Sequential Inverse Optimal Control of Discrete-Time Systems

Sheng Cao, Zhiwei Luo, and Changqin Quan

***Abstract*—This paper presents a novel sequential inverse optimal control (SIOC) method for discrete-time systems, which calculates the unknown weight vectors of the cost function in real time using the input and output of an optimally controlled discrete-time system. The proposed method overcomes the limitations of previous approaches by eliminating the need for the invertible Jacobian assumption. It calculates the possible-solution spaces and their intersections sequentially until the dimension of the intersection space decreases to one. The remaining one-dimensional vector of the possible-solution space's intersection represents the SIOC solution. The paper presents clear conditions for convergence and addresses the issue of noisy data by clarifying the conditions for the singular values of the matrices that relate to the possible-solution space. The effectiveness of the proposed method is demonstrated through simulation results.**

***Index Terms*—Inverse optimal control, promised calculation step, sequential calculation.**

I. Introduction

THE standard optimal control problem concerns finding the state and input trajectories for a dynamical system. In this regard, the inverse optimal control (IOC) approach is generally employed to obtain the weighting parameters of the cost function using the input/output data generated by optimal control.

In many fields, such as robotics [1], biological systems [2], [3] and marketing systems [4], the optimization of the systems' behavior has been studied using the IOC method.

In [1], a method is proposed to estimate the cost function of human operators for human-robot interaction control. The IOC method was utilized in [5] to analyze the route choices of taxi drivers. To evaluate the cost combination of human motion, in [6], the IOC method was utilized to analyze the reaching movement of the human arm. Furthermore, biological behavior has been modeled as an inverse linear quadratic regulator problem, and an adaptive method was proposed to model and analyze human reach-to-grasp behavior by [7].

As mentioned in the literature, there are two main groups of IOC. One has a hierarchy structure that updates the cost function in the higher stage while the forward optimal control problem is repeatedly solved in the lower stage to minimize the evaluation function between the original data and the generated data. In [8], where the IOC problem is formulated by another form (the inverse-reinforcement-learning method), cost weights are adjusted to better evaluate the observation-feature values and maximize the entropy of the trajectory-probability distribution. When considering the IOC problem as a special bilevel optimal control problem [9], where the lower level is the optimal control problem and the upper level is the inversion problem, the IOC problem has been considered for simple dynamical models [10]. To analyze locomotion movements, the authors of [3] proposed a bilevel optimization method based IOC method. In [11], mathematical programs with complementary constraints in the context of the IOC method are discussed for the application of locomotion analysis. In [12], the bilevel optimization problem for IOC is transformed into a single-level problem, and a globally optimal solution is computed. Although these methods are effective and have been utilized in many applications, such as human-motor behavior analysis, robot navigation, and autonomous driving, they require significant computational costs in the lower stage to repeatedly verify the updated cost function.

Conversely, the second class of IOC research focus on solving this problem by exploiting several optimality equations, such as Pontryagin's maximum principle-based equations [13], and Euler-Lagrange equations [2]. In [14], a linear combination of feature functions with unknown weight parameters was formulated to approximate the original optimal cost function. In [15], the recovery of the weight parameters of the finite-horizon, discrete-time optimal control was considered. For practical applications, it is important to obtain the cost weights in real time. In [16], a method is proposed for the online calculation of discrete-time IOC in both finite and infinite horizons; however, it requires the invertibility condition of a Jacobian. In addition, the convergence of the cost weights has not been theoretically investigated.

Control Lyapunov function (CLF) based IOC is an essential paradigm in control theory, where a stabilizing feedback is designed first and then shown to be optimal for a cost function. In recent years, there have been numerous contributions to address the limitations of CLF-based IOC. The extended Kalman filter (EKF) has been utilized in some studies to construct a CLF [17], [18]. Furthermore, some researches have applied CLF-based IOC to non-linear inverse optimal control

Manuscript received April 24, 2023; revised June 12, 2023; accepted July 5, 2023. Recommended by Associate Editor Lei Zou. *(Corresponding author: Sheng Cao.)*

Citation: S. Cao, Z. Luo, and C. Quan, "Sequential inverse optimal control of discrete-time systems," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 3, pp. 608–621, Mar. 2024.

The authors are with the Graduate School of System Informatics, Kobe University, Kobe 657-8501, Japan (e-mail: jasonsosen@gold.kobe-u.ac.jp; luo@gold.kobe-u.ac.jp; quanchqin@gold.kobe-u.ac.jp).

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/JAS.2023.123762

problems [19]. Additionally, some works have developed data-driven non-linear stabilization schemes utilizing the Koopman operator [20], and focused on learning simple polynomial CLFs from counterexamples and demonstrations for non-linear dynamical systems [21]–[23]. Despite these advancements, notable limitations of CLF-based IOC method persist, including the inability to explicitly specify a clear optimal cost function and the difficulty in selecting and adjusting parameters, which often requires extensive experience and expert knowledge.

Moreover, the problem of noisy data is a source of concern in IOC research. The authors of [24] analyzed the IOC problem for discrete-time finite-horizon LQR as an optimization problem and proved that the solution to the problem is statistically consistent. They utilized the data from multi-observations to complete the offline IOC method. In [25], building on the work of [24], the authors recast the IOC problem as the identification of a parameterized causal and anti-causal mixed system exited by boundary conditions. They clarified sufficient identifiability conditions for the unknown parameters in terms of the system model. Besides, [26], [27] also emphasize the consideration of data noise. Although the aforementioned methods are effective in tackling problems of noise for the offline IOC problem of discrete-time finite-horizon LQR, it is still necessary to consider the noisy problem in the IOC method in (1) online calculation and (2) nonlinear system's IOC method.

This paper proposes a novel SIOC method to address the issues mentioned above. The method recovers the cost function of an optimal control problem from input/output data acquired step-by-step, and solves the inverse optimal control problem sequentially with a promised convergence speed. By assuming the positivity of a Jacobian, this method is applicable even if the invertible Jacobian assumption required in [16] is not satisfied. Most importantly, the method clarifies the conditions for the convergence of the weight estimation in every step, which can be utilized in the analysis of real applications. On the other hand, in this study, the noisy data problem is considered for the online recovery of the cost-weight vector. We also analyze the effect of noisy data on the possible-solution space and propose a one step selection method of the possible-solution space for the noisy case using noisy data from multiobservations. Contribution of this research comes from three aspects.

1) The first contribution is the promised solution's calculation speed and the conditions that enable it. We discuss the conditions for the decrease in the dimension of the intersection space. Consequently, if any of the conditions listed (Theorem 1) is satisfied, the dimension of the intersection of the solution's space will decrease in each step, allowing us to obtain solutions within a promised number of steps. This is beneficial for applications that require high-speed calculations.

2) The second is the establishment of a sequential IOC method without utilizing the invertible Jacobian assumption in [16]. Notably, although the invertible Jacobian assumption is widely used in the solution of the forward finite and infinite horizon optimal control of the discrete system, as highlighted in ([28]), some necessary conditions are proposed to apply the method for the system dynamics even if the invertible Jacobian assumption is not satisfied.

3) The third is the development of an efficient method for tackling the noisy data problem in the online IOC calculation. If we have prior knowledge of the maximum error value, we can analyze the effect of the error on the possible-solution space in our method and propose two conditions for selecting the possible-solution space. Simulation results show that this method is effective.

II. Problem Formulation

Consider the dynamics of a discrete system

$$x_{k+1} = f(x_k, u_k) \tag{1}$$

where $f(:,:) : R^n \times \mathcal{U} \to R^n$ is a continuously differentiable function, $x_k = [x_k^1, \ldots, x_k^n]^T \in R^n$ represents the system states, $u_k = [u_k^1, \ldots, u_k^m]^T \in \mathcal{U}$ denotes the control input of the system belonging to a closed, bounded and convex constrained set $\mathcal{U} \subseteq R^m$. $x_{[0,K]}$ denotes the state sequence $\{x_k : 0 \le k \le K\}$ and $u_{[0,K]}$ denotes control input sequence $\{u_k : 0 \le k \le K\}$.

In the standard optimal control problem, we design the optimal control input $u^*_{[0,K]}$ and obtain a series of optimal state $x^*_{[0,K]}$ to minimize the following cost function, subject to dynamics (1) (upperscript $*$ stands for the optimal condition):

$$C(x_k, u_k, q) = \sum_{k=0}^{K} q^T F(x_k, u_k) \tag{2}$$

where $F(x_k, u_k)$ is a feature vector function defined as

$$F(x_k, u_k) = [F_{XU}^T, F_U^T]^T \in R^{n_f}. \tag{3}$$

Here, the terminal step K can be $K < \infty$ or $K \to \infty$. The vector, $q = [q_{xu}^T, q_u^T]^T \in R^{n_f}$ represents cost weights, in which vector $q_{xu} \in \mathbb{R}^{n_{xu}}$ represents a weight vector with respect to x_k and u_k, and vector $F_{XU} = [F_{x_k u_k(1)}, \ldots, F_{x_k u_k(n_{xu})}]^T \in \mathbb{R}^{n_{xu}}$. The scalar $F_{x_k u_k(i)}$ represents the i-th feature function related to x_k^i and u_k^i. $q_u \in \mathbb{R}^{n_u}$ denotes the weight vector accounting for the control input u_k, while $F_U = [F_{u_k(1)}, \ldots, F_{u_k(n_u)}]^T \in \mathbb{R}^{n_u}$ represents the feature function purely related to u_k. It is assumed that the norm, $\|q\| = 1$, is fixed and known priorly. The total number of features n_f satisfies $n_f = n_{xu} + n_u$.

In our problem, we assume that $x_{[0,K]}$ as well as $u_{[0,K]}$ constitutes a solution to minimize (2) for system dynamics (1). The objective of this research is to realize the online estimation of the cost weight vector q in cost function (2), i.e., to calculate vector q online for the given system state x_k and the control input u_k without the storage and batch processing. The time horizon K is known priorly.

III. Sequential IOC for the Noise Free Case

In this section, we introduce the maximum principle for the finite and infinite horizon, optimal control problems, which is applicable when any condition provided in Assumption 1 is satisfied. Based on the introduced maximum principle, we propose a sequential IOC method.

A. Maximum Principle for the Finite and Infinite Horizon Optimal Control Problems

The Hamiltonian function associated with the optimal control problems given by (1) and (2) is defined as

$$H_k(x_{k-1},u_{k-1},\lambda_k,q)=q^TF(x_{k-1},u_{k-1})+\lambda_k^Tf(x_{k-1},u_{k-1}) \tag{4}$$

where $\lambda_k\in R^n$ $(k>0)$ denotes the co-state vector.

