Two Time Scale Partial Unknown Dynamics System Tracking Control Based On Off-policy Inverse Reinforcement Learning

MinYin Shen[1] and Fei Liu[1]

[1] Key Laboratory of Advanced Process Control for Light Industry
(Ministry of Education), Institute of Automation, Jiangnan University, Wuxi 214122, China
smy2010623@163.com, fliu@jiangnan.edu.cn

Abstract. This article, integrating the singular perturbation technique with inverse reinforcement learning, proposes a novel linear two time scale system tracking control method grounded in off-policy inverse reinforcement learning. This method addresses the challenge of unknown cost functions prevalent in industrial processes. First, the singular perturbation method is leveraged to decompose the original problem into fast and slow subsystem issues. Without manually designing a cost function, the method learns from known optimal behavioral data by reconstructing cost functions tailored to each subsystem, enabling the system to mimic optimal behaviors. Then, for the fast time scale system, a model-based inverse reinforcement learning method is adopted, while for the slow time scale system, a model-free off-policy inverse reinforcement learning strategy is employed, which reconstructs the system's cost function solely using measured expert behavioral data inputs. Finally, using a mixed separation thickening industrial process to illustrate the effectiveness of this method in two time scale tracking.

Keywords: Two Time Scale, Tracking Control, Inverse Reinforcement Learning.

1 Introduction

It is well known that the control loops and manipulated variables of industrial equipment have different operation speeds and interact with each other. The mixed separation thickening process is a typical industrial process with two different time scales, where exist two different time scales. The fast process is slurry pump, the unit device, while the underflow concentration of the slow process is the operation index [1]. In order to ensure the stability and reliability of the equipment, this interaction must be considered in the design process. Usually, the singular perturbation method [2] is used to deal with the interaction of system with different operation speeds. This method establishes a singular perturbation system model by introducing a smaller time scale to describe the difference between the fast and slow modes in the system, so as to accurately describe the two time scale dynamic behavior of the process. Due to the high complexity and mechanism uncertainty of industrial process, accurate modeling is a difficult problem. [1] combined reinforcement learning with the singular perturbation method to develop a data-driven Q-learning singular perturbation technology to control the output of the two time scale industrial process to track the operational index. However, this method does not take into account the influence of probing noise on the incentive system. For eliminating the influence of probing noise, off-policy reinforcement learning [3] is proposed. [4] used off-policy reinforcement learning with the singular perturbation method to achieve tracking control of two time scale industrial processes. However, the performance index, also known as cost function, of these methods need to be given in advance.

In complex industrial processes, it is subjective and difficult to set the cost function artificially. Therefore, [5] proposed an inverse reinforcement learning (IRL) method to reconstruct the unknown reward function by using a set of optimal behavior data demonstrated by experts, so as to imitate the optimal behavior [6]. However, IRL cannot guarantee the stability of the learning process [7]. In addition, the inverse optimal control (IOC) proposed by Kalman [8] has a similar idea to IRL, but IOC is Model-based [9, 10] and requires that the demonstration data used be stable points rather than optimal points. In order to ensure the stability of IRL, [7] solves the IOC problem as a sub-problem of IRL. On this basis, [11-15] have continued to study, of which [11, 14] use off-policy IRL. However, these methods all operate on single-scale systems.

Inspired by [1, 7, 11, 14], this paper aims to solve the tracking control problem of two time scale industrial process by reconstructing the cost function using IRL without complete knowledge of cost function knowledge, and to imitate optimal behavior data. The main contributions are as follows:

1.The singular perturbation method is combined with IRL to reconstruct a cost function for the tracking control problem of two time scale industrial processes. By using the reconstructed cost function, a near-optimal controller is learned to imitate the known optimal behavior data.

2.Unlike existing literature which use reinforcement learning, this paper develops an IRL method for two time scale tracking control, which does not require artificially designing a cost function for the two time scale industrial processes.

3.A decentralized composite IRL control scheme is designed, a model-based IRL is designed for the fast time scale system, and the system cost function is reconstructed using system model. A model-free off-policy IRL is designed for the slow time scale system, and the cost function is reconstructed using only behavior data.

In Section 2, the objective of the two time scale IRL tracking control problem is formulated. In Section 3, a decentralized composite IRL control scheme is proposed to design the two time scale partially unknown systems. The scheme consists of a model-based IRL design for the fast time scale system and a model-free off-policy IRL design for the slow time scale system. Simulation results are shown in Section 4.

2 Problem Formulation

Consider the following singularly perturbed system model of a two time scale industrial process:

$$\varepsilon\dot{y}(t) = A_1\varepsilon y(t) + B_1u^*(t) \tag{1}$$

$$\begin{cases} \dot{x}(t) = A_2x(t) + B_2\varepsilon y(t) \\ \quad r(t) = Cx(t) \end{cases} \tag{2}$$

where (1) is the fast process of the control loop of the device, and (2) is the slow process of the operation process. $\varepsilon y(t) \in \mathbb{R}^{ny}$ is the state vector of the control loop, the time scale constant $\varepsilon \in (0,1)$ is known. $u^*(t) \in \mathbb{R}^{nu}$ is the optimal control input. $x(t) \in \mathbb{R}^{nx}$ is the state vector of the operation process. $r(t) \in \mathbb{R}^{nr}$ is the operation index. A_1, A_2, B_1, $B_2$and C are matrices of appropriate dimensions. Because there is a fast and slow coupling between (1) and (2), a time scale decomposition is performed on (1) and (2), let $u^*(t) = \tilde{u}^*(t) + \bar{u}^*(t)$, $y(t) = \bar{y}(t) + \tilde{y}(t)$, where $\bar{u}(t)$ and $\bar{y}(t)$ represent the slow components of $u^*(t)$ and $y(t)$, respectively, $\tilde{u}^*(t)$ and $\tilde{y}(t)$ represent their fast components.

Note that the dynamical models of device, typically actuator, is straightforward to identify in practice. Nevertheless, the operational process frequently encompasses intricate reactions, characterized by unknown mechanisms and uncertainties. Consequently, the precise model for the operational process is hardly to establish[16]. So there have following assumption.

