}B_1^T + B_1R^{-1}s^{(i)}), \tag{4.1}$$

which is observed to be a symmetrical matrix. Subsequently, (3.14) can be reformulated as

$$P^{(i+1)} = P^{(i)} - \alpha\phi^{(i)}(s). \tag{4.2}$$

Substituting (4.2) in (3.15) yields

$$\begin{aligned}\bar{Q}^{(i+1)} = &-\left(A_1^T P^{(i)} + P^{(i)} A_1 - P^{(i)} B_1 R^{-1} B_1^T P^{(i)}\right) + \alpha\left(A_1^T \phi^{(i)}(s) + \phi^{(i)}(s) A_1\right) \\ &- \alpha\left(P^{(i)} B_1 R^{-1} B_1^T \phi^{(i)}(s) + \phi^{(i)}(s) B_1 R^{-1} B_1^T P^{(i)}\right) + \alpha^2 \phi^{(i)}(s) B_1 R^{-1} B_1^T \phi^{(i)}(s).\end{aligned} \tag{4.3}$$

By employing the gradient descent method to develop (3.9) for tuning $P^{(i)}$, $K^{(i)}$ is driven toward K_0. Consequently, the error diminishes with each iteration, that is for all i=0,1,2…,

$$E^{(i)} > E^{(i+1)} \geq 0, \tag{4.4}$$

from which we have $\lim_{i\to\infty} E^{(i)} = 0$, $\lim_{i\to\infty} s^{(i)} = 0$, and $\lim_{i\to\infty} \phi^{(i)}(s) = 0$. It follows from (4.2) that

$$\lim_{i\to\infty} P^{(i+1)} = \lim_{i\to\infty}\left(P^{(i)} - \alpha\phi^{(i)}(s)\right) = \lim_{i\to\infty} P^{(i)}. \tag{4.5}$$

Considering that K_0 is stabilizing, together with (3.13), it follows that

$$\lim_{i\to\infty} K^{(i)} = \lim_{i\to\infty} R^{-1} B_1^T P^{(i)} = K_0. \tag{4.6}$$

Taking the limit on both sides of (4.3) and using (4.2), considering (4.4) and (4.5), we get

$$\lim_{i\to\infty} \bar{Q}^{(i+1)} = -\lim_{i\to\infty}\left(A_1^T P^{(i)} + P^{(i)} A_1\right) + \lim_{i\to\infty} P^{(i)} B_1 R^{-1} B_1^T P^{(i)} = \lim_{i\to\infty} \bar{Q}^{(i)}. \tag{4.7}$$

Let $\lim_{i\to\infty} P^{(i)} = P^*$ and $\lim_{i\to\infty} \bar{Q}^{(i)} = \bar{Q}^*$. Thus, (4.7) and (4.6) become (3.10) and (3.11), respectively.

The proof is thus completed. □

Given Algorithm 1's convergence results, Theorem 2 is presented below to demonstrate the stability of the corresponding closed-loop system.

Theorem 2. *Given that $s^{(i)}, i = 0.1, \ldots$ in* (3.13) *are bounded, there exists a learning rate $\alpha > 0$ that ensures every $P^{(i)}$ and $\bar{Q}^{(i)}$ derived from Algorithm 1 globally asymptotically stabilize system (A_1, B_1) with control law* (2.16)*.*

Proof. Utilizing the gradient optimization method to tune $P^{(i)}$ in (3.14) will consequently drive $K^{(i)}$ toward K_0. Thus, the error diminishes with each iteration, meaning $E^{(i)} > E^{(i+1)} \geq 0$, for all i=0, 1, 2, …, leading to $\lim_{i\to\infty} E^{(i)} = 0$ and $\lim_{i\to\infty} s^{(i)} = 0$. From the monotonic boundedness principle, it can be known that $s^{(i)}$ in (3.13) is bounded. In Algorithm 1, by initializing a positive definite reward weight $\bar{Q}^{(i)} \geq 0$ and setting $i = 0$, the positive definite matrix $P^{(i)} > 0$ and the control gain $K^{(i)}$ can be uniquely determined by Eqs (3.9) and (3.12), respectively. Simultaneously, the stability of the system (A_1, B_1) should be ensured. Referring to [1, 27], it follows from System (2.9) that