The column vectors of the partial derivatives of the Hamiltonian with respect to x_{k-1} and u_{k-1} are $\nabla_xH_k(x_{k-1},u_{k-1},\lambda_k,q)\in R^n$ and $\nabla_uH_k(x_{k-1},u_{k-1},\lambda_k,q)\in R^m$, respectively.

Assumption 1: The partial derivative of the dynamics satisfies any of the following two conditions for all $k\geq 0$:

1) $\frac{\partial f}{\partial x_k}$ is invertible;

2) $\frac{\partial f}{\partial x_k}$ satisfies the following positivity condition:

$$\frac{\partial f^{n_j}(x_k,u_k)}{\partial x_k^{n_i}}\geq 0\ \forall n_i,n_j\in 1,\ldots,n \text{ where } n_i\neq n_j \tag{5}$$

$$\frac{\partial f^{n_j}(x_k,u_k)}{\partial x_k^{n_j}}>0\quad \forall n_j\in 1,\ldots,n \tag{6}$$

where f^{n_j} represents the n_j-th element of function $f(x_k,u_k)$ and $x_k^{n_i}$ represents n_i-th element of vector x_k.

In [28] (Theorems 2.2 and 2.6), it has been shown that both the assumptions of Jaocbian's invertibility (Assumption 1-1)) and positivity (Assumption 1-2)) can be used to establish Pontryagin's maximum principles in both finite and infinite horizon discrete-time optimal control problems. These assumptions are especially necessary for infinite horizon problems. While the previous study [16] of online IOC assumed Jacobian's invertibility to establish the discrete-time maximum principle, this assumption may not hold for some system dynamics. Therefore, the Jacobian's positivity assumption (condition) is proposed in [28] as an alternative to the invertible assumption (condition). For example, it suffices to consider the case where $\frac{\partial f^{n_j}(x_k,u_k)}{\partial x_k^{n_i}}=1$ for all n_i,n_j to see that Assumption 1-2) is fulfilled and $\frac{\partial f}{\partial x_k}$ is not invertible since its rank is equal to 1.

In addition, as in [16], the inactive constraint times of the control input are defined as

Definition 1: Given the control input, u_k, for $k>0$,

$$\kappa_k\triangleq\{0\leq k\leq l: u_k\in \text{int}(\mathcal{U})\} \tag{7}$$

is defined as the set when the control constraints are inactive. Here l represents time larger than 0, and int($\mathcal{U}$) denotes the interior of the inactive control constraint set, $\mathcal{U}$.

Based on [16], [28]–[30], for the above assumptions and definition, the following lemma holds:

Lemma 1: Suppose that the optimal control problems defined by (1) and (2) are solved by trajectories $x_{[0,K]}$ and $u_{[0,K]}$ and if the Assumptions 1-1) or 1-2) hold, then there exist co-state vectors $\lambda_0,\ldots,\lambda_K$ that satisfy the combined Pontryagin's maximum principle as

$$\bar{F}_{x(k-1)}^Tq+\bar{f}_{x(k-1)}^T\lambda_k=\lambda_{k-1} \tag{8}$$

for all $0\leq k\leq K$ with $\lambda_{K+1}=0$ if $K<\infty$, and λ_{K+1} undefined if $K\to\infty$, and

$$\bar{F}_{u(k-1)}^Tq+\bar{f}_{u(k-1)}^T\lambda_k=0 \tag{9}$$

for all $k\in\kappa_k$, where κ_k denotes the inactive constraint times up to and including time K.

Here, $\bar{F}_{x(k)}=\frac{\partial F}{\partial x_k}$, $\bar{F}_{u(k)}=\frac{\partial F}{\partial u_k}$, $\bar{f}_{x(k)}=\frac{\partial f}{\partial x_k}$ and $\bar{f}_{u(k)}=\frac{\partial f}{\partial u_k}$. The co-state λ_k varies in backward recursion in discrete-time optimal control.

Proof: Generally, (8) and (9) can be obtained by calculating the gradients of (2) for (1) using (4) with respect to the vectors x_k and u_k, respectively. The brief proof of this lemma for the assumption is given in [16] (Lemma 1) by using the results of [29] (Proposition 3.3.2) and [30] (Theorem 2). It is also satisfied for Assumption 1-2), as is proven in [28] (Theorem 2.6). ■

B. Construction of the Sequential IOC Method

If the optimal control problems of (1) and (2) are solved by $(x_{[T_0,T_f]},u_{[T_0,T_f]})$ which are the solutions to (8) and (9), then by considering (9) in steps k and $k-1$ and substituting (8) in step k, we have

$$H_ks_k=0 \tag{10}$$

where $H_k=\begin{bmatrix}\bar{f}_{u(k-1)}^T\bar{f}_{x(k)}^T & \bar{F}_{u(k-1)}^T+\bar{f}_{u(k-1)}^T\bar{F}_{x(k)}^T\\ \bar{f}_{u(k)}^T & \bar{F}_{u(k)}^T\end{bmatrix}$ and $s_k=\begin{bmatrix}\lambda_{k+1}\\ q\end{bmatrix}$, by considering (8), a backward recursive relation from s_k to s_{k-1} can be formulated as follows:

$$M_ks_k=s_{k-1} \tag{11}$$

where M_k is defined as $M_k=\begin{bmatrix}\bar{f}_{x(k)}^T & \bar{F}_{x(k)}^T\\ 0_{n_f\times n} & I_{n_f}\end{bmatrix}\in R^{(n+n_f)\times(n+n_f)}$.

Here, $I_{n_f}\in\mathbb{R}^{n_f\times n_f}$ denotes the unit matrix and $0_{n_f\times n}\in\mathbb{R}^{n_f\times n}$ denotes the matrix where all the elements are zero. Notably, (10) holds only when $k,k-1\in\kappa_k$ and (11) holds always even if $k\notin\kappa_k$.

Thus, by introducing all historical M_i $(i=h,\ldots,k)$ with h denoting step $h\leq k$, it is easy to backward calculate s_h of step h with

$$\bar{M}_{h:k}s_k=s_h \tag{12}$$

where $\bar{M}_{h:k}$ denotes a matrix defined as $\bar{M}_{h:k}=\prod_{l=h+1}^{k}M_l$. and it satisfies the forward recursion as $\bar{M}_{h:k}=\bar{M}_{h:k-1}M_k$.

Since vector s_h contains the cost-weight vector, q, as well as the co-state λ_{h+1}, our goal of constructing the online IOC changes to finding vector s_h based on the finite forward steps of k.

From (10), it is known that s_k locates in the null space of H_k, and s_k can be calculated as follows:

$$s_k=(I_N-H_k^+H_k)r_k \tag{13}$$

where $r_k\in\mathbb{R}^N$ denotes an arbitrary vector and H_k^+ represents the pseudo inverse matrix of H_k.

For step i in the duration from step h $(T_0\leq h<k)$ to k $(T_0<k\leq T_f)$, we have

$$\Phi_{h(i)}r_i=\bar{M}_{h:i}\Theta_ir_i=s_h \tag{14}$$

where $\Phi_{h(i)} = \bar{M}_{h:i}(I_N - H_i^+ H_i) = \bar{M}_{h:i}\Theta_i$ with $h \le i \le k$ and $\Theta_i = I_N - H_i^+ H_i$.

Here, the column vector space of $\Phi_{h(i)}$ can be denoted as $\Gamma_{\Phi_{h(i)}} = \text{span}(\Phi_{h(i)})$ for $h \le i \le k$, where span(:) denotes the span of the matrix. From (14), we have $s_i \in \Gamma_{\Phi_{h(i)}}$ for $h \le i \le k$. Therefore, the spaces of $\Gamma_{\Phi_{h(h)}}, \ldots, \Gamma_{\Phi_{h(k)}}$ are the possible-solution spaces, and the intersection of these possible-solution spaces is defined as $\Gamma_{\Omega_{h:k}} = \Gamma_{\Phi_{h(h)}} \cap \cdots \cap \Gamma_{\Phi_{h(k)}}$.

Thus, we have $s_h \in \Gamma_{\Omega_{h:k}}$, which implies that vector s_h exists and always belongs to the intersection subspace.

To obtain s_h, it is necessary to discuss the decrease in the dimension of $\Gamma_{\Omega_{h:k}}$ as k increases.

Proposition 1: If $\Gamma_{\Omega_{h:i-1}} \nsubseteq \Gamma_{\Phi_{h(i)}}$, the dimension of the intersection $\Gamma_{\Omega_{h:i}}$ will be decreased; that is,

$$\dim(\Gamma_{\Omega_{h:i-1}}) > \dim(\Gamma_{\Omega_{h:i}}) \tag{15}$$

where $\Gamma_{\Omega_{h:i}}$ means $\Gamma_{\Omega_{h:i}} = \Gamma_{\Phi_{h(h)}} \cap \cdots \cap \Gamma_{\Phi_{h(i)}}$.

Proof: It is clear that, for any vector space, $\Gamma_{\Phi_{h(i)}}$ and $\Gamma_{\Phi_{h(j)}}$,

$$\dim(\Gamma_{\Phi_{h(i)}} \cap \Gamma_{\Phi_{h(j)}}) \le \dim(\Gamma_{\Phi_{h(i)}}) \tag{16}$$

the equality sign holds when $\Gamma_{\Phi_{h(i)}} \subseteq \Gamma_{\Phi_{h(j)}}$.

Since $\Gamma_{\Omega_{h:i}} = \Gamma_{\Omega_{h:i-1}} \cap \Gamma_{\Phi_{h(i)}}$, we have

$$\dim(\Gamma_{\Omega_{h:i}}) \le \dim(\Gamma_{\Omega_{h:i-1}}), \quad h < i < K \tag{17}$$

the equality sign holds when $\Gamma_{\Omega_{h:i-1}} \subseteq \Gamma_{\Phi_{h(i)}}$.

Therefore, if $\Gamma_{\Omega_{h:i-1}} \nsubseteq \Gamma_{\Phi_{h(i)}}$, the dimension of $\Gamma_{\Omega_{h:i}}$ will decrease as k increases. ■

From Proposition 1, it is clear that with the increase in the step, at some step instant k_f, the rank of the common intersection subspace of $\Gamma_{\Phi_{h(h)}}, \ldots, \Gamma_{\Phi_{h(k_f)}}$ becomes one and this unique intersection is s_h.

Here, we give the main result to clarify the condition for the decrease in the dimension of intersection space $\Gamma_{\Omega_{h:i}}$.