Assumption 1: A_1, B_1 are known. And A_1 is a invertible matrix. A_2, B_2 are unknown. (A_1, B_1), (A_2, B_2) is controllable, and (A_2, C) is observable.

When $\varepsilon = 0$

$$\bar{y}(t) =- (A_1\varepsilon)^{-1}B_1\bar{u}^*(t)) \tag{3}$$

Remark 1: According to [17], when the fast and slow subsystems are stable, $y(t) = \bar{y}(t) + \tilde{y}(t) + 0(\varepsilon)$, $x(t) = \bar{x}(t) + 0(\varepsilon)$, $r(t) = \bar{r}(t) + 0(\varepsilon)$. Where $\bar{x}(t)$ is the slow component of $x(t)$, $\bar{r}(t)$ is the slow component of $r(t)$.

Using the singular perturbation method like [1], (1) and (2) are discretized and decomposed into fast subsystem (4) and slow subsystem (5):

$$\tilde{y}(k+1) = M_f\tilde{y}(k) + N_f\tilde{u}^*(k) \tag{4}$$

$$\begin{cases} \bar{x}(k) = M_s\bar{x}(k) + N_s\bar{u}^*(k) \\ \quad \bar{r}(k) = C\bar{x}(k) \end{cases} \tag{5}$$

where $M_f = e^{A_1\varepsilon T}$, $N_f = \int_0^T e^{A_1\varepsilon T}B_1dt$, $M_s = e^{A_2T}$, $N_s =- \int_0^T e^{A_2T}B_2{A_1}^{-1}B_1dt$. The sampling period of the fast process (1) is $\Delta t_f = \varepsilon T$, the sampling period of the slow process (2) is $\Delta t_s = T$.

Define the desired operation index trajectory as

$$r^*(k+1) = Fr^*(k) \tag{6}$$

where F is a constant matrix with appropriate dimensions.

For the original global system (1) and (2), the performance index is defined as

$$J = \min_{u(i)} \sum_{i=k}^{\infty} \gamma[(r(i) - r^*(i))^T\, Q_1^*(r(i) - r^*(i))$$

$$+ (y(i) - \bar{y}(i))^T Q_2^*(y(i) - \bar{y}(i)) + u^*(i)^T R^* u^*(i)] \qquad (7)$$

where the discount factor $\gamma \in (0,1)$ reflects the rate at which the system performance decays over time. Due to the presence of the discount factor, the matrix F is not assumed to be Hurwitz, which means that the reference trajectory does not have to tend to zero. Without discount factor, the cost function may be unbounded when the reference trajectory does not tend to zero. Q_1^*, Q_2^* and R^* are positive definite matrices. $y(i) - \bar{y}(i)$ is used to represent the high-frequency transients of the device control loop.

Assumption 2: In (7), (Q_1^*, Q_2^*, R^*) are unknown, but is full rank. And the optimal behavior data$(x(k), u^*(k), \varepsilon y(k))$ are known.

Problem 1: Considering Assumption 1 and Assumption 2, for any $R \in \mathbb{R}^{m\times m} > 0$, reconstruct the performance index of the two time scale systems so that the system behavior data imitates the optimal behavior data.

The performance index of the original global system can be decomposed into the performance indices (8) and (9) of the slow and fast subsystems, and their equivalence can be found in [1].

$$\sum_{i=k}^{\infty} \gamma[\tilde{y}(i)^T\, Q_2^* \tilde{y}(i) + \tilde{u}^*(i)^T R^* \tilde{u}^*(i)] \qquad (8)$$

$$\sum_{i=k}^{\infty} \gamma[(\bar{r}(i) - r^*(i))^T\, Q_1^*(\bar{r}(i) - r^*(i)) + \bar{u}^*(i)^T R^* \bar{u}^*(i)] \qquad (9)$$

Rewrite slow subsystem (5) into an augmented form

$$\bar{X}(k+1) = M\bar{X}(k) + N\bar{u}^*(k) \qquad (10)$$

where $\bar{X}(k) = \begin{bmatrix} \bar{x}(k) \\ r^*(k) \end{bmatrix}$, $M = \begin{bmatrix} M_s & 0 \\ 0 & F \end{bmatrix}$, $N = \begin{bmatrix} N_s \\ 0 \end{bmatrix}$. Substitute (10) into (9), the performance index of the slow subsystem is

$$\sum_{i=k}^{\infty} \gamma[\bar{X}(k)^T\, Q^* \bar{X}(k) + \bar{u}^*(i)^T R^* \bar{u}^*(i)] \qquad (11)$$

where $Q^* = [C \quad -I]^T Q_1^* [C \quad -I]^T$. According to the necessary conditions for optimal control, the optimal control $u^*(k) = \tilde{u}^*(k) + \bar{u}^*(k) =- K_f^* \tilde{y}(k) - K_s^* \bar{X}(k)$

$$K_f^* = \gamma(R^* + \gamma N_f{}^T P_f^* N_f)^{-1} N_f{}^T P_f^* M_f \qquad (12)$$

$$K_s^* = \gamma(R^* + \gamma N^T P_s^* N)^{-1} N^T P_s^* M \qquad (13)$$

P_f^* and P_s^* satisfy the following algebraic Riccati equation

$$P_f^* = \gamma M_f P_f^* M_f + Q_2^* - \gamma^2 M_f{}^T P_f^* N_f (R^* + \gamma N_f{}^T P_f^* N_f) N_f{}^T P_f^* M_f \qquad (14)$$

$$P_s^* = \gamma M^T P_s^* M + Q^* - \gamma^2 M^T P_s^* N (R^* + \gamma N^T P_s^* N) N^T P_s^* M \qquad (15)$$

Define 1: In (8) (Q_2^*, R^*) can obtain the optimal K_s^*, and P_f^* satisfies (14). Given any $R \in \mathbb{R}^{m\times m} > 0$, if there exists a $Q_f \in \mathbb{R}^{n\times n} \geq 0$ such that (Q_f, R) makes the corresponding unique stable $P_f \in \mathbb{R}^{n\times n} \geq 0$ produce the same K_f^*, then Q_f is called the equivalent weight of Q_2^*, and similarly, P_f is equivalent weight of P_f^*. In this way, the reconstruction cost function (8) is simplified to find an equivalent weight Q_f for Q_2^*.So the reconstruction cost function (9) is simplified to find an equivalent weight Q_s for Q^*.