$$\left(A_1 - B_1 K^{(i)}\right)^T P^{(i)} + P^{(i)}\left(A_1 - B_1 K^{(i)}\right) = -P^{(i)} B_1 R^{-1} B_1^T P^{(i)} - \bar{Q}^{(i)} < 0, \tag{4.8}$$

which implies that the control law $K^{(i)}$ stabilizes the system (A_1, B_1). Subsequently, $\bar{Q}^{(i+1)}$ is updated using (3.15) with $P^{(i+1)}$ from (3.14). We then demonstrate that $\bar{Q}^{(i+1)}, i \geq 0$ also stabilizes the system (A_1, B_1). Referring to (4.3), we define the terms involving $\phi^{(i)}(s)$ in (4.3) as

$$\Phi^{(i)}(s) = A_1^T \phi^{(i)}(s) + \phi^{(i)}(s) A_1 - P^{(i)} B_1 R^{-1} B_1^T \phi^{(i)}(s) - \phi^{(i)}(s) B_1 R^{-1} B_1^T P^{(i)} + \alpha\phi^{(i)}(s) B_1 R^{-1} B_1^T \phi^{(i)}(s). \tag{4.9}$$

Then, (4.4) can be reformulated as

$$\bar{Q}^{(i+1)} = \bar{Q}^{(i)} + \alpha\Phi^{(i)}(s). \tag{4.10}$$

Theorem 1 demonstrates that as i tends to infinity, the bounded term $s^{(i)} \to 0$ in (4.1), and consequently, $\phi^{(i)}(s) \to 0$ in (4.1) as $i \to \infty$. This demonstrates $\Phi^{(i)}(s) \to 0$ in (4.9) as $i \to \infty$, which are bounded. Given that both the control law $K^{(i)}$ and K_0 are stabilizing, and $P^{(i)}$ has a linear effect on $K^{(i)}$ in (3.12), selecting an appropriate $\alpha > 0$ in (4.2) to stabilize $P^{(i)}$ ensures that $K^{(i+1)}$ is also stabilizing. In other words, with the stabilization of $\bar{Q}^{(i)}$, there exists a learning rate enabling $\bar{Q}^{(i+1)}$ in (4.10) to yield $P^{(i+1)}$ from (3.9) and $K^{(i+1)}$, which provide the control input (3.12) to stabilize the controlled system (2.9) asymptotically.

The proof is thus completed. □

Remark 3. *In Algorithm 1, the value function matrices $P^{(i)}$ obtained at each iterative step are all positive definite. This is primarily due to two reasons. First, the initial $P^{(0)}$ is the unique positive definite solution of the ARE, establishing a foundation for positive definiteness. Second, an appropriate learning rate α ensures that the gradient descent update preserves the iterative process that maintains the positive definite property. This result is also the key to effectively achieve inverse optimal output tracking, ensuring that the control law of each iteration can stabilize the system and that the weight matrix ultimately converges to a final value.*

5. Simulative examples

The availability of the presented algorithm is verified through two practical engineering cases in this section.

5.1. Single-tank control system

The control system for a single tank can be expressed as follows [37]:

$$\dot{x} = -\frac{1}{ar}x + \frac{K}{a}u, y = x,$$

where u denotes the valve opening, K the flow coefficient, a the tank's cross-sectional area, r the hydraulic resistance, and x the liquid level height—serving as the system output y. Setting $a = 0.25$, $r = 2$, $K = 1$, the single-tank control system is

$$\dot{x} = -2x + 4u, y = x.$$

Set the desired trajectory as $\dot{y}_d = -3y_d, y_d(0) = 2$. It follows from (2.9) that

$$T = \begin{bmatrix} -2 & 0 \\ 0 & -3 \end{bmatrix}, B_1 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}.$$

Set $\gamma = 0.1$ and $R = 3$. Considering $A_1 = T - 0.5\gamma I$, $C_1 = \begin{bmatrix} C & -I \end{bmatrix}$, it yields

$$A_1 = \begin{bmatrix} -2.05 & 0 \\ 0 & -3.05 \end{bmatrix}, C_1 = \begin{bmatrix} 1 & -1 \end{bmatrix}.$$