Theorem 1: Under Assumption 1-1), if any of the following two conditions is satisfied, then $\Gamma_{\Omega_{h:i-1}} \nsubseteq \Gamma_{\Phi_{h(i)}}$:

a)

i) $\bar{f}^T_{u(\bar{j})} \forall h \le j \le i-1$ is full rank;

ii) $\bar{F}^T_{u(i)}$ is full rank.

b)

i) $\dim(\bar{f}^T_{u(j)}) < m \forall h \le j \le i-1$;

ii) $\bar{F}^T_{u(i)}$ is full rank;

iii)

$$\sum_{j=h}^{i-1} \dim(\text{null}(\bar{f}^T_{u(j)})) < N_z(i)$$

where $N_z(i)$ denotes the dimension of the null space of $\Gamma_{\Phi_{h(i)}}$ and dim(:) denotes the dimension of the column vector space of the matrix.

Proof: The proof of this theorem is shown in Appendix. ■

Therefore, if Conditions a) or b) of Theorem 1 is satisfied, the dimension of the intersection of the possible-solution space will decrease in every step.

C. Calculation of Vector s_h

1) Calculation of the Intersection Space: Here, Ω_s is the matrix related to the intersection of the possible-solution spaces, which is initialized in step h and is updated in every cycle.

From $\Gamma_{\Omega_{h:i-1}} \cap \Gamma_{\Phi_{h(i)}} = \text{null}(\text{null}(\Gamma_{\Phi_{h(i)}}) \cup \text{null}(\Gamma_{\Omega_{h:i-1}}))$, we can calculate Ω_s by

$$\Omega_s = \Omega_{h:i} = \text{null}(Y_{h(i)}) \tag{18}$$

where $Y_{h(i)} = [\text{null}(\Omega_{h:i-1})^T, \text{null}(\Phi_{h(i)})^T]^T$ can be represented by singular-value decomposition by $Y_{h(i)} = W\Lambda V^T$. W and V are unitary matrices, Λ is a rectangular diagonal matrix with non-negative values on the diagonal.

When $\dim(\Gamma_{\Omega_{h:i}}) = 1$, Ω_s becomes vector V_n, which is the row vector of V related to the smallest singular-value of Λ. From $Y_{h(i)} s_h = 0$, it is clear that s_h may maintain the same direction with V_n or $-V_n$. From the fact that right s_h can make the cost function be positive while using $-s_h$ instead of s_h make the cost function become negative on the contrary, s_h can be selected as

$$s_h = \begin{cases} V_n a_p, & C(x_k, u_k, \hat{q}) \ge 0 \\ -V_n a_p, & C(x_k, u_k, \hat{q}) < 0 \end{cases} \tag{19}$$

where $\hat{q} = [V_n a_p]_{n+1:n+n_f}$ is the vector constructed by the $(n+1)$-th element to the $(n+n_f)$-th element of vector $V_n a_p$, $a_p = \frac{\|q\|}{\|[V_n]_{n+1:n+n_f}\|}$, $[V_n]_{n+1:n+n_f}$ denotes the vector constructed by the $(n+1)$-th element to the $(n+n_f)$-th element of V_n, and the $\|q\|$ is known prior. Here, we use a previously known norm of q to scale V_n to obtain a unique and right-cost weight vector.

2) Calculation With Control Constraints: When control constraints exist, the calculation procedure is designed as follows:

$$\begin{cases} \Gamma_{\Omega_{h:i}} = \Gamma_{\Omega_{h:i-1}} \cap \Gamma_{\Phi_{h(i)}}, & u_i \in \text{int}(\mathcal{U}) \ \& \ u_{i-1} \in \text{int}(\mathcal{U}) \\ \textit{Reinitialize} \quad \Gamma_{\Omega_{h:i}}, & u_i \in \text{int}(\mathcal{U}) \ \& \ u_{i-1} \notin \text{int}(\mathcal{U}) \\ \textit{Skip the Step}, & \text{Otherwise.} \end{cases} \tag{20}$$

We halt the calculation when the control constraint is active and reinitialize, restart the IOC calculation when it is inactive. The high calculation speed of our method makes it possible to quickly complete the IOC calculation after the active duration of the control constraints so that it is not necessary to store the data of the duration before the control constraints are active [16].

The total calculation in the noise-free case is shown in Algorithm 1.

IV. When Noise Exists

When measurement noise or numerical errors exist, the noisy-system state, $\hat{x}$, and control input, $\hat{u}$, introduce errors into the calculation of $\Phi_{h(i)}$, resulting in the deviation of the calculated possible-solution space from the correct result. Here, we also introduced a method for tackling this problem.

First, in this research, we assume that the noise of the measured $\hat{x}$ and $\hat{u}$ satisfies the following conditions:

$$\begin{aligned} \|\tilde{x}\| &= \|x - \hat{x}\| \le \epsilon_x \\ \|\tilde{u}\| &= \|u - \hat{u}\| \le \epsilon_u \end{aligned} \tag{21}$$

where ϵ_x and ϵ_u are two positive scalars. Furthermore, we

introduce the following assumption of the system dynamics and feature function.

Algorithm 1 Online Implementation (Noise-Free Case)

Input: $x_i, u_i, \|q\|$

Output: s_h

Initialization :

1: Compute a H_h and $\Phi_{h(h)}$.

2: Initialize matrix $\Omega_s = \Phi_{h(h)}$ which represents the intersection of possible-solution spaces.

LOOP Process

3: **for** $i = h+1$ to K **do**

4: **if** $u_i \in \text{int}(\mathcal{U})$ **then**

5: **if** $u_{i-1} \notin \text{int}(\mathcal{U})$ **then**

6: Calculate $\Phi_{h(i)}$ and reinitialize $\Omega_s = \Phi_{h(i)}$

7: **Continue**.

8: **end if**

9: Calculate $\Phi_{h(i)}$, $\text{null}(\Phi_{h(i)})$ and $\text{null}(\Omega_s)$.

10: Calculate $\Gamma_{\Omega_s} \cap \Gamma_{\Phi_{h(i)}}$ by constructing the matrix $Y_{h(i)} = [\text{null}(\Phi_{h(i)})^T, \text{null}(\Omega_s)^T]^T$ and compute the null space of $Y_{h(i)}$. Here, Γ_{Ω_s} denotes the vector space of Ω_s.

11: **if** $\text{rank}(Y_{h(i)}) < N-1$ **then**

12: Update $\Omega_s = \text{null}(Y_{h(i)})$

13: **end if**

14: **if** $(\text{rank}(Y_{h(i)}) = N-1)$ **then**

15: Ω_s is decreased to a vector with $\Omega_s = \text{null}(Y_{h(i)})$.

16: Get the unique solution following (19).

17: **end if**

18: **end if**

19: **end for**

20: **return** s_h

Assumption 2: Function $\bar{f}_{u(k)}, \bar{f}_{x(k)}, \bar{F}_{u(k)}, \bar{F}_{x(k)}$ satisfy

$$\begin{aligned}
&\|\bar{f}_{u(k)} - \hat{\bar{f}}_{u(k)}\|_F \le \psi_u^f \|(u_k - \hat{u}_k)\| \le \psi_u^f \epsilon_u \\
&\|\bar{f}_{x(k)} - \hat{\bar{f}}_{x(k)}\|_F \le \psi_x^f \|(x_k - \hat{x}_k)\| \le \psi_x^f \epsilon_x \\
&\|\bar{F}_{u(k)} - \hat{\bar{F}}_{u(k)}\|_F \le \psi_u^F \|(u_k - \hat{u}_k)\| \le \psi_u^F \epsilon_u \\
&\|\bar{F}_{x(k)} - \hat{\bar{F}}_{x(k)}\|_F \le \psi_x^F \|(x_k - \hat{x}_k)\| \le \psi_x^F \epsilon_x
\end{aligned} \tag{22}$$

where symbols $\hat{\bar{f}}_{u(k)}$, $\hat{\bar{f}}_{x(k)}$, $\hat{\bar{F}}_{u(k)}$ and $\hat{\bar{F}}_{x(k)}$ denote the terms with noises and $\|:\|_F$ denotes the Frobenius norm, and ψ_u^f, ψ_x^f, ψ_u^F and ψ_x^F represent the positive scalars.

In practical applications, the parameters ψ_u^f, ψ_x^f, ψ_u^F, and ψ_x^F represent the degree of influence of the disturbances in the system input u_k and system state x_k on the system dynamics equations $\bar{f}u(k)$, $\bar{f}x(k)$, $\bar{F}u(k)$, and $\bar{F}x(k)$. They quantify the weight of input and state disturbances in the system dynamics equations and feature functions, which is crucial for designing robust controllers and evaluating system performance. In order to determine the values of these parameters, one would typically rely on prior knowledge about the system, such as the system's sensitivity to disturbances in its input and state variables. This information can be obtained from system identification or previous experimental data.

Using the noisy data, (10) and (11) can be expressed as follows:

$$\begin{aligned}
\hat{H}_k s_k &= e_k^h \\
\hat{M}_k s_k &= s_{k-1} + e_k^m
\end{aligned} \tag{23}$$

where $\hat{H}_k = \begin{bmatrix} \hat{\bar{f}}_{u(k-1)}^T \hat{\bar{f}}_{x(k)}^T & \hat{\bar{F}}_{u(k-1)}^T + \hat{\bar{f}}_{u(k-1)}^T \hat{\bar{F}}_{x(k)}^T \\ \hat{\bar{f}}_{u(k)}^T & \hat{\bar{F}}_{u(k)}^T \end{bmatrix}$ and $\hat{M}_k = \begin{bmatrix} \hat{\bar{f}}_{x(k)}^T & \hat{\bar{F}}_{x(k)}^T \\ 0_{n_f \times n} & I_{n_f} \end{bmatrix}$, respectively.

Here, since the existence of noise will affect $\hat{\Phi}_{h(i)} = \hat{\bar{M}}_{h:i} \hat{\Theta}_i$ at each step for $(h \le i < k)$, s_h will not belong to the column vector space of $\hat{\Phi}_{h(i)}$ and the direction of the final estimated cost-weight vector will deviate from the correct result.

To solve this problem, we adjust the calculation of the possible-solution space by selecting an appropriate method to replace the original calculation method of Θ_i and $\text{null}(\Phi_{h(i)})$ in each step in the noise-free case. With the following method, this goal is achieved in three steps:

i) Based on $\hat{H}_k$ and the range of the noise error, we calculate matrix, $T_{\hat{H}_k}$, which spans a vector space containing s_k.

ii) We utilize $T_{\hat{H}_k}$ and $\hat{M}_k$ to calculate matrix $\hat{\Phi}_{h(k)}$ and subsequently calculate the matrix $T_{\hat{\Phi}_{h(k)}}$ which may span a vector space containing s_h.

iii) We calculate the intersection of spaces spanned by $T_{\hat{\Phi}_{h(k)}}$ $(h \le i \le k)$ obtained in each step.