Therefore, Problem 1 is transformed into solving Problem 2 for the fast subsystem (4) and solving Problem 3 for the slow subsystem (5).

Problem 2: Considering Assumption 1 and 2, using the system dynamics knowledge and the optimal gain K_f^*, for any $R \in \mathbb{R}^{m\times m} > 0$, find an equivalent weight Q_f for the performance index (8) such that the fast system imitates the optimal behavior data.

Problem 3: Considering Assumption 1 and 2, using only the optimal behavior data, for any $R \in \mathbb{R}^{m\times m} > 0$, find an equivalent weight Q_s for the performance index (9) such that the slow system imitates the optimal behavior data Q_s.

3 IRL Algorithm For Tow Time Scale system Tracking Control

To solve problems 2 and 3, Section 3.1 first proposes a model-based IRL algorithm for the fast time scale system to find the equivalent weight Q_f according to the known K_u^* when the model is known. Then, Section 3.2 proposes an off-policy IRL algorithm for the slow time scale system, which only uses the known behavior data ($x(k)$, $u^*(k)$, $\varepsilon y(k)$) to find the equivalent weight Q_s.

3.1 Model-based IRL Algorithm Fast Time Scale System

This subsection proposes a model-based IRL algorithm for the fast time scale system to solve problem 2.

```
Algorithm 1: Model-based IRL Iteration Algorithm
Step 1: Initialization. Let $i = 0$, where i is the iteration step length. For any R, $Q_f^0 = 0 \in \mathbb{R}^{n\times n}$. Given the initial gain $K_f^0$, $\alpha \in [0,1]$ tuning parameter.
Step 2: Policy evaluation. Evaluate $P_f^i$ by solving (16)
    $P_f^i = \gamma(M_f - N_f K_f^i)^T P_f^i (M_f - N_f K_f^i) + Q_f^i + {K_f^i}^T R K_f^i + \alpha(K_f^i - K_f^*)^T R (K_f^i - K_f^*)$   (16)
Step 3:Inverse optimal control. Update $Q_f^{i+1}$ by (17)
    $Q_f^{i+1} = P_f^i - \gamma(M_f - N_f K_u^i)^T P_f^i (M_f - N_f K_u^i) - {K_f^i}^T R K_f^i$   (17)
Step 4: Policy improvement. Update the gains $K_f^i$ according to (18)
    $K_f^i = -\gamma(R + \gamma {N_f}^T P_f^i N_f)^{-1} {N_f}^T P_f^i M_f$   (18)
Step 5: $\tilde{u}^i(k) = K_f^i \tilde{y}(k)$. Set $i = i + 1$, and go to Step 2. Stop when for any small normal number $\sigma_1$, $||K_f^* - K_f^i|| < \sigma_1$.
```

Algorithm 1 finds the equivalent weight Q_f using system dynamics (M_f, N_f) and optimal control gain K_f^*. Since (14) has unknown parameters P_f^* and Q_2^*, it is difficult to solve directly, so the policy iteration method is used to solve it. In each iteration i,

$i = 0,1\ldots$, P_f^i is calculated. $\alpha \in [0,1]$ is a tuning parameter to ensure the convergence of (16). Q_f^{i+1} is obtained by inverse optimal control (17) and K_f^i is obtained by optimal control (18) using P calculated by (16).

Remark 2 : [6, 11, 18] It has been proved that Algorithm 1 terminates after a finite number of iterations and can obtain a unique equivalent weight Q_f of Q_2^*. P_f^i converges to P_f^*, K_f^i converges to K_f^*. K_f^i can stabilize the system (4). This paper only adds a discount factor γ on its basis, so $\tilde{u}^i(k)$ converges to $\tilde{u}^*(k)$.

As mentioned in [11, 12, 18-19], the IOC problem usually has non-unique solutions, and Corollary 1 shows the non-uniqueness of Algorithm 1.

Lemma 1 (Non-uniqueness): Suppose Algorithm 1 converges to Q_{f1}, Q_{f1} satisfying

$$Q_{f1} = P_1 - \gamma {M_f}^T P_1 M_f + \gamma {M_f}^T P_1 N_f K_f^* \tag{19}$$

where P_1 satisfying

$$\gamma {N_f}^T P_f^* M_f = (R^* + \gamma {N_f}^T P_f^* N_f) K_f^* \tag{20}$$

Proof: The proof is similar to Theorem 4 in [13], except that there is a discount factor here.

Algorithm 1 requires complete knowledge of system dynamics to solve. Section 3.2 designs an off-policy IRL algorithm that does not require knowledge of system dynamics for the slow time scale system, as shown in Algorithm 2.

3.2 Model-free Off-policy IRL Algorithm For Slow Subsystem

Algorithm 1 requires complete knowledge of system dynamics to solve (16-18). In this subsection, an off-policy IRL Algorithm 2 is proposed for the slow time scale system without requiring knowledge of system dynamics, which finds the equivalent weight Q_s only using the known behavior data $(x(k), u^*(k), y(k))$ to solve problem 3.

Since $\bar{x}(k)$ is unmeasurable, if $M - NK_s^i$ is Hurwitz, there exists $\varepsilon^* \geq 0$ such that for any $\varepsilon \in (0, \varepsilon^*)$, $X(k) = \bar{X}(k) + 0(\varepsilon)$ holds for $k \geq 0$ [1]. There is the following approximation relationship: $\bar{x}(k)$ approximates $x(k)$, so $X(k) = \begin{bmatrix} x(k) \\ r^*(k) \end{bmatrix}$ can be used to replace $\bar{X}(k) = \begin{bmatrix} \bar{x}(k) \\ r^*(k) \end{bmatrix}$.