Given the desired tracking control gain $K_0 = \begin{bmatrix} -0.2 & 1 \end{bmatrix}$, we select an initial weight matrix as $Q^{(0)}$=5. Therefore, $\bar{Q}^{(0)} = C_1^T Q^{(0)} C_1$. Solving (3.1) gives the initial $P^{(0)}$:

$$P^{(0)} = \begin{bmatrix} 0.6574 & -0.5810 \\ -0.5810 & 0.5245 \end{bmatrix}.$$

Set the initial value of liquid level height $x(0) = 2$ and the learning rate $\alpha = 0.1$. The converged state weight matrix and ARE solution, obtained via iterative calculation using Algorithm 1, are as follows:

$$\bar{Q}^* = \begin{bmatrix} 0.7351 & -4.4140 \\ -4.4140 & 6.1848 \end{bmatrix},$$

and

$$P^* = \begin{bmatrix} 0.1500 & -0.7481 \\ -0.7481 & 0.5245 \end{bmatrix}.$$

The optimal tracking control gain generated by Algorithm 1 is $K^* = \begin{bmatrix} -0.2000 & 0.9970 \end{bmatrix}$, which exhibits only a minor deviation from the given tracking control law K_0.

For the single-tank control system, the convergence of the state reward weight matrix $\bar{Q}^{(i)}$, the positive definite cost matrix $P^{(i)}$ in ARE, and control law $K^{(i)}$ via Algorithm 1 are shown in Figure 1. Figure 2 presents the optimal trajectory of the augmented system (A_1, B_1), where all state variables approach 0, indicating successful trajectory tracking in the single-tank control system. Figure 3 demonstrates that the state variable x closely tracks the reference trajectory y_d, confirming that the tank's water level consistently achieves the desired height. Note that optimal tracking is only achievable for stable reference signals, owing to the system's inherent stability. Figure 4 shows the evolution of the tracking control signal.

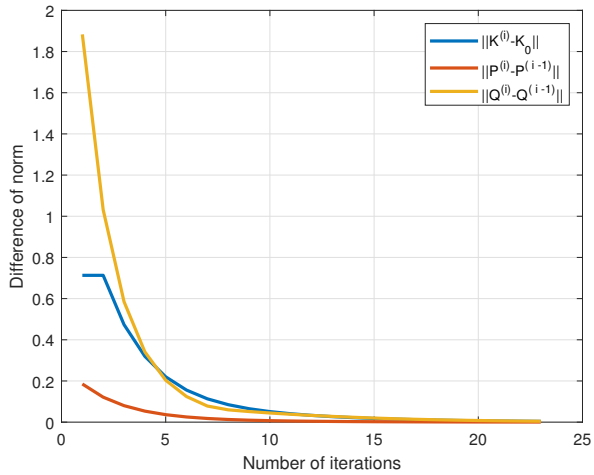

Figure 1. The convergence of control gain $K^{(i)}$, ARE solution $P^{(i)}$, and $\bar{Q}^{(i)}$.

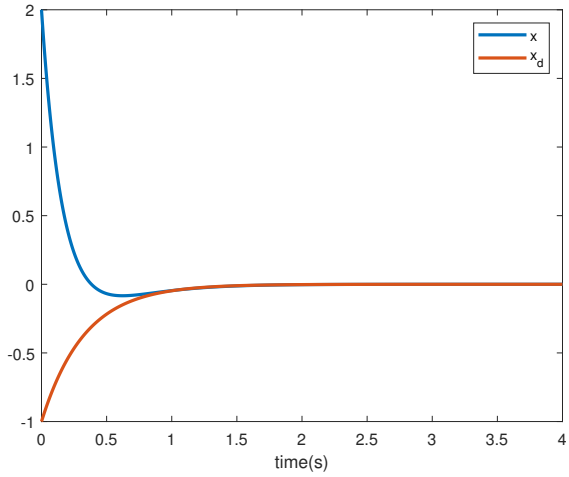

Figure 2. The optimal state trajectory.