1) Method of Tackling the Effect of Noises on $\hat{H}_k$: From (23) and $H_k s_k = 0$, we have

$$\hat{H}_k s_k = e_k^h = \tilde{H}_k s_k \tag{24}$$

where $\tilde{H}_k = H_k - \hat{H}_k$. By performing singular-value decomposition (SVD) of matrix $\hat{H}_k$, we have

$$\hat{H}_k = W_{\hat{H}_k} \Lambda_{\hat{H}_k} V_{\hat{H}_k}^T$$

with two unitary matrices $W_{\hat{H}_k}$ and $V_{\hat{H}_k}$ and a rectangular diagonal singular-value matrix, $\Lambda_{\hat{H}_k}$.

Here, it is clear that the existence of the $\tilde{H}_k$ term affects the estimation result; as such, s_k will not lie in the null space of $\hat{H}_k$. We analyze the condition of the Frobenius norm of $\tilde{H}_k$ in the following lemma first.

Lemma 2: The Frobenius norm of $\tilde{H}_k$ is bounded and satisfies the following:

$$\|\tilde{H}_k\|_F \le \sqrt{\beta_1 + \beta_2} \tag{25}$$

where

$$\begin{aligned}
\beta_1 &= 2\psi_u^{f2} \epsilon_u^2 \text{Tr}(\hat{\bar{f}}_{x(k)}^T \hat{\bar{f}}_{x(k)}) \\
&\quad + \psi_x^{f2} \epsilon_x^2 \text{Tr}(\hat{\bar{f}}_{u(k-1)}^T \hat{\bar{f}}_{u(k-1)}) + \psi_u^{f2} \epsilon_u^2 \\
\beta_2 &= 2\psi_u^{F2} \epsilon_u^2 + 3\psi_x^{F2} \epsilon_x^2 \text{Tr}(\hat{\bar{F}}_{x(k)}^T \hat{\bar{F}}_{x(k)}) \\
&\quad + 3\psi_x^{F2} \epsilon_x^2 \text{Tr}(\hat{\bar{f}}_{u(k-1)}^T \hat{\bar{f}}_{u(k-1)}) + \psi_u^{F2} \epsilon_u^2.
\end{aligned}$$

Proof: From the definitions of H_k and $\hat{H}_k$, $\tilde{H}_k^T \tilde{H}_k$ can be represented as

$$\tilde{H}_k^T \tilde{H}_k = \begin{bmatrix} h_1 & h_0 \\ h_0^T & h_2 \end{bmatrix} \tag{26}$$

where $h_1 = (\hat{\tilde{f}}_{x(k)}\tilde{\tilde{f}}_{u(k-1)} + \tilde{\tilde{f}}_{x(k)}\hat{\tilde{f}}_{u(k-1)})(\tilde{\tilde{f}}^T_{u(k-1)}\hat{f}^T_{x(k)} + \hat{f}^T_{u(k-1)}\tilde{\tilde{f}}^T_{x(k)}) + \tilde{\tilde{f}}_{u(k)}\tilde{\tilde{f}}^T_{u(k)}$ and $h_2 = (\tilde{\tilde{F}}_{u(k-1)} + \hat{\tilde{F}}_{x(k)}\tilde{\tilde{f}}_{u(k-1)} + \tilde{\tilde{F}}_{x(k)}\hat{\tilde{f}}_{u(k-1)})(\tilde{\tilde{F}}^T_{u(k-1)} + \tilde{\tilde{f}}^T_{u(k-1)}\hat{\tilde{F}}^T_{x(k)} + \hat{\tilde{f}}^T_{u(k-1)}\tilde{\tilde{F}}^T_{x(k)}) + \tilde{\tilde{F}}_{u(k)}\tilde{\tilde{F}}^T_{u(k)}$.

Then, from (21) and (22), we have

$$\mathrm{Tr}(h_1) \le 2\psi_u^{f2}\epsilon_u^2\mathrm{Tr}(\hat{\tilde{f}}^T_{x(k)}\hat{\tilde{f}}_{x(k)}) + \psi_x^{f2}\epsilon_x^2\mathrm{Tr}(\hat{\tilde{f}}^T_{u(k-1)}\hat{\tilde{f}}_{u(k-1)}) + \psi_u^{f2}\epsilon_u^2 = \beta_1 \quad (27)$$

$$\mathrm{Tr}(h_2) \le 2\psi_u^{F2}\epsilon_u^2 + 3\psi_x^{F2}\epsilon_x^2\mathrm{Tr}(\hat{\tilde{F}}^T_{x(k)}\hat{\tilde{F}}_{x(k)}) + 3\psi_x^{F2}\epsilon_x^2\mathrm{Tr}(\hat{\tilde{f}}^T_{u(k-1)}\hat{\tilde{f}}_{u(k-1)}) + \psi_u^{F2}\epsilon_u^2 = \beta_2 \quad (28)$$

Since $\mathrm{Tr}(\tilde{H}_k^T\tilde{H}_k) = \mathrm{Tr}(h_1) + \mathrm{Tr}(h_2)$, we have

$$\|\tilde{H}_k\|_F = \sqrt{\mathrm{Tr}(\tilde{H}_k^T\tilde{H}_k)} \le \sqrt{\beta_1 + \beta_2} \quad (29)$$

where Tr(:) represents the trace of the matrix. ■

Based on the above condition of Lemma 2, we attempt to derive a condition of singular-value relating to the appropriate column vectors in $V_{\hat{H}_k}$, which can be utilized to build a subspace that always contains s_k.

Theorem 2: If the following condition of the j-th singular-value of $\hat{H}_k$ is satisfied:

$$\Lambda^j_{\hat{H}_k} > \frac{1}{\epsilon_v}\sqrt{\beta_1 + \beta_2} \quad (30)$$

where ϵ_v is a positive scalar. The corresponding $V^j_{\hat{H}_k}$ satisfies

$$V^{jT}_{\hat{H}_k}\frac{s_k}{\|s_k\|} < \epsilon_v.$$

Proof: First, since $V_{\hat{H}_k}$ is a full rank matrix, a vector, $\hat{r}^c_k$, will always exist such that $s_k = V_{\hat{H}_k}\hat{r}^c_k$. Here, since we attempted to calculate the intersection of the possible-solution space obtained in each step, if the matrices spanning these spaces are all full rank, the dimension of the intersection space will not decrease and the estimation of the cost weight will not be completed. Therefore, it is required to select an appropriate subspace of the column vector space spanned by $V_{\hat{H}_k}$ containing vector s_h at each step and finally calculate the intersection of these vector spaces to obtain the result.

From (24), we have $Tr(s_k^T\hat{H}_k^T\hat{H}_k s_k) = Tr(s_k^T\tilde{H}_k^T\tilde{H}_k s_k)$. Furthermore, we get

$$Tr(s_k^T V_{\hat{H}_k}\Lambda^T_{\hat{H}_k}\Lambda_{\hat{H}_k}V^T_{\hat{H}_k}s_k) = Tr(\tilde{H}_k^T\tilde{H}_k s_k s_k^T). \quad (31)$$

Using $s_k = V_{\hat{H}_k}\hat{r}_k$, we obtain the inequality formulated below:

$$Tr(\Lambda^T_{\hat{H}_k}\Lambda_{\hat{H}_k}\hat{r}^o_k\hat{r}^{oT}_k) = Tr(\tilde{H}_k^T\tilde{H}_k\hat{r}^o_k\hat{r}^{oT}_k) \le Tr(\tilde{H}_k^T\tilde{H}_k) \quad (32)$$

where $\hat{r}^o_k = V^T_{\hat{H}_k}s_k/\|s_k\|$ is the unit vector of $\hat{r}_k$. Here, although we cannot determine the exact value of $\hat{r}^o_k$, each element of unit vector $\hat{r}^o_k$ can be regarded as the score evaluating the relation between vector $s_k/\|s_k\|$ and each column vector of $V_{\hat{H}_k}$ (orthogonal, parallel, otherwise). When the j-th element ($\forall j \le N$) of $\hat{r}^o_k$ satisfies $\|\hat{r}^{jo}_k\| = V^{jT}_{\hat{H}_k}s_k/\|s_k\| = 0$, $V^{jT}_{\hat{H}_k}$ has no relation with s_k, and the corresponding j-th diagonal element in matrix $\Lambda^T_{\hat{H}_k}\Lambda_{\hat{H}_k}$ will not have any effect on $Tr(\Lambda^T_{\hat{H}_k}\Lambda_{\hat{H}_k}\hat{r}^o_k\hat{r}^{oT}_k)$. Here, we design a set, $\mathbb{T}_1$, of the diagonal elements of $\Lambda_{\hat{H}_k}$ as follows:

$$\{\mathbb{T}_1 : \Lambda^j_{\hat{H}_k} \mid \hat{r}^{jo}_k = V^{jT}_{\hat{H}_k}s_k/\|s_k\| \le \epsilon_v\}$$

where the positive scalar ϵ_v is a threshold that can be appropriately selected.

Since $\Lambda^T_{\hat{H}_k}\Lambda_{\hat{H}_k}$ is a diagonal matrix, we can rewrite $Tr(\Lambda^T_{\hat{H}_k}\Lambda_{\hat{H}_k}\hat{r}^o_k\hat{r}^{oT}_k)$ as

$$Tr(\Lambda^T_{\hat{H}_k}\Lambda_{\hat{H}_k}\hat{r}^o_k\hat{r}^{oT}_k) = \sum_{j=1}^{N}\Lambda^{jT}_{\hat{H}_k}\Lambda^j_{\hat{H}_k}[\hat{r}^o_k\hat{r}^{oT}_k]^j \le Tr(\tilde{H}_k^T\tilde{H}_k) \quad (33)$$

where $[:]^j$ represents the j-th diagonal element of the matrix. All $[\hat{r}^o_k\hat{r}^{oT}_k]^j$ satisfy $0 < [\hat{r}^o_k\hat{r}^{oT}_k]^j \le 1$.

Thus, from (33), it is clear that any $\Lambda^j_{\hat{H}_k}$ satisfying $\Lambda^j_{\hat{H}_k} > \frac{1}{\epsilon_v}\sqrt{Tr(\tilde{H}_k^T\tilde{H}_k)} = \frac{1}{\epsilon_v}\sqrt{\beta_1+\beta_2}$ is the possible singular-value corresponding to the column vector $V^j_{\hat{H}_k}$, making $V^{jT}_{\hat{H}_k}s_k < \epsilon_v\|s_k\|$. ■

Theorem 2 shows that all the column vectors of $V_{\hat{H}_k}$ corresponding to the diagonal elements of $\Lambda_{\hat{H}_k}$ that satisfy the condition $\Lambda^j_{\hat{H}_k} > \frac{1}{\epsilon_v}\sqrt{\beta_1+\beta_2}$ have little correlation with the vector s_k. All the column vectors in $V_{\hat{H}_k}$ that satisfy $V^{j\mathrm{T}}_{\hat{H}_k}s_k \ge \epsilon_v\|s_k\|$ are related to the diagonal elements of $\Lambda_{\hat{H}_k}$ that satisfy the condition $\Lambda^j_{\hat{H}_k} \le \frac{1}{\epsilon_v}\sqrt{\beta_1+\beta_2}$.