Substituting the slow time scale dynamics (M, N) and (13) into (16), multiplying both sides by $X(k)^T$ and $X(k)$, and replacing with the model-free data-driven equation, get

$$\begin{aligned} X(k)^T P_s^i X(k) &= \gamma X(k)^T (M - NK_s^i)^T P_s^i (M - NK_s^i) X(k) \\ &+ X(k)^T (Q_s^i + {K_s^i}^T R K_s^i + \alpha (K_s^i - K_s^*)^T R (K_s^i - K_s^*)) X(k) \end{aligned} \tag{21}$$

Rewrite the augmented system equation (10) as

$$X(k+1) = M^i X(k) + N(\bar{u}^*(k) - K_s^i X(k)) \tag{22}$$

where $M^i = M + NK_s^i$, $\bar{u}^i(k) = K_s^i X(k)$ is the target policy being updated. $\bar{u}^*(k)$ is the behavior policy actually applied to the system. $\bar{u}^*(k)$ cannot be directly measured, but since $\bar{u}^*(k) = u^*(k) - \tilde{u}^*(k) = u^*(k) + K_f^*\tilde{y}(k)$, substituting (3) and $\tilde{y}(k) = \varepsilon^{-1}\varepsilon y(k) - \bar{y}(k)$ into $\bar{u}^*(k)$ yields

$$\bar{u}^*(k) = (1 - K_f^*(A_1\varepsilon)^{-1}B_1)^{-1}(u^*(k) + \varepsilon^{-1}K_f^*\varepsilon y(k)) \tag{23}$$

Since $u^*(k)$, $y(k)$, K_f^*, A_1, ε and B_1 are known, $\bar{u}^*(k)$ can be obtained by combining measurable variables. (21) can be rewritten as

$$\begin{aligned} X(k)^T P_s^i X(k) - \gamma X(k)^T {M^i}^T P_s^i M^i X(k) = X(k)^T P_s^i X(k) \\ -\gamma(X(k+1) + N(\bar{u}^*(k) - \bar{u}^i(k)))^T P_s^i (X(k+1) + N(\bar{u}^*(k) - \bar{u}^i(k))) \\ = X(k)^T(Q_s^i + {K_u^i}^T R K_u^i)X(k) + \alpha(\bar{u}^*(k) - \bar{u}^i(k))^T R(\bar{u}^*(k) - \bar{u}^i(k)) \end{aligned} \tag{24}$$

Expanding (24) and writing it in the form of Kronecker product, get

$$\begin{aligned} (X(k)^T \otimes X(k)^T - \gamma X(k+1)^T \otimes X(k+1)^T)\mathrm{vech}(P_s^i) \\ +2(u^*(k) - u^i(k))^T \otimes X(k)^T \mathrm{vech}(\gamma N^T P_s^i M) \\ + (u^*(k) - u^i(k))^T \otimes (u^*(k) + u^i(k))^T \mathrm{vech}(\gamma N^T P_s^i N) \\ = X(k)^T(Q_s^i + {K_u^i}^T R K_u^i)X(k) + \alpha(\bar{u}^*(k) - \bar{u}^i(k))^T R(\bar{u}^*(k) - \bar{u}^i(k)) \end{aligned} \tag{25}$$

Set
$\Gamma^i = [\Gamma_1^i \quad \Gamma_2^i \quad \Gamma_3^i]^T$, $\Gamma_1^i = \mathrm{vech}(P_s^i)^T$, $\Gamma_2^i = \mathrm{vech}(\gamma N^T P_s^i M)^T$, $\Gamma_3^i = \mathrm{vech}(\gamma N^T P_s^i N)$,
$\psi^i = \begin{bmatrix} \psi_1^1 & \psi_2^1 & \psi_3^1 \\ \vdots & \ddots & \vdots \\ \psi_1^{N_1} & \psi_2^{N_1} & \psi_3^{N_1} \end{bmatrix}$, $\psi_1^{N_1} = (X(k)^T \otimes X(k)^T - \gamma X(k+1)^T \otimes X(k+1)^T)$,
$\psi_2^{N_1} = 2(\bar{u}^*(k) - \bar{u}^i(k))^T \otimes X(k)^T$, $\psi_3^{N_1} = (\bar{u}^*(k) - \bar{u}^i(k))^T \otimes (\bar{u}^*(k) + \bar{u}^i(k))^T$,
$\rho^i = [\rho_1^i \cdots\cdots \rho_N^i]^T$, $\rho_N^i = X(k+N-1)^T Q_s^i X(k+N-1) + \bar{u}^i(k+N-1)^T R \bar{u}^i(k+N-1) + \alpha(\bar{u}^i(k+N-1) - \bar{u}^*(k+N-1))^T R(\bar{u}^i(k+N-1) - \bar{u}^*(k+N-1))$.
Therefore, the Bellman equation of off-policy IRL can be written as

$$\psi^i \Gamma^i = \begin{bmatrix} \rho_1^i & \cdots & \rho_{N_1}^i \end{bmatrix} \tag{26}$$

The control gain K_s^i can be designed as

$$K_s^i =-(R + \Gamma_3^i)^{-1}\,\Gamma_2^i \tag{27}$$

Since Γ^i is a symmetric matrix with $n = (nx + nr)(nx + nr + 1)/2 + nu(nu + 1)/2 + (nx + nr)nu$ independent elements. Therefore, at least $N \geq n$ data sets are required to use the least square method to solve (26). The solution of (22) is $\Gamma^i = ({\psi^i}^T\psi^i)^{-1}{\psi^i}^T\begin{bmatrix} \rho_1^i & \cdots & \rho_{N_1}^i \end{bmatrix}$. In order to ensure ${\psi^i}^T\psi^i$ invertibility, the data set must satisfy the persistent excitation condition (PE). Add exploration noise $e_1(k)$ to the control input, let $\bar{u}^i(k) + e_1(k)$ replace $\bar{u}^i(k)$ in (26).