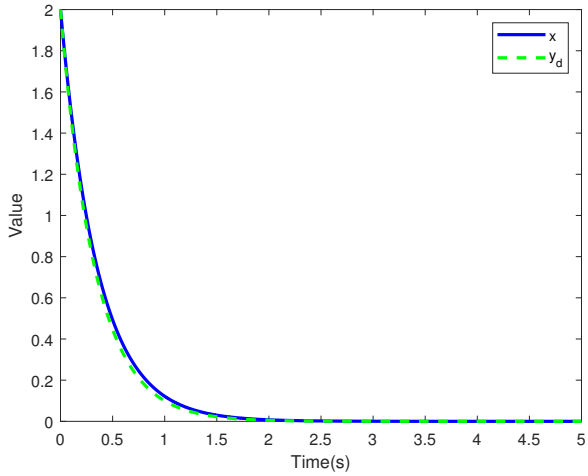

Figure 3. The single-tank control system output versus the reference trajectory.

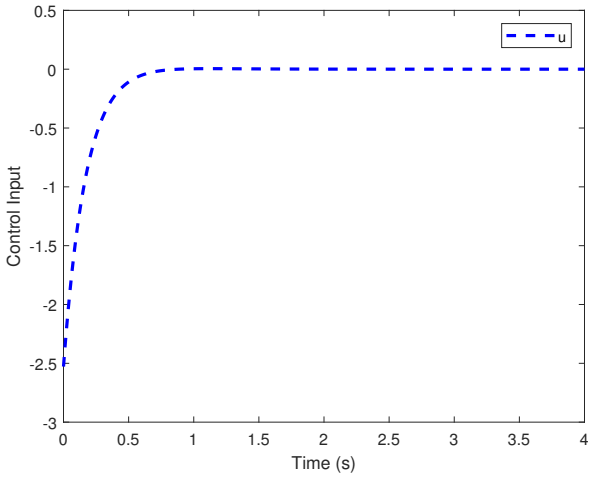

Figure 4. The tracking control signal.

5.2. Motor speed-position control system

The motor speed-position control system can be indicated as follows [38]:

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\left(\frac{K_t K_e}{JR_a} + \frac{b}{J}\right) \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_t}{JR_a} \end{bmatrix} u,$$

where the parameters K_t and K_e denote the torque constant and the back electromotive force (back-EMF) constant, respectively. The moment of inertia is denoted by J, the armature resistance by R_a, and the damping coefficient by b. The armature voltage is designated as input u. The angular position θ and and angular velocity ω are defined as system outputs, with the corresponding output matrix $C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. Setting $R_a = 1$, $K_t = 2$, $K_e = 1$, $J = 1$, $b = 2$, the motor speed-position control system is

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix} \begin{bmatrix} \theta \\ \omega \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u.$$

The desired reference trajectory is generated by the following dynamic system:

$$\dot{y}_d = \begin{bmatrix} -3 & 0 \\ 0 & -3 \end{bmatrix} y_d, y_d(0) = \begin{bmatrix} 3 \\ 4 \end{bmatrix}.$$

From (2.11), we have

$$T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -4 & 0 & 0 \\ 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}.$$

Set $\gamma = 0.1$ and $R = 3$. Considering $A_1 = T - 0.5\gamma I$ and $C_1 = \begin{bmatrix} C & -I \end{bmatrix}$, we obtain

$$A_1 = \begin{bmatrix} -0.05 & 1 & 0 & 0 \\ 0 & -4.05 & 0 & 0 \\ 0 & 0 & -3.05 & 0 \\ 0 & 0 & 0 & -3.05 \end{bmatrix}, C_1 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}.$$

Given the desired tracking control gain $K_0 = \begin{bmatrix} -18 & -2.5 & 0.5 & -1 \end{bmatrix}$ and selecting the initial weight matrix as $Q^{(0)} = I_{2\times 2}$, we compute $\bar{Q}^{(0)} = C_1^T Q^{(0)} C_1$. Solving (3.1) yields the initial solution:

$$P^{(0)} = \begin{bmatrix} 1.9124 & 0.6359 & -0.3020 & 0.0639 \\ 0.6359 & 0.4537 & -0.0503 & -0.1558 \\ -0.3020 & -0.0503 & 0.1631 & -0.0026 \\ 0.0639 & -0.1558 & -0.0026 & 0.1560 \end{bmatrix}.$$