Therefore, we select the n_v column vectors in $V_{\hat{H}_k}$ corresponding to the diagonal elements of $\Lambda_{\hat{H}_k}$ that satisfy the condition $\Lambda^j_{\hat{H}_k} \le \frac{1}{\epsilon_v}\sqrt{\beta_1+\beta_2}$, which makes it possible for $V^j_{\hat{H}_k}$ to satisfy $V^{j\mathrm{T}}_{\hat{H}_k}s_k \ge \epsilon_v\|s_k\|$. We can then construct a new matrix $T_{\hat{H}_k}$ with n_v columns and find a vector $\hat{r}_k$ such that $T_{\hat{H}_k}$ satisfies the following inequality:

$$\|s_k - T_{\hat{H}_k}\hat{r}_k\| \le \sqrt{1-\epsilon_v^2 n_v}\|s_k\| \quad (34)$$

where $\epsilon_v^2 n_v \le 1$ according to the conditions $\|T^T_{\hat{H}_k}s_k\| \ge \epsilon_v^2 n_v\|s_k\|$ and $\|T^T_{\hat{H}_k}s_k\| \le \|V^{\mathrm{T}}_{\hat{H}_k}s_k\| = \|s_k\|$.

In practice, the degree of correlation between s_k and the column vectors in $T_{\hat{H}_k}$ can be estimated using the threshold of $\sqrt{1-\epsilon_v^2 n_v}\|s_k\|$, allowing us to choose suitable values for ϵ_v and n_v to ensure a satisfactory degree of correlation. Therefore, we can utilize $T_{\hat{H}_k}$ to replace Θ_k in the calculation.

2) Method of Tackling the Effect of Noise in $\hat{\Phi}_{h(k)}$: In the noisy case, due to the noise on $\hat{\tilde{M}}_{h:k}$ in step k, (12) becomes

$$\hat{\tilde{M}}_{h:k}s_k = s_h + e^r_{Mk} \quad (35)$$

where $e^r_{Mk} = \tilde{\tilde{M}}_{h:k}s_k$. Furthermore, by representing s_k using Theorem 2's result, we have

$$e^r_{Mk} \approx \tilde{\tilde{M}}_{h:k}T_{\hat{H}_k}\hat{r}_k \quad (36)$$

and the calculation of $\Phi_{h(i)}$ in the noisy case can be formulated as follows:

$$\hat{\Phi}_{h(k)}\hat{r}_k = \hat{\tilde{M}}_{h:k}T_{\hat{H}_k}\hat{r}_k \approx s_h + e^r_{Mk}. \quad (37)$$

Therefore, after performing SVD in the noise case (37), we

have

$$\hat{\Phi}_{h(k)} = W_{\hat{\Phi}_{h(k)}} \Lambda_{\hat{\Phi}_{h(k)}} V^T_{\hat{\Phi}_{h(k)}}.$$

Furthermore, from the result of Theorem 2, the formula for calculating of $\Phi_{h(i)}$ can be expressed as

$$\Phi'_{h(k)} \hat{r}_k = \bar{M}_{h:k} T_{\hat{H}_k} \hat{r}_k \approx s_h. \tag{38}$$

It is clear that the remaining effect originates from the noise term, $\tilde{\bar{M}}_{h:k} = \bar{M}_{h:k} - \hat{\bar{M}}_{h:k}$.

We propose a lemma to clarify this effect below.

Lemma 3: The Frobenius norm of matrix $\|\tilde{\bar{M}}_{h:k}\|_F$ is bounded and satisfies the following condition:

$$\|\tilde{\bar{M}}_{h:k}\|_F \le \gamma_k \tag{39}$$

where $\gamma_k = (\|\hat{\bar{M}}_k\|_F + b_k)\,\gamma_{k-1} + b_k\,\|\hat{\bar{M}}_{h:k-1}\|_F$, $b_k = \sqrt{Tr(\tilde{\bar{f}}^T_{x(k)}\tilde{\bar{f}}_{x(k)} + \tilde{\bar{F}}^T_{x(k)}\tilde{\bar{F}}_{x(k)})} = \sqrt{\psi^f_x \epsilon_x + \psi^F_x \epsilon_x}$ and $\gamma_1 = \|\tilde{M}_1\|_F = b_k$

Proof: First, from the definition of M_k, it obtains

$$\tilde{M}_k = \begin{bmatrix} \tilde{\bar{f}}_{x(k)} & \tilde{\bar{F}}_{x(k)} \\ 0_{n_f \times n} & 0_{n_f} \end{bmatrix}. \tag{40}$$

From this equation, we have

$$\|\tilde{M}_k\|_F \le \sqrt{Tr(\tilde{\bar{f}}^T_{x(k)}\tilde{\bar{f}}_{x(k)} + \tilde{\bar{F}}^T_{x(k)}\tilde{\bar{F}}_{x(k)})} = b_k.$$

Since $\tilde{\bar{M}}_{h:k} = \bar{M}_k \bar{M}_{h:k-1} - \hat{\bar{M}}_k \hat{\bar{M}}_{h:k-1}$,

$$\begin{aligned} \|\tilde{\bar{M}}_{h:k}\|_F &= \|\bar{M}_k \tilde{\bar{M}}_{h:k-1} + \tilde{\bar{M}}_k \hat{\bar{M}}_{h:k-1}\|_F \\ &\le \|\bar{M}_k\|_F \|\tilde{\bar{M}}_{h:k-1}\|_F + \|\tilde{\bar{M}}_k\|_F \|\hat{\bar{M}}_{h:k-1}\|_F \end{aligned} \tag{41}$$

where $\|\bar{M}_k\|_F \le \|\hat{\bar{M}}_k\|_F + \|\tilde{\bar{M}}_k\|_F$.

Then, we have

$$\|\tilde{\bar{M}}_{h:k}\|_F \le \gamma_k = (\|\hat{\bar{M}}_k\|_F + b_k)\gamma_{k-1} + b_k \|\hat{\bar{M}}_{h:k-1}\|_F. \tag{42}$$

■

The following part is proposed to clarify the condition of singular-value of $\hat{\Phi}_{h(k)}$, whose related column vector in $W_{\hat{\Phi}_{h(k)}}$ is the possible null space of $\hat{\Phi}'_{h(k)}$.

Theorem 3: The singular-value of $\hat{\Phi}_{h(k)}$, corresponding to

$$\|Z^T_{\Phi'_{h(k)}} W^j_{\hat{\Phi}_{h(k)}}\| \ge \epsilon_a$$

lies in the set that

$$\Lambda^j_{\hat{\Phi}_{h(k)}} \le \frac{1}{\epsilon_a} \sqrt{\gamma_k} N \tag{43}$$

where $Z_{\Phi'_{h(k)}}$ denotes the null space of the column space of $\Phi'_{h(k)}$ as $Z_{\Phi'_{h(k)}} = \text{null}(\Phi'_{h(k)})$ and ϵ_a is a positive scalar.

Here, $\|Z^T_{\Phi'_{h(k)}} W^j_{\hat{\Phi}_{h(k)}}\|$ indicates that the projection of the j-th column vector of $W_{\hat{\Phi}_{h(k)}}$ on the null space of $\Phi'_{h(k)}$ is larger than ϵ_a.

Proof: First, from (37) and (38), we have

$$Z^T_{\Phi'_{h(k)}} \hat{\Phi}_{h(k)} = Z^T_{\Phi'_{h(k)}} \tilde{\bar{M}}_{h:k} T_{\hat{H}_k}. \tag{44}$$

By performing SVD on $\hat{\Phi}_{h(k)}$, we have $\hat{\Phi}_{h(k)} = W_{\hat{\Phi}_{h(k)}} \Lambda_{\hat{\Phi}_{h(k)}} V^T_{\hat{\Phi}_{h(k)}}$. Thus, from (44), we have

$$\begin{aligned} &Tr(\Lambda^T_{\hat{\Phi}_{h(k)}} W^T_{\hat{\Phi}_{h(k)}} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} W_{\hat{\Phi}_{h(k)}} \Lambda_{\hat{\Phi}_{h(k)}}) \\ &\quad = Tr(T^T_{\hat{H}_k} \tilde{\bar{M}}^T_{h:k} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} \tilde{\bar{M}}_{h:k} T_{\hat{H}_k}) \end{aligned} \tag{45}$$

where $\hat{r}^o_k = \hat{r}_k / \|\hat{r}_k\|$ is a unit vector. Since $p_{\hat{\Phi}_{h(k)}} = V^T_{\hat{\Phi}_{h(k)}} \hat{r}^o_k$ is a unit vector, the left hand side of this equation satisfies

$$\begin{aligned} &Tr(\Lambda^T_{\hat{\Phi}_{h(k)}} W^T_{\hat{\Phi}_{h(k)}} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} W_{\hat{\Phi}_{h(k)}} \Lambda_{\hat{\Phi}_{h(k)}}) \\ &\quad \le Tr(\Lambda^T_{\hat{\Phi}_{h(k)}} \Lambda_{\hat{\Phi}_{h(k)}} W^T_{\hat{\Phi}_{h(k)}} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} W_{\hat{\Phi}_{h(k)}}). \end{aligned} \tag{46}$$

Each column vector in $Z_{\Phi'_{h(k)}}$ and $W_{\hat{\Phi}_{h(k)}}$ can be represented as $Z_{\Phi'_{h(k)}} = [n_1, \ldots, n_{j_s}]$ and $W_{\hat{\Phi}_{h(k)}} = [W^1_{\hat{\Phi}_{h(k)}}, \ldots, W^N_{\hat{\Phi}_{h(k)}}]$, it is clear that j-th diagonal element of matrix $\Lambda^T_{\hat{\Phi}_{h(k)}} W^T_{\hat{\Phi}_{h(k)}} \times Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} W_{\hat{\Phi}_{h(k)}} \Lambda_{\hat{\Phi}_{h(k)}}$ can be calculated as $((W^{jT}_{\hat{\Phi}_{h(k)}} n_1)^2 + \cdots + (W^{jT}_{\hat{\Phi}_{h(k)}} n_{j_s})^2)\Lambda^{j2}_{\hat{\Phi}_{h(k)}}$ where $\Lambda^j_{\hat{\Phi}_{h(k)}}$ represents the j-th diagonal element in $\Lambda_{\hat{\Phi}_{h(k)}}$. Here, value of $(W^{jT}_{\hat{\Phi}_{h(k)}} n_1)^2 + \cdots + (W^{jT}_{\hat{\Phi}_{h(k)}} n_{j_s})^2$ evaluates the relation between column vector $W^j_{\hat{\Phi}_{h(k)}}$ and null space of $\hat{\Phi}_{h(k)}$. When $(W^{jT}_{\hat{\Phi}_{h(k)}} n_1)^2 + \cdots + (W^{jT}_{\hat{\Phi}_{h(k)}} n_{j_s})^2 = 0$, it represents that $W^j_{\hat{\Phi}_{h(k)}}$ is completely not in the null space of $\Phi'_{h(i)}$ and $\Lambda^j_{\hat{\Phi}_{h(k)}}$ have no effects on the value of $Tr(\Lambda^T_{\hat{\Phi}_{h(k)}} \Lambda_{\hat{\Phi}_{h(k)}} \times W^T_{\hat{\Phi}_{h(k)}} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} W_{\hat{\Phi}_{h(k)}})$.