Substitute (13), (16), and slow time scale dynamics (M,N) into (17), multiply both sides by and , and replace with the model-free data-driven equation

$$X(k)^T Q_s^{i+1} X(k) = X(k)^T Q_s^i X(k) + \alpha(\bar{u}^*(k) - \bar{u}^i(k))^T R(\bar{u}^*(k) - \bar{u}^i(k))) \quad (28)$$

where $\theta^i = [\theta_1^i \cdots\cdots \theta_q^i]^T$, $\vartheta^i = [\vartheta_1^i \cdots\cdots \vartheta_N^i]^T$, $\theta_N^i = X(k+N-1)^T \otimes X(k+N-1)^T$, $\vartheta^i = X(k)^T(Q_s^i + \alpha(K_u^i - K_u^*)^T R(K_u^i - K_u^*))X(k)$. Since Q_s^i is a symmetric matrix with $n_1 = (nx + nr)(nx + nr + 1)/2$ independent elements. Therefore, at least $N_1 \geq n_1$ data sets are required to use the least square method to solve (29). Since $n_1 < n$, the whole algorithm needs to collect $N \geq n$ data sets.

So far, the model-free off-policy IRL algorithm for the slow time scale system is shown in Algorithm 2.

```
Algorithm 2: Model-free Off-policy IRL Algorithm
Step 1: Initialization. Let $i = 0$, where i is the iteration step length. For any R, $Q_s^0 = 0 \in \mathbb{R}^{n \times n}$. $\alpha \in [0,1]$ tuning parameter.
Step 2:Data collection. Collect $N \geq n$ groups data $\{X(k)\}$ $\{u^*(k)\}$ $\{y(k)\}$ and add the detection noise $e_1(k)$ to form $\psi^i$ and $[\rho_1^i \quad \cdots \quad \rho_{N_1}^i]$ in (26).
Step 3: Policy evaluation. Evaluate $P_s^i$ by solving (26)
Step 4:Inverse optimal control. Update $Q_s^{i+1}$ by (28)
Step 5: Policy improvement. Update the gains $K_s^i$ according to (27)
Step 6: $\bar{u}^i(k) = K_s^i X(k)$. Set $i = i + 1$, and go to Step 2. Stop when for any small normal number $\sigma_1$, $||K_s^* - K_s^i|| < \sigma_1$.
```

Theorem 1 (Convergence): Algorithm 2 can obtain the unique equivalent weight Q_s, K_s^i converges to the optimal solution K_s^* given in (13), and therefore $\bar{u}^i(k)$ converges to $\bar{u}^*(k)$.

Proof: Since (28) is obtained by substituting (16) into (17) and multiplying both sides by $X(k)^T$ and $X(k)$. Because of the existence of the tracking desired operation index error, $X(k) \neq 0$, (28) is equivalent to (17) . Therefore, the inverse optimal control step of Algorithm 1 is equivalent to that of Algorithm 2. [20] proved that the remaining parts of Algorithm 1 are equivalent to the remaining parts of Algorithm 2. Therefore, Algorithm 2 is equivalent to Algorithm 1. So, Algorithm 2 can obtain the unique equivalent weight Q_s, K_s^i converges to the optimal solution K_s^* given in (13), and therefore $\bar{u}^i(k)$ converges to $\bar{u}^*(k)$.

Lemma 2 (Stability): Algorithm 2 yields control inputs that stabilize the closed-loop system.

Proof: Since Algorithm 2 is equivalent to Algorithm 1, and Algorithm 1's control strategy stabilizes the system, it follows that the control inputs obtained from Algorithm 2 also stabilize the system.

Lemma 3 (Unbiasedness): The exploration noise will not cause bias in the estimation of Γ^i, Q_s^{i+1}, K_s^i in algorithm 2.

Proof: [11] proved that the off-policy IRL algorithm can eliminate the influence of exploration noise, so the estimation of Γ^i, Q_s^{i+1}, K_s^i will not produce bias. Here, only discount factor is added.

3.3 Composite Control and Performance Analysis

Based on the singular perturbation optimal control [21], the decentralized composite control input can be constructed as

$$u^i(k) = \bar{u}^i(k) + \tilde{u}^i(k) = K_s^i\bar{X}(k) + K_f^i\tilde{y}(k) \tag{30}$$

In order to implement the controller (30) by measured data directly from global system, get

$$\begin{aligned} u^i(k) &= K_s^iX(k) + K_f^i(y(k) - (A_1\varepsilon)^{-1}B_1K_s^iX(k)) \\ &= (K_s^i - (A_1\varepsilon)^{-1}B_1K_f^iK_s^i)X(k) + \varepsilon^{-1}K_f^i\varepsilon y(k) \end{aligned} \tag{31}$$

Lemma 4: The control strategy $u^i(k)$ obtained by the algorithm converges to the optimal optimal control strategy $u^*(k)$, that is, $\lim_{i\to\infty} u^i(k) = u^*(k)$. In addition, the original global system (1), (2) is asymptotically stable under $\lim_{i\to\infty} u^i(k)$.

Proof: Lemma 2 in [1] proved that when $\tilde{u}^i(k)$ converges to $\tilde{u}^*(k)$ and $\bar{u}^i(k)$ converges to $\bar{u}^*(k)$, $u^i(k)$ converges to $u^*(k)$. The optimal strategy $u^*(k)$ can make the original global system (1), (2) asymptotically stable, so in the original global system (1), (2) is asymptotically stable under $\lim_{i\to\infty} u^i(k)$.

Lemma 5: The control strategy $u^i(k)$ is a $0(\varepsilon)$ approximate optimal solution to the problem 1, the tracking error $r(k) - r^*(k)$ and $y(k) - \bar{y}(k)$ are stable.