With a learning rate $\alpha = 0.1$, the iteratively computed state weight matrix and ARE solution via Algorithm 1 converge to

$$\bar{Q}^* = \begin{bmatrix} 647.9921 & 125.8679 & -18.9307 & 36.1860 \\ 125.8679 & -13.2445 & -4.7473 & 10.0342 \\ -18.9307 & -4.7473 & 1.4948 & -1.0153 \\ 36.1860 & 10.0342 & -1.0153 & 2.9507 \end{bmatrix},$$

and

$$P^* = \begin{bmatrix} 1.9124 & 17.9972 & -0.3020 & 0.0639 \\ 17.9972 & 2.5000 & -0.4999 & 0.9998 \\ -0.3020 & -0.4999 & 0.1631 & -0.0026 \\ 0.0639 & 0.9998 & -0.0026 & 0.1560 \end{bmatrix}.$$

The optimal tracking control gain generated by Algorithm 1 is $K^* = \begin{bmatrix} -17.9969 & -2.5000 & 0.4999 & -0.9998 \end{bmatrix}$, which exhibits only a minor deviation from the given tracking control law K_0.

For the motor speed-position control system, the convergence of the state reward weight matrix $\bar{Q}^{(i)}$, the positive definite solution matrix $P^{(i)}$ in ARE, and the control law $K^{(i)}$ via Algorithm 1 is shown in Figure 5. Figure 6 presents the optimal trajectories of the augmented system (A_1, B_1), with all states approaching 0, indicating that trajectory tracking in the motor speed-position system has been successfully achieved. Figure 7 illustrates the optimal control tracking performance, showing the system asymptotically tracks reference y_d. It should be clarified that optimal tracking is only achievable for stable reference signals, owing to the motor speed-position system's inherent stability. Figure 8 shows the evolution of the tracking control law.

Remark 4. *Learning rate α governs the magnitude of the iterative update applied to $P^{(i)}$ in the direction that minimizes the gain error. This update is subsequently propagated to the augmented state weight matrix $\bar{Q}^{(i)}$ via the ARE, thereby shaping the convergence behavior of the gain matrix $K^{(i)}$ toward its steady-state value K_0. Empirical analysis indicates that a step size within the range $[0.1, 0.5]$ ensures convergence stability. Specifically, $\alpha = 0.1$ consistently yields optimal convergence performance across benchmark systems, including the single-tank control system and motor speed-position control system. The parameter α below 0.1 results in diminished convergence speed. Moreover, excessively large values of α may induce numerical instability or divergence.*

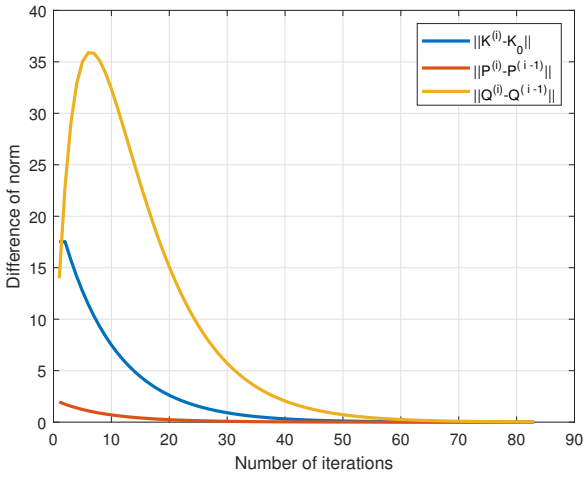

Figure 5. The convergence of control gain $K^{(i)}$, ARE solution $P^{(i)}$, and $\bar{Q}^{(i)}$.

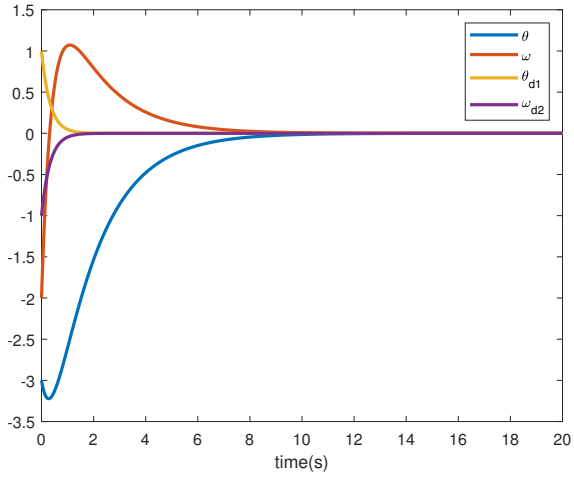

Figure 6. The optimal state trajectory.