Moreover, for the left hand side of (45), we have

$$\begin{aligned} &Tr(T^T_{\hat{H}_k} \tilde{\bar{M}}^T_{h:k} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} \tilde{\bar{M}}_{h:k} T_{\hat{H}_k}) \\ &\quad \le Tr(T_{\hat{H}_k} T^T_{\hat{H}_k} \tilde{\bar{M}}^T_{h:k} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} \tilde{\bar{M}}_{h:k}) \\ &\quad \le \|\tilde{\bar{M}}_{h:k}\|^2_F N^2 = \gamma_k N^2. \end{aligned} \tag{47}$$

Therefore, the right-hand side of (45) satisfies

$$\begin{aligned} &Tr(\Lambda^T_{\hat{\Phi}_{h(k)}} W^T_{\hat{\Phi}_{h(k)}} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} W_{\hat{\Phi}_{h(k)}} \Lambda_{\hat{\Phi}_{h(k)}}) \\ &\quad \le \gamma_k N^2 \end{aligned} \tag{48}$$

which indicates that

$$\sum_j \Lambda^{j2}_{\hat{\Phi}_{h(k)}} [W^T_{\hat{\Phi}_{h(k)}} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} W_{\hat{\Phi}_{h(k)}}]^j \le \gamma_k N^2 \tag{49}$$

where maximum value of each $[W^T_{\hat{\Phi}_{h(k)}} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} W_{\hat{\Phi}_{h(k)}}]^j$ is one and

$$\sum_j [W^T_{\hat{\Phi}_{h(k)}} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} W_{\hat{\Phi}_{h(k)}}]^j \le N - 1.$$

Here, from (49), when

$$[W^T_{\hat{\Phi}_{h(k)}} Z_{\Phi'_{h(k)}} Z^T_{\Phi'_{h(k)}} W_{\hat{\Phi}_{h(k)}}]^j \ge \epsilon^2_a$$

we have

$$\Lambda^j_{\hat{\Phi}_{h(k)}} \le \frac{1}{\epsilon_a} \sqrt{\gamma_k} N.$$

■

With the selection of high $\epsilon_a \le 1$, vector $W^j_{\hat{\Phi}_{h(k)}}$ exhibits a

high correlation with the null space of $\Phi'_{h(k)}$.

Here, we utilize data from multiple observations $\{\hat{x},\hat{u}\}^{i_s}$, where $i_s = 1:\Xi_s$ denotes the multiple samples of data for one original set of system states and control input pairs $\{x,u\}$. Online measurement can be conducted using different sensors.

From Theorem 3, we select $W^{j,i_s}_{\hat{\Phi}_{h(k)}}\ \forall j \le N$ corresponding to $\Lambda^{j,i_s}_{\hat{\Phi}_{h(k)}} \le \frac{1}{\epsilon_a}\sqrt{\gamma_k}N$ for constructing subspace $T^{i_s}_{\hat{\Phi}_{h(k)}}$ with each observations. Here, the superscript i_s denotes that the matrix is obtained using the data of $\{\hat{x},\hat{u}\}^{i_s}$.

Thus, to replace the calculation of $\text{null}(\Phi_{h(k)})$ in Algorithm 1 to tackle the noises, we select $T^{i_s}_{\hat{\Phi}_{h(k)}}$ using

$$\Gamma_{T^o_{\hat{\Phi}_{h(k)}}} = \Gamma_{T^1_{\hat{\Phi}_{h(k)}}} \cap \cdots \cap \Gamma_{T^{i_s}_{\hat{\Phi}_{h(k)}}} \tag{50}$$

with calculating

$$T^o_{\hat{\Phi}_{h(k)}} = \text{null}\left(\left[\text{null}(T^1_{\hat{\Phi}_{h(k)}})^T \quad \cdots \quad \text{null}(T^{i_s}_{\hat{\Phi}_{h(k)}})^T\right]\right) \tag{51}$$

where $\Gamma_{T^{i_s}_{\hat{\Phi}_{h(k)}}}$ represents the column vector space of matrix $T^1_{\hat{\Phi}_{h(k)}}$.

The total calculation in noisy case is shown in Algorithm 2.

Algorithm 2 Online Implementation (Noise Case)

Input: $\{x_i,u_i\}^{i_s}, \|q\|$

Output: s_h

Initialization :

1: Compute $H^{i_s}_h$ and $\Phi^{i_s}_{h(h)}$ using $\{x,u\}^{i_s}$ and reconstruct $T^1_{\hat{\Phi}_{h(k)}},\ldots, T^{i_s}_{\hat{\Phi}_{h(k)}}$.

2: Initialize matrix $\Omega_s = T^o_{\hat{\Phi}_{h(h)}}$ following (51) which represents the intersection of possible-solution spaces.

LOOP Process

3: **for** $i = h+1$ to K **do**

4: **if** $\bar{u}_i \in \text{int}(\mathcal{U})$ **then**

5: **if** $\bar{u}_{i-1} \notin \text{int}(\mathcal{U})$ **then**

6: Calculate $T^o_{\hat{\Phi}_{h(i)}}$ and reinitialize $\Omega_s = T^o_{\hat{\Phi}_{h(i)}}$

7: **Continue**.

8: **end if**

9: Calculate $T^o_{\hat{\Phi}_{h(i)}}$, $\text{null}(T^o_{\hat{\Phi}_{h(i)}})$ and $\text{null}(\Omega_s)$.

10: Calculate $\Gamma_{\Omega_s} \cap \Gamma_{T^o_{\hat{\Phi}_{h(i)}}}$ by constructing the matrix $Y_{h(i)} = [\text{null}(T^o_{\hat{\Phi}_{h(i)}})^T, \text{null}(\Omega_s)^T]^T$ and compute the null space of $Y_{h(i)}$. Here, Γ_{Ω_s} denotes the vector space of Ω_s.

11: **if** $\text{rank}(Y_{h(i)}) < N-1$ **then**

12: Update $\Omega_s = \text{null}(Y_{h(i)})$

13: **end if**

14: **if** $(\text{rank}(Y_{h(i)}) = N-1)$ **then**

15: Ω_s is decreased to a vector with $\Omega_s = \text{null}(Y_{h(i)})$.

16: Get the unique solution following (19).

17: **end if**

18: **end if**

19: **end for**

20: **return** s_h

3) Calculation With Control Constraints: When there exist control constraints and noise in data, we replace the calculation procedure for the control constraints with

$$\begin{cases} \Gamma_{\Omega_{h:i}} = \Gamma_{\Omega_{h:i-1}} \cap \Gamma_{T^o_{\hat{\Phi}_{h()}}}, & \bar{u}_i \in \text{int}(\mathcal{U})\ \&\ \bar{u}_{i-1} \in \text{int}(\mathcal{U}) \\ \textit{Reinitialize}\quad \Gamma_{\Omega_{h:i}}, & \bar{u}_i \in \text{int}(\mathcal{U})\ \&\ \bar{u}_{i-1} \notin \text{int}(\mathcal{U}) \\ \textit{Skip the Step}, & \text{Otherwise.} \end{cases} \tag{52}$$

where $\bar{u}_i$ represents the mean value of the control input of the multiple data series $\{x,u\}^{i_s}$.

V. Simulation Examples

We perform several simulations in different cases to verify the effectiveness of our method.

To ensure the reproducibility of the calculation process, we reinitialize a new IOC calculation cycle in all simulations after obtaining a result in step i. This also allows us to verify the performance of the algorithm under different initial conditions, as the initial state at the start of each new calculation cycle of SIOC will be different from the previous one.

A. Simulation 1: Comparison With [16]

This section illustrates our method with simulation in the settings of nonlinear system. The system dynamics are

$$x_{k+1} = f(x_k) + Bu_k \tag{53}$$

where $x_k \in \mathbb{R}^3$, $u_k \in \mathbb{R}^2$, $f(x_k) \in \mathbb{R}^{3\times 1}$, and $B \in \mathbb{R}^{3\times 2}$. We select $f(x_k)$ and B as

$$f(x_k) = \begin{bmatrix} \sin(x_k^1) \\ \sin(x_k^2) \\ \sin(x_k^3) \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0.5 & -0.5 \end{bmatrix}$$

where x_k^j denotes the j-th element in vector x_k. The initial value of x_k, denoted as x_0, is selected as $[0.38, -0.02, 0.19]^T$.

The cost function selected in the simulations is

$$V(x,t) = \sum_{k=0}^{\infty} \frac{1}{2}(x_k^T Q_x x_k + u_k^T Q_u u_k) \tag{54}$$

where Q_x is a diagonal matrix with its diagonal elements as vector $q_x = [0.0006, 0.0002, 0.002]^T$, and Q_u is also a diagonal matrix with its diagonal elements as vector $q_u = [1,1]^T$. Here, we suppose that the norm of vector $q = [q_x^T, q_u^T]^T$ selected as 1.4142 is known prior.

Here, the diagonal matrix $\frac{\partial f(x_k)}{x_k}$ with $\cos(x_k^i) \forall 1 \le i \le 3$ on its diagonal position satisfies the invertible assumption.

We recover the cost weights in (56) and compared the simulation results with the results of [16] according to two aspects: the number of time steps required to recover the cost weight vectors in each cycle (Figs. 1 and 2) and the recovery error of the cost weights (Figs. 3 and 4).