Proof:According to [22]

$$\|K_s^* - K_s^i\| \le \|0(\varepsilon)\| \le d\varepsilon \tag{32}$$

Then

$$\begin{aligned} \left\|u^*(k) - \lim_{i\to\infty} u^i(k)\right\| &= \|-(K_s^i - {A_1}^{-1}B_1K_s^*)X(k) \\ &-(K_s^i - {A_1}^{-1}B_1K_s^i)\bar{X}(k)\| \le e\varepsilon \end{aligned} \tag{33}$$

where $e = d\left\|I - {A_1}^{-1}B_1\right\|(\|\bar{X}(k)\| + \|K_s^* - K_s^i\|)$, therefore, $u^*(k) = \lim_{i\to\infty} u^i(k) + 0(\varepsilon)$ can be obtained. According to Lemma 3 in [1], the tracking error $r(k) - r^*(k) = \bar{r}(k) - r^*(k) + 0(\varepsilon)$ and $y(k) - \bar{y}(k) = 0(\varepsilon)$ are convergent.

4 Simulation Results

For a typical industrial process object, the mixed separation thickening process is described in detail [1]. Taking the linearized model given in [1] as an example, the effectiveness of the proposed method is verified.

$$\begin{cases} \varepsilon\dot{y}(t) =- 0.68\varepsilon y(t) + 2.6u(t) \\ \dot{r}(t) =- 0.057r(t) + 0.055\varepsilon y(t) \\ \qquad r(t) = x(t) \end{cases} \tag{34}$$

In order to obtain satisfactory concentrate grade and tail grade, the underflow concentration $r(t)$, underflow slurry flow rate $\varepsilon y(t)$, and its high-frequency transients $\varepsilon(y - \bar{y})$ are controlled within the target range. The time scale constant $\varepsilon = 0.1$. The desired underflow concentration value should be set by $r^*(k) = 33$. Select parameters

Q_1^*=100, $R^* = 0.1$, Q_2^*=2.5, $T = 0.1$, $Q^* = \begin{bmatrix} 100 & -100 \\ -100 & 100 \end{bmatrix}$. The optimal optimal fast control strategy gain is $K_f^* = 2.6150$. The optimal optimal slow control strategy gain is $K_s^* = [21.0018 \quad -21.2432]$, initial value $r(0) = 32.2$, $y(0) = 1$, generate optimal behavior data ($r(k)$, $u^*(k)$, $y(k)$).

In the case of unknown parameters (Q_1^*, Q_2^*, R^*), unknown system dynamics in (34), and unknown K_s^*, measurement noise $e_1(k) \in [-0.01\ 0.01]$, $\alpha = 0.7$, $\sigma_1 = 0.02$, $\sigma_2 = 0.1$, Q_f^0 =0, $Q_s^0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, choosing $R = 0.1$. According to the optimal behavior data ($r(k)$, $u^*(k)$, $y(k)$), use inverse reinforcement learning to find equivalent weights, get $Q_f = 2.5274$, $Q_s = \begin{bmatrix} 108.4668 & -108.5642 \\ -108.5642 & 108.6620 \end{bmatrix}$, the corresponding learned control gains are $K_f = 2.4575$, $K_s = [20.9904 \quad -21.2318]$.

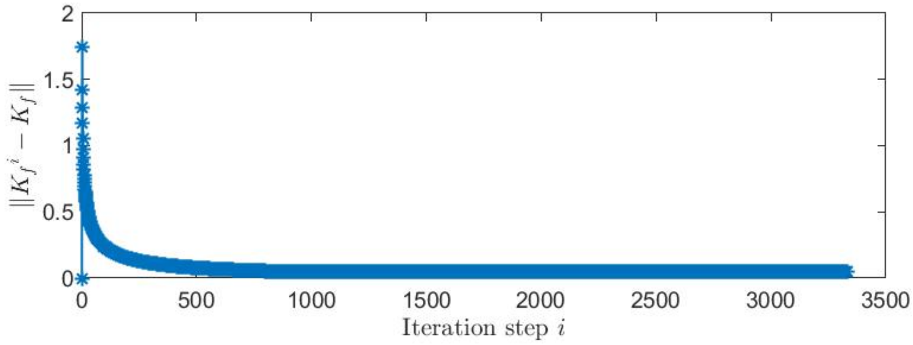

Fig. 1. The convergence process of K_f.

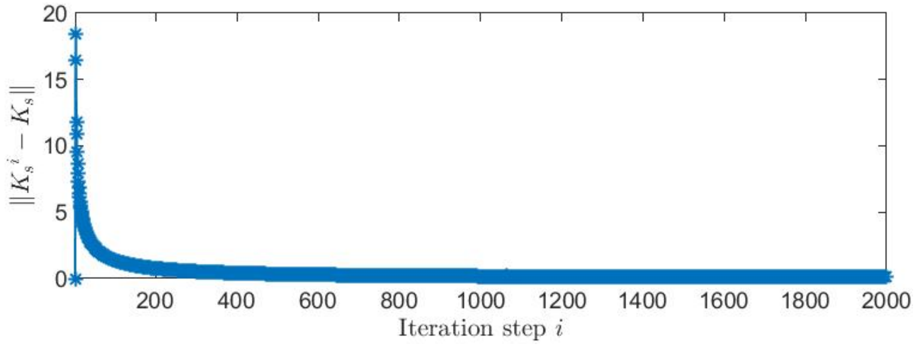

Fig. 2. The convergence process of K_s.

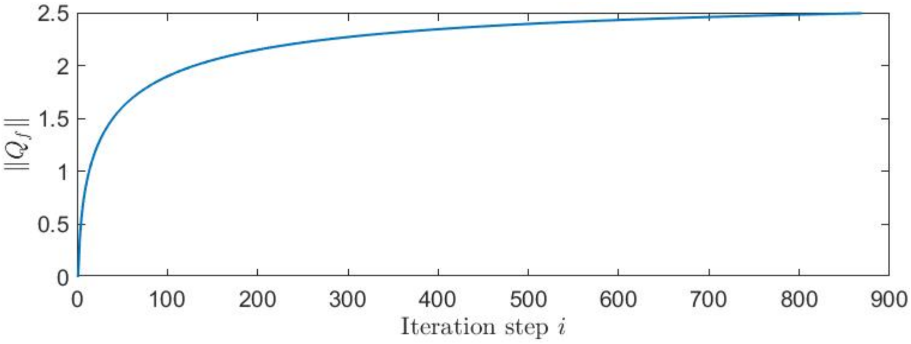

Fig. 3. The convergence process of Q_f.