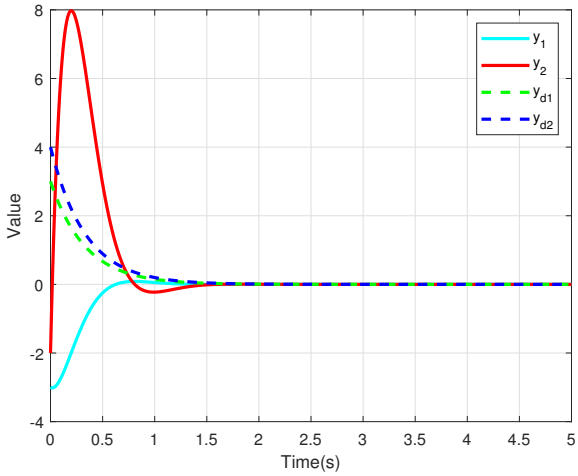

Figure 7. The motor speed-position control system output versus the reference trajectory.

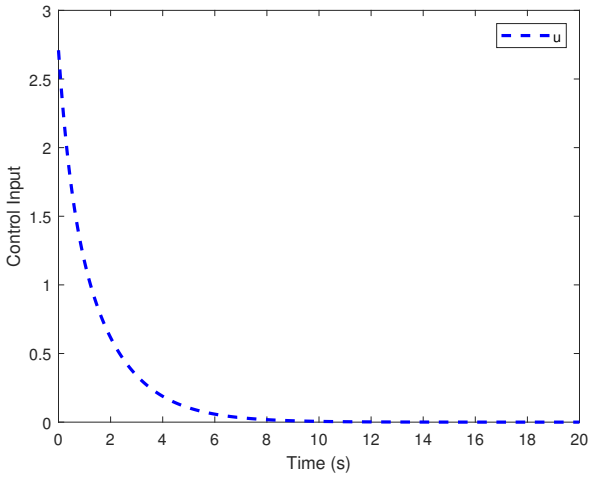

Figure 8. The tracking control signal.

6. Conclusions

The optimal output tracking problem is reframed as a regulator problem by expanding the state variables in this work. We then introduce a model-based inverse RL algorithm to address the objective function reconstruction challenge for continuous-time linear systems. The algorithm's convergence and the stability of the resulting closed-loop system are both validated. Simulation results confirm the effectiveness of the proposed algorithm. We plan to extend the current model-based inverse optimal tracking algorithm to scenarios where the model is unknown in future research.

Author contributions

Yi Mo: Conceptualization, writing-review & editing, software; Jingling Zhao: Writing-original draft, software, investigation; Kunyu Xiang: Software, formal analysis, investigation; Dengguo Xu: Conceptualization, methodology, writing-review & editing, supervision.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research is supported by the technology development fund project commissioned by Guangxi Multidimensional Cloud Space Network Security Technology Co., Ltd. Grant No. GYHX2026003.

Conflict of interest

The authors declare no conflicts of interest.

References

1. Y. Zhang, X. Yan, W. Zou, Z. Xiang, Fuzzy optimal tracking control for autonomous surface vehicles with prescribed-time convergence analysis, *IEEE Trans. Fuzzy Syst.*, **32** (2024), 6523–6533. https://doi.org/10.1109/TFUZZ.2024.3451493
2. W. Tan, R. Luo, Z. Peng, Q. Ling, Online adaptive optimal control algorithm based on weighted policy iteration, *IEEE Trans. Neur. Net. Lear.*, **36** (2025), 15723–15734. https://doi.org/10.1109/TNNLS.2025.3564329
3. L. Cui, B. Pang, M. Krstić, Z. Jiang, Learning-based adaptive optimal control of linear time-delay systems: a value iteration approach, *Automatica*, **171** (2025), 111944. https://doi.org/10.1016/j.automatica.2024.111944
4. D. Wang, J. Wang, D. Liu, J. Qiao, General multistep value iteration for optimal learning control, *Automatica*, **175** (2025), 112168. https://doi.org/10.1016/j.automatica.2025.112168
5. R. Bellman, R. E. Kalaba, An inverse problem in dynamic programming and automatic control, *J. Math. Anal. Appl.*, **7** (1963), 322-325. https://doi.org/10.1016/0022-247X(63)90056-8