Fig. 1 shows the result of the step performed in our method, while Fig. 2 shows the result in [16]. The horizontal axes in both figures represent the total steps during the simulations. The dotted blue line is the end of each cycle while its height related to the left vertical axis is the number of steps spent in each IOC cycle. The red lines in these two figures represent the dimension variation of the intersection of the possible-solution spaces whose value relates to the right vertical axis.

The result in Fig. 1 shows that the dimension of $\Gamma_{\Omega_{h:i}}$

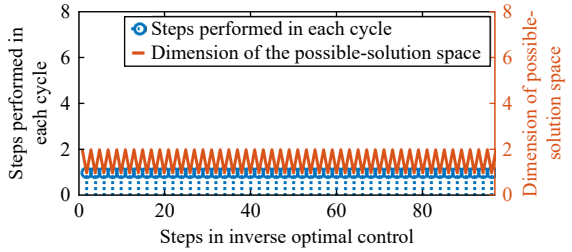


Fig. 1. Steps performed in our method for Simulation 1 are shown here. The orange line shows the variation of the dimension of the possible-solution space during the IOC cycles. When the dimension decreases to one, it indicates that the cost weights vector has been successfully calculated, and a new IOC cycle starts. The blue cycles show the total number of steps taken in one IOC cycle.

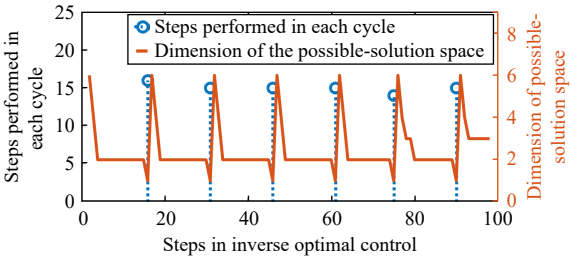


Fig. 2. Steps performed in method of [16] for Simulation 1 are shown here. The meaning of the lines is the same as in Fig. 1.

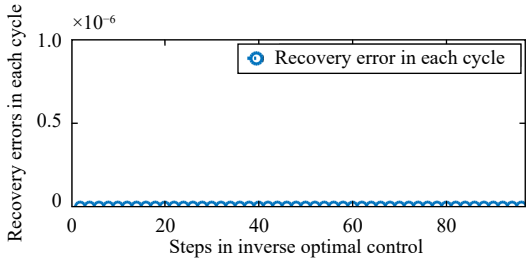


Fig. 3. Recovery errors of Simulation 1 by our method are shown here. The blue circles represent the recovery error in each IOC cycle.

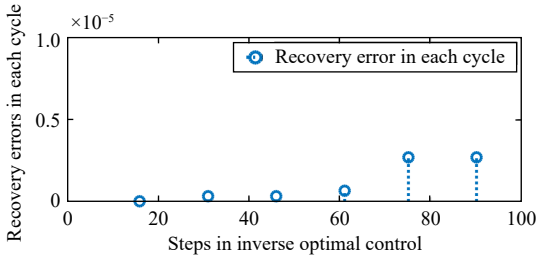


Fig. 4. Recovery errors of Simulation 1 by [16] are shown here. The meaning of the lines is the same as in Fig. 3.

decreases in every step in each calculation cycle and that the maximum number of steps in one cycle is 1. Compared with our result, the dimension of the possible-solution spaces in [16], does not decrease continually. As a result, the number of steps of one cycle for [16] in Fig. 2 is always larger than that in our method.

Figs. 3 and 4 show the recovery error of both methods calculated by $e = \|\hat{q} - q\|$ where $\hat{q}$ denotes the estimation vector of q. From these two figures, it is clear that the estimation error of our method is minor. Therefore, our proposed method can effectively improve the calculation speed while preserving the recovery accuracy of IOC.

B. Simulation 2: When Jacobian is not Invertible

In Simulation 2,

$$x_{k+1} = Ax_k + Bu_k \tag{55}$$

where A and B are selected as $A = \begin{bmatrix} 0.9 & 1.8 & 0 \\ 0.13 & 0.26 & 0 \\ 0.38 & 0.76 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 0.0284 & 0.0142 \\ 0.0020 & 0.0010 \\ 0.0056 & -0.0028 \end{bmatrix}$.

Here, the system matrix, A, does not satisfy the Jacobian's invertibility assumption, but it satisfies the Jacobian's positivity assumption. In this case, [16] cannot be applied. The cost function selected in the simulation is

$$V(x,t) = \sum_{k=0}^{\infty} \frac{1}{2}(x_k^T Q_x x_k + u_k^T Q_u u_k) \tag{56}$$

where Q_x is a diagonal matrix with its diagonal elements as vector $q_x = [1,4,2]^T$, and Q_u is a diagonal matrix with its diagonal elements as vector $q_u = [3,1]^T$. Here, we suppose that the norm of vector $q = [1,4,2,3,1]^T$ selected as 5.5678 is known prior.

Moreover, the control-constraint problem is also considered: $\mathcal{U} \triangleq \{u_{ki} \geq -0.2\ \forall i\}$ where u_{ki} denotes the i-th element of u_k

Fig. 5 shows the calculation steps in every cycle, which is 1. Here, similarly to the result obtained in Simulation 1, the variation in the dimension of the possible-solution spaces verifies Theorem 1, and the variation in the dimension of the possible-solution spaces after the activation of the control constraints shows that the proposed algorithm is effective for handling the control constraint problem.

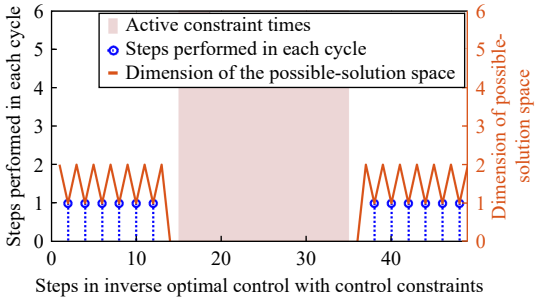


Fig. 5. Steps performed in our method for Simulation 2 are shown here. The meaning of the lines is the same as in Fig. 1.

Fig. 6 shows the estimation error in this simulation. Even when A is rank deficient and control constraints exist, the errors in all cycles are still extremely small, which shows that the proposed method can effectively recover the required cost weights with considerable accuracy.

Fig. 7 shows a comparison between the original trajectories of system states and control inputs and the trajectories gener-

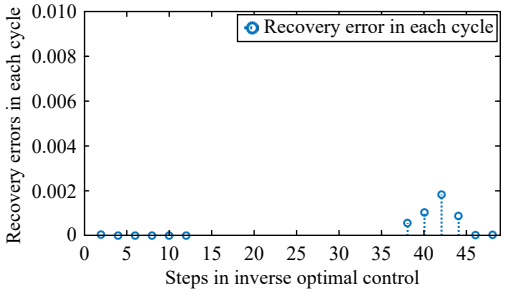


Fig. 6. Recovery errors of Simulation 2 by our method are shown here. The meaning of the lines is the same as in Fig. 3.

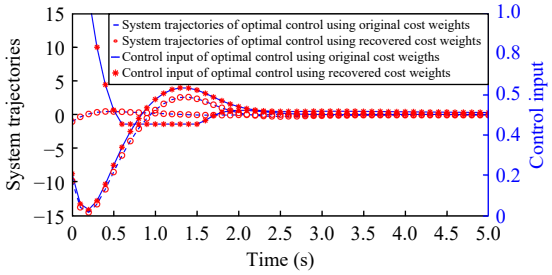


Fig. 7. Comparison of original trajectories (system states and control inputs) with the trajectories using recovered cost weights.

ated using the recovered cost weights. The red and blue lines in the figure are identical, indicating that the recovered cost weights can be used to replicate the original optimal trajectories. This demonstrates the potential of our method to be further applied in demonstration tasks.

C. Simulation 3: Verification of SIOC Under Different Conditions

In Simulation 3, we conducted a comprehensive evaluation of the proposed SIOC method under different initial conditions and system dynamics. Specifically, we simulated 1000 different linear systems with randomly generated initial states (x_0) and system matrices (A and B). All system settings A and B used in our simulations were randomly generated using the MATLAB function $rand(3,3)$ for A and $rand(3,2)$ for B and the initial states were generated using $10\times rand(3,1)$. For each system, we applied the SIOC method and evaluated its performance in recovering the cost weights. Fig. 8 shows the recovery errors of the SIOC method performed with 1000 different system dynamics and initial states. The results demonstrate that the recovery errors are consistently very small (average error of 4.0312×10^{-15}), indicating the effectiveness of our method.

D. Simulation 4: When There Exists Noise

Due to the decaying property of the state sequence and the stationary property of the measurement noise, the signal-to-noise ratio (SNR) is defined as

$$SNR = 10\log\frac{Tr[cov(\begin{bmatrix}x_{t_F}\\u_{t_F}\end{bmatrix})]}{Tr[cov(\begin{bmatrix}\tilde{x}_{t_F}\\\tilde{u}_{t_F}\end{bmatrix})]} \tag{57}$$

where t_F represents the time index at which the IOC calculation is terminated. To verify the effectiveness of our method, we performed simulations at several noise levels. In simulations at each noise level, we performed simulation 100 times with different system dynamics. Here, we performed simulations on linear systems with $A\in R^{3\times3}$ and $B\in R^{3\times2}$ randomly selected using matlab function *rand*(). The initial state is randomly selected, and noises at different SNRs are generated following the standard Gaussian distribution.

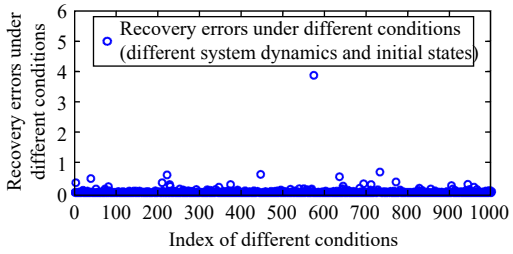


Fig. 8. Verification of SIOC under different conditions (1000 different system dynamics and initial conditions).

Fig. 9 shows the comparison results between our method and that in [16] with $SNR=$ values of 20, 65, 94, 191, and 238, respectively. There are 100 results in comparison study of each SNR's settings. In the simulations, the estimation error is evaluated by relative-estimation error, defined as $\frac{\|\hat{s}_h - s_h\|}{\|s_h\|}$.

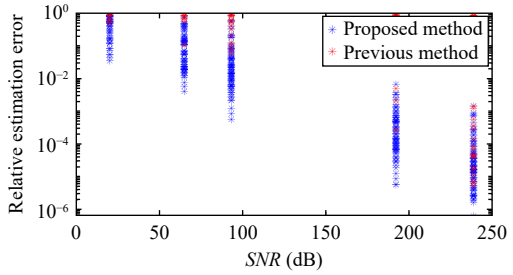


Fig. 9. Comparison of our method (blue points) with the method in [16] (red points) under noisy conditions. 1) SNR = 20, 65, 94, 191, 238; 2) There are 100 simulation samples in each selection of SNR.