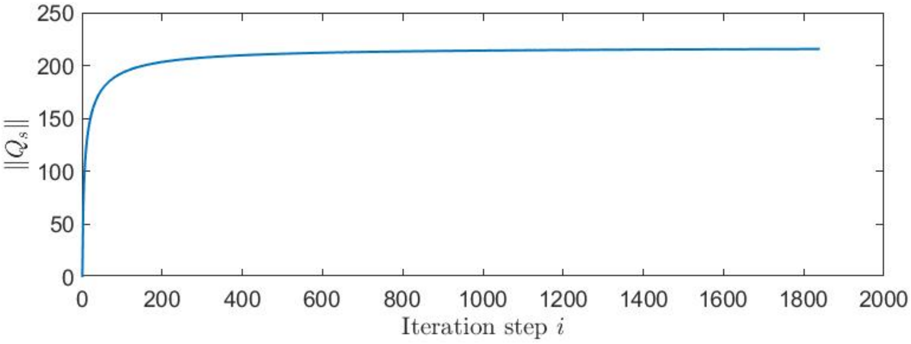

Fig. 4. The convergence process of Q_s.

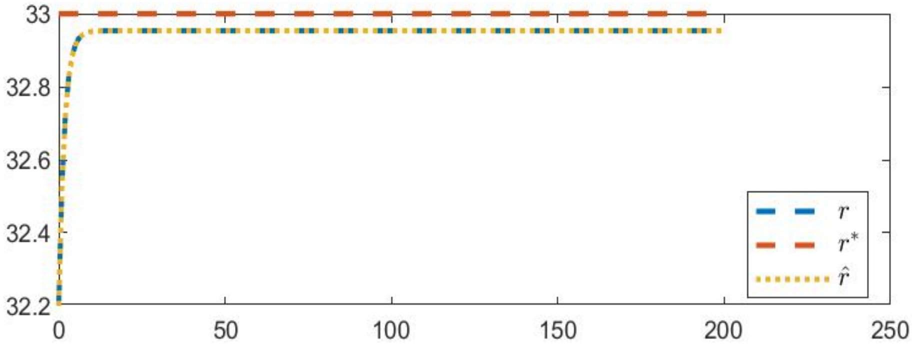

Fig. 5. Imitates performance of system output $r(k)$.

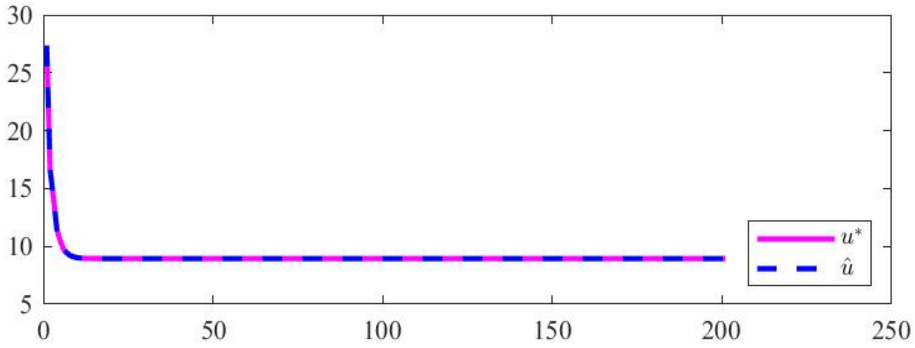

Fig. 6. Imitates performance of control $u^*(k)$.

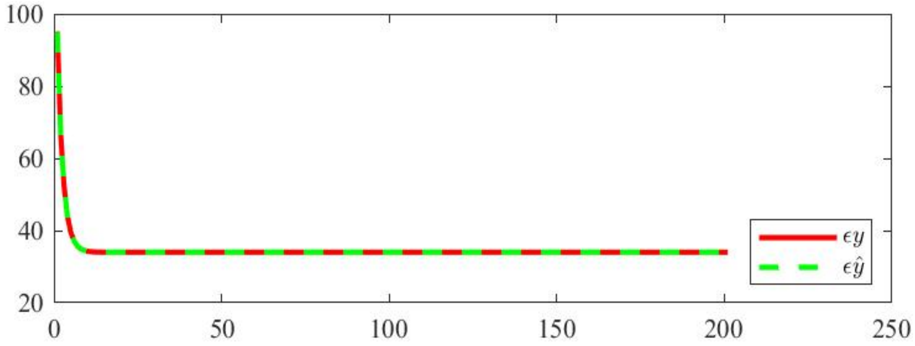

Fig. 7. Imitates performance of flow rate $\varepsilon y(k)$.

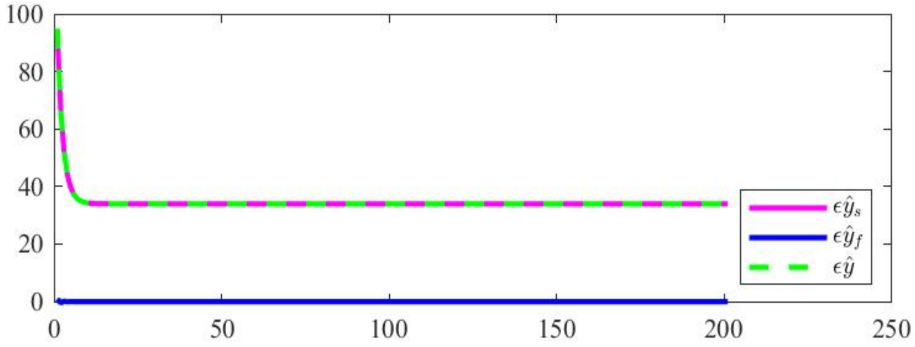

Fig. 8. Flow rate changing of flow rate $\varepsilon y(k)$.