6. R. E. Kalman, When is a linear control system optimal? *J. Basic Eng.*, **86** (1964), 51–60. https://doi.org/10.1115/1.3653115
7. T. L. Molloy, J. J. Ford, T. Perez, Finite-horizon inverse optimal control for discrete-time nonlinear systems, *Automatica*, **87** (2018), 442–446. https://doi.org/10.1016/j.automatica.2017.09.023
8. M. Krstíc, Z. Li, Inverse optimal design of input-to-state stabilizing nonlinear controllers, *IEEE Trans. Automat. Contr.*, **43** (1998), 336–350. https://doi.org/10.1109/9.661589
9. T. Rajpurohit, W. M. Haddad, Nonlinear-nonquadratic optimal and inverse optimal control for stochastic dynamical systems, *Int. J. Robust Nonlin.*, **27** (2017), 4723–4751.
10. N. Yokoyama, Inference of aircraft intent via inverse optimal control including second-order optimality condition, *J. Guid. Control Dynam.*, **41** (2018), 1–11. https://doi.org/10.2514/1.G002792
11. K. Lu, S. Han, J. Yang, H. Yu, Inverse optimal adaptive tracking control of robotic manipulators driven by compliant actuators, *IEEE Trans. Ind. Electron.*, **71** (2024), 6139–6149. https://doi.org/10.1109/TIE.2023.3296831
12. J. Zhao, D. Xu, X. Zhang, X. Li, Inverse optimal tracking control for AC/DC converter based on inverse reinforcement learning, *Int. J. Control Autom. Syst.*, **24** (2026), 320–331. https://doi.org/10.1007/s12555-026-00015-8
13. A. Y. Ng, S. Russell, Algorithms for inverse reinforcement learning, *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, 663–670.
14. P. Abbeel, A. Y. Ng, Apprenticeship learning via inverse reinforcement learning, *Proceedings of the twenty-first International Conference on Machine Learning*, 2004, 1. https://doi.org/10.1145/1015330.1015430
15. J. Zheng, S. Liu, L. M. Ni, Robust bayesian inverse reinforcement learning with sparse behavior noise, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, 2198–2205. https://doi.org/10.1609/aaai.v28i1.8979
16. L. Yu, J. Song, S. Ermon, Multi-agent adversarial inverse reinforcement learning, *Proceedings of the 36th International Conference on Machine Learning*, 2019, 7194–7201.
17. G. Kalweit, M. Huegle, M. Werling, J. Boedecker, Deep inverse Q-learning with constraints, *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, 14291–14302.
18. S. Choi, S. Kim, H. Jin Kim, Inverse reinforcement learning control for trajectory tracking of a multirotor UAV, *Int. J. Control Autom. Syst.*, **15** (2017), 1826–1834. https://doi.org/10.1007/s12555-015-0483-3
19. C. You, J. Lu, D. Filev, P. Tsiotras, Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning, *Robot. Auton. Syst.*, **114** (2019), 1–18. https://doi.org/10.1016/j.robot.2019.01.003
20. Q. Tang, H. Guo, Q. Chen, Multi-market bidding behavior analysis of energy storage system based on inverse reinforcement learning, *IEEE Trans. Power Syst.*, **37** (2022), 4819–4831. https://doi.org/10.1109/TPWRS.2022.3150518

21. J. Zhang, X. Wang, L. Li, G. Qu, L. Wan, Parameter recovery of neural network-based hammerstein system via immersion and invariance adaptive optimization scheme, *Expert Syst. Appl.*, **274** (2025), 127069. https://doi.org/10.1016/j.eswa.2025.127069

22. B. Lian, W. Xue, F. Lewis, T. Chai, Online inverse reinforcement learning for nonlinear systems with adversarial attacks, *Int. J. Robust Nonlin. Control*, **31** (2021), 6646–6667. https://doi.org/10.1002/rnc.5626

23. W. Xue, B. Lian, J. Fan, P. Kolaric, T. Chai, F. Lewis, Inverse reinforcement Q-learning through expert imitation for discrete-time systems, *IEEE Trans. Neur. Net. Lear.*, **34** (2023), 2386–2399. https://doi.org/10.1109/TNNLS.2021.3106635

24. Z. Sun, G. Jia, Inverse reinforcement learning by expert imitation for the stochastic linear-quadratic optimal control problem, *Neurocomputing*, **633** (2025), 129758. https://doi.org/10.1016/j.neucom.2025.129758

25. J. Huang, D. Xu, Y. Li, X. Zhang, J. Zhao, Inverse reinforcement learning for discrete-time linear systems based on inverse optimal control, *ISA Trans.*, **163** (2025), 108–119. https://doi.org/10.1016/j.isatra.2025.04.027

26. Z. Pang, S. Tang, J. Cheng, S. He, Scaling policy iteration based reinforcement learning for unknown discrete-time linear systems, *Automatica*, **176** (2025), 112227. https://doi.org/10.1016/j.automatica.2025.112227

27. H. Wu, Q. Hu, J. Zheng, F. Dong, Z. Ouyang, D. Li, Discounted inverse reinforcement learning for linear quadratic control, *IEEE Trans. Cybernetics*, **55** (2025), 1995–2007. https://doi.org/10.1109/TCYB.2025.3540967

28. M. Ghiyasi, L. Zhao, W. Zhu, Non-cooperative two-stage inverse DEA: a Stackelberg games approach for the efficiency analysis of China's regional economic development and people's living standards, *Ann. Oper. Res.*, **346** (2025), 2035–2063. https://doi.org/10.1007/s10479-025-06489-9

29. H. J. Asl, E. Uchibe, Inverse reinforcement learning methods for linear differential games, *Syst. Control Lett.*, **193** (2024), 105936. https://doi.org/10.1016/j.sysconle.2024.105936

30. E. Martirosyan, M. Cao, Reinforcement learning for inverse linear-quadratic dynamic non-cooperative games, *Syst. Control Lett.*, **191** (2024), 105883. https://doi.org/10.1016/j.sysconle.2024.105883

31. B. Lian, W. Xue, F. Lewis, A. Davoudi, Inverse value iteration and Q-learning: algorithms, stability, and robustness, *IEEE Trans. Neur. Net. Lear.*, **36** (2025), 6970–6980. https://doi.org/10.1109/TNNLS.2024.3409182

32. B. Lian, W. Xue, F. Lewis, A. Davoudi, Inverse Q-learning using input-output data, *IEEE Trans. Cybernetics*, **54** (2024), 728–738. https://doi.org/10.1109/TCYB.2023.3338197

33. B. Lian, W. Xue, Y. Xie, F. Lewis, A. Davoudi, Off policy inverse Q-learning for discrete-time antagonistic unknown systems, *Automatica*, **155** (2023), 111171. https://doi.org/10.1016/j.automatica.2023.111171

34. G. Zong, R. Liu, H. Xie, Y. Wang, X. Zhao, Observer-based adaptive NN tracking control for nonlinear NCSs under intermittent DoS attacks: a finite-time prescribed performance method, *IEEE Trans. Syst. Man Cy.*, **55** (2025), 2322–2331. https://doi.org/10.1109/TSMC.2024.3521025

35. H. Modares, F. L. Lewis, Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning, *IEEE Trans. Automat. Contr.*, **59** (2014), 3051–3056. https://doi.org/10.1109/TAC.2014.2317301

36. W. Xue, B. Lian, J. Fan, T. Chai, F. Lewis, Inverse reinforcement learning for trajectory imitation using static output feedback control, *IEEE Trans. Cybernetics*, **54** (2024), 1695–1707. https://doi.org/10.1109/TCYB.2023.3241015

37. E. Masero, G. Mussita, A. La Bella, R. Scattolini, A novel reinforcement learning-based approach for optimal control: an application to multi-tank water systems, *Eng. Appl. Artif. Intel.*, **162** (2025), 112407. https://doi.org/10.1016/j.engappai.2025.112407

38. G. Wang, D. Wang, H. Lin, J. Wang, X. Yi, A DC error suppression adaptive second-order backstepping observer for sensorless control of PMSM, *IEEE Trans. Power Electr.*, **39** (2024), 6664–6676. https://doi.org/10.1109/TPEL.2024.3367326

39. D. P. Bertsekas, *Nonlinear programming*, Belmont: Athena Scientific, 1995.

AIMS Press