Fig. 1-2 shows the convergence process of K_f and K_s by the proposed method. Fig. 3-4 shows the convergence process of equivalent weight Q_f and Q_s. Fig. 5-7 show that the system output $(\hat{r}(k), \hat{u}(k), \hat{y}(k))$ imitates the optimal behavior effect $(r(k), u^*(k), y(k))$, and the system output can track the desired operation index. Fig 8 shows the transient value of the underflow slurry flow rate converges to zero, and the final output reaches a quasi-steady state.

5 Conclusion

This paper studies the IRL problem of two time scale and proposes a tracking control method based on off-policy IRL. The method combines the singular perturbation method with IRL to reconstruct a cost function for the tracking control problem of the two time scale industrial process. Through the reconstructed cost function, an approximate optimal controller is learned to imitate the known optimal behavior data without the need for manual design of the cost function. The method designs a decentralized composite IRL control schemes: For the fast time scale system, a model-based IRL method is used for design; For the slow time scale system, a model-free off-policy IRL method is used, which only use the measured optimal behavior data to reconstruct the system cost function. Simulation experiments verified the imitation performance of the proposed method in two time scale systems. Although this method has shown excellent performance, it still needs to be further explored and overcome some limitations, especially the influence of noise interference and dynamic changes of the system, which cannot be ignored. Looking ahead, future work will focus on addressing stochastic perturbations that are common in real industrial environments, and work to extend this method to more complex system applications to further enhance its practicality and adaptability.

References

1. Xue, W., Fan, J., Lopez, V., Li, J., Jiang, Y., Chai, T., Lewis, F.: New methods for optimal operational control of industrial processes using reinforcement learning on two time scale. IEEE Transactions on Industrial Informatics 16(5), 3085-3099 (2020).
2. Kokotovic, P., Khalil, H., Reilly, J.: Singular perturbation methods in control : analysis and design. Academic Press, London (1986).
3. Jiang, Y., Jiang, Z.: Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. Automatica 48(7), 699-2704 (2012).
4. Li, J., Kiumarsi, B., Chai, T., Lewis, F., Fan, J.: Off-policy reinforcement learning for tracking in continuous-time systems on two time scale. IEEE Transactions on Neural Networks and Learning Systems 32(7), 4334-4346 (2021).
5. Abbeel, P., Ng, A.: Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the Twenty-First International Conference on Machine Learning,pp.1 – 8. Banff, Canda (2004).
6. Choi, S., Kim, S., Jin Kim, H.: Inverse reinforcement learning control for trajectory tracking of a multirotor UAV. Int. J. Control Autom. Syst. 15, 1826–1834 (2017).
7. Xue, W., Kolaric, P.,Li, J., Lian, B., Chai, T., Lewis, F.: Inverse reinforcement learning in tracking control based on inverse optimal control. IEEE Transactions on Cybernetics, 52(10), 10570-10581 (2022).
8. Kalman.: When is a linear control system optimal?. Journal of Basic Engineering 86(1), 51–60 (1964).
9. Zhang, H., Umenberger, J., Hu, X.: Inverse optimal control for discrete-time finite-horizon linear quadratic regulators. Automatica 110, 108593 (2019).
10. Molloy, T., Ford, J., Perez, T.: Finite-horizon inverse optimal control for discrete-time nonlinear systems. Automatica 87, 442-446 (2018).

11. Lian, B., Xue, W., Xie, Y., Lewis, F., Davoudi, A.: Off-policy inverse Q-learning for discrete-time antagonistic unknown systems. Automatica 155, 111171 (2023).
12. Xue, W., Lian, B., Li, J., Chai, T., Lewis, F.: Inverse reinforcement learning for trajectory imitation using static output feedback control. IEEE Transactions on Cybernetics 54(3), 1695-1707 (2024).
13. Xue, W., Lian, B., Fan, J., Kolaric, P.,Chai, T., Lewis, F.: Inverse reinforcement Q-learning through expert imitation for discrete-time systems. IEEE Transactions on Neural Networks and Learning Systems 34(5), 2386-2399 (2023).
14. Lian, B., Donge, V., Lewis, F., Chai, T., Davoudi, A.: Data-driven inverse reinforcement learning control for linear multiplayer games. IEEE Transactions on Neural Networks and Learning Systems 35(2), 2028-2041 (2024).
15. Lian, B., Xue, W., Lewis, F., Chai, T.: Robust inverse Q-learning for continuous-time linear systems in adversarial environments. IEEE Transactions on Cybernetics 52(12), 13083-13095 (2022).
16. Zhao, J., Yang, C., Dai, W., Gao, W.: Reinforcement learning-based composite optimal operational control of industrial systems with multiple unit devices. IEEE Transactions on Industrial Informatics 18(2), 1091-1101 (2022).
17. Klimushchev, A., Krasovskii, N.: Uniform asymptotic stability of systems of differential equations with a small parameter in the derivative terms. Journal of Applied Mathematics and Mechanics 25(9), 1011-1025 (1962).
18. Lian, B., Xue, W., Lewis, F., Chai, T.: Inverse reinforcement learning for adversarial apprentice games. IEEE Transactions on Neural Networks and Learning Systems 34(8), 4596-4609 (2023).
19. Lian, B., Xue, W., Lewis, F., Chai, T.: Inverse reinforcement learning for multi-player noncooperative apprentice games. Automatica 145, 110524 (2022).
20. Yang, Y.: Data-driven Optimal Control Method Based on Reinforcement Learning. Science Press, Beijing (2022).
21. Zhou, L., Zhao, J., Ma, L., Yang, C.: Decentralized composite suboptimal control for a class of two-time-scale interconnected networks with unknown slow dynamics. Neurocomputing 382,71-79 (2020).
22. Mukherjee, S., Bai, H., Chakrabortty, A.: On model-free reinforcement learning of reduced-order optimal control for singularly perturbed systems. In: 2018 IEEE Conference on Decision and Control (CDC),pp. 5288-5293. (2018).