From Fig. 9, it is clear that in our method, the relative-estimation errors decrease along with the increase in the SNR, indicating that this noise-tackling method can be utilized in the noise-free case. Moreover, a comparative study with the method in [16], revealed that our SIOC method considering noises is more robust in each setting of SNR.

Therefore, from Simulations 1 and 2, it is verified that the proposed method can solve the online IOC problem even for the systems that are not applicable in [16]. Our method effectively improves the calculation speed of IOC. From Simulation 4, it is evident that the proposed method can effectively tackle the noise problem, which is not considered in the previous online IOC study.

VI. Discussion

A. Computation Complexity

The computational complexity of our method in one step is $O(3(n+n_f)^3+(n+n_f)n'^2+(n+n_f)(n+n_f+n')^2)$ in the noise-

free case, where n' is the dimension of the possible-solution space and it decreases as the step number increases. The computational complexity of our method does not contain horizon K, it is typically less than that in [15], wherein the horizon K was contained in the computation-complexity calculation. Conversely, the computational complexity of [16] is $O((n+n_f)^3+m(n+n_f)^2)$ in one step.

We conducted simulations using 1000 different system settings and initial states, and compared the calculation time of our method with that of the previous method. The results, presented in Fig. 10, show that while our method has a slightly longer calculation time in one step than the previous method [16] on average, it is more stable, with less variability in the calculation time across the different system settings and initial states. Notably, our method requires fewer computational steps, and therefore, the choice of the method with lower total computational complexity may depend on the specific case.

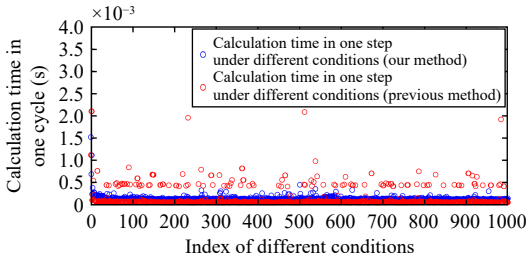


Fig. 10. Calculation time in one step under different conditions (1000 different system dynamics and initial conditions): Our method versus previous method [16].

B. Importance of the Sequential Calculation of the IOC Method

This paper proposed a sequential inverse optimal control IOC method that derives the conditions of the possible-solution space and tackling method for noisy data.

The first advantage of SIOC is that it saves computational time. This is a significant advantage for programs that require real time computing.

Secondly, there is no assurance that the cost weights will remain constant across all the previously well-selected feature functions while studying the complex dynamic movements. In [31], the authors suggested a method for calculating the multiphase cost weights based on window shifts, and when using this method to study complex motions, the length of the window must be minimized to recover the cost weight with multiphase changes in high precision. In this case, our SIOC method can be used to reduce the length of the window.

Additionally, the high calculation speed of the SIOC strategy helps to lessen the impact of noise. Notably, achieving noise reduction in the analysis of the observations from different steps is challenging, and this process must be completed for the calculation of each different IOC method. The impact of noise increases with the accumulated data step by step. The method proposed introduces a calculation method for the IOC with a minimum number of steps, and this high calculation speed helps to reduce the effect of noisy data on the final cost-weight estimates.

C. Future Work

Although the problem of sequential IOC has been solved in this study considering the calculation speed and noisy data, the algorithm still requires improvement in the following areas.

1) It is possible to further improve the noisy-tackling ability in the sequential IOC method. Since this method has a high convergence speed, we can start an IOC calculation cycle at each step and obtain time-series groups of solutions. By theoretically analyzing the result in each calculation cycle and considering the effect of the multiphase cost weight, it may be possible to further enhance the precision of the estimation result. We will also address the matter of special system dynamics in real-world application examples where the system states exhibit insensitivity to changes in cost weights. This insensitivity magnifies the effect of noise on the accuracy of the IOC calculation in noisy scenarios.

2) It is also required to discuss the selection of the feature function. To analyze the complex nature behavior, the selection of the feature function will highly affect the approximation results. Additionally, the aforementioned problem of multiphase cost weights is highly related to the selection of the feature function.

3) In addition, while the SIOC method proposed in this study solves the online recovery of the cost function, further investigation could be undertaken to develop effective and efficient algorithms for online tuning of the control input, especially crucial in control problems where minimizing a specific cost function is challenging or selecting suitable cost weights poses difficulties.

VII. Conclusion

A sequential method for discrete-time IOC is presented in this paper to realize the online estimation of cost weights for either finite or infinite horizon optimal control in cases with significant data noise. This method calculates the possible-solution space of the IOC and sequentially calculates the intersection of all solution spaces in each step. The conditions for the decrease in dimension of the intersection space in the noise-free case are clarified first. When the dimension of the possible-solution space decreases to one, the remaining vector in the intersection space is the required solution of the cost weight of the IOC. In the noise case, an adjusted calculation of the possible-solution space is proposed based on the analysis of the noise effect. Finally, simulation results illustrate that the sequential IOC algorithm is effective, has a high convergence speed, and can sequentially tackle the problem of noisy data. More theoretical studies on the influences of the feature function selection on the solution spaces should be conducted for practical applications.

Appendix

Proof of Theorem 1

Proof: If $\Gamma_{\Omega_{h:i-1}} \subseteq \Gamma_{\Phi_{h(i)}}$, we have $\text{null}(\Gamma_{\Omega_{h:i-1}}) \supseteq \text{null}(\Gamma_{\Phi_{h(i)}})$, indicating that a full rank matrix $\xi = [\xi_h \ \cdots \ \xi_j \ \cdots \ \xi_{i-1}] \in R^{N_{z(i)} \times \sum_{j=h}^{i} N_{z(j)}}$, exists, which satisfies the equation below, where $N_{z(j)} \ \forall h \le j \le i$ denotes the dimension of the null space of $\Gamma_{\Phi_{h(j)}}$:

$$\xi\bar{\Omega}_{h:i-1} = \text{null}(\Phi_{h(i)})^T \tag{58}$$

where $\bar{\Omega}_{h:i-1} = \begin{bmatrix} \text{null}(\Phi_{h(h)})^T \\ \text{null}(\Phi_{h(h+1)})^T \\ \vdots \\ \text{null}(\Phi_{h(i-1)})^T \end{bmatrix}$.

It also means that

$$\xi\bar{\Omega}_{h:i-1}\Phi_{h(i)} = 0_{N_z(i)\times N_p(i)} \tag{59}$$

where $N_p(i) \le N$ is the dimension of the vector space of $\Phi_{h(i)}$. $0_{N_z(i)\times N_p(i)} \in R^{N_z(i)\times N_p(i)}$ represents the zero matrix. Row vector space of $\xi\bar{\Omega}_{h:i-1}$ is an orthogonal complement to the column vector space of $\Phi_{h(i)}$.

Here, under Assumption 1-1) and from the definition of $\bar{M}_{h:j}$, we know that $\bar{M}_{h:j} \quad \forall h \le j \le i-1$ is invertible, it obtains

$$H_j\bar{M}_{h:j}^{-1}\Phi_{h(j)} = 0_{N_z(i)\times N_p(i)}.$$

Since $\Phi_{h(j)} = \bar{M}_{h:j}\Theta_j$ and row vector space of H_j is the orthogonal complement vector space of Θ_j, it can get that column vector space of matrix $\bar{M}_{h:j}^{-T}H_j^T$ is the null space of column vector space of $\Phi_{h(j)}$. Equation (59) can be satisfied if and only if there exists a nonzero matrix ξ_o satisfies

$$\xi_o\bar{\Omega}'_{h:i-1}\Theta_i = 0_{N_z(i)\times N_p(i)} \tag{60}$$

where

$$\bar{\Omega}'_{h:i-1} = \begin{bmatrix} H_h\bar{M}_{h+1:i} \\ \vdots \\ H_j\bar{M}_{j+1:i} \\ \vdots \\ H_{i-1}M_i \end{bmatrix}$$

and

$$\xi_o\bar{\Omega}'_{h:i-1} = \xi\bar{\Omega}_{h:i-1}\bar{M}_{h+1:i}.$$

Since $\bar{M}_{h+1:i}$ is full rank, we have

$$rank(\xi_o\bar{\Omega}'_{h:i-1}) = rank(\xi\bar{\Omega}_{h:i-1}\bar{M}_{h+1:i}) = rank(\xi\bar{\Omega}_{h:i-1})$$
$$rank(\Theta_i) = rank(\Phi_{h(i)}).$$

Due to that $\xi\bar{\Omega}_{h:i-1}$ is an orthogonal complement to the column vector space of $\Phi_{h(i)}$ that

$$rank(\xi\bar{\Omega}_{h:i-1}) + rank(\Phi_{h(i)}) = N$$

we have

$$rank(\xi_o\bar{\Omega}'_{h:i-1}) + rank(\Theta_i) = N. \tag{61}$$

From (60) and (61), it is known that the row vector space of $\xi_o\bar{\Omega}'_{h:i-1}$ is an orthogonal complement to the vector space of Θ_i, meaning that matrix ξ_s, exists, which satisfies

$$[\xi_s \quad I_i]\begin{bmatrix} H_h\bar{M}_{h+1:i} \\ \vdots \\ H_j\bar{M}_{j+1:i} \\ \vdots \\ H_{i-1}M_i \\ H_i \end{bmatrix} = 0_{N_z(i)\times N} \tag{62}$$

where $I_i \in R^{N_z(i)\times N_z(i)}$ is a unit matrix.

Equation (62) also means that dimension of the null space of the column vector space of $\begin{bmatrix} H_h\bar{M}_{h+1:i} \\ \vdots \\ H_j\bar{M}_{j+1:i} \\ \vdots \\ H_{i-1}M_i \\ H_i \end{bmatrix}$ should be at least $N_z(i)$.

Here, H_i can be represented as

$$H_i = \begin{bmatrix} \mathbb{H}_{(i)1} & \mathbb{H}_{(i)2} \\ \mathbb{H}_{(i)3} & \mathbb{H}_{(i)4} \end{bmatrix} = \begin{bmatrix} \bar{f}_{u(i-1)}^T\bar{f}_{x(i)}^T & \bar{F}_{u(i-1)}^T + \bar{f}_{u(i-1)}^T\bar{F}_{x(i)}^T \\ \bar{f}_{u(i)}^T & \bar{F}_{u(i)}^T \end{bmatrix} \tag{63}$$

and from the definition of H_j and $\bar{M}_{j+1:i}$, $H_j\bar{M}_{j+1:i} \quad \forall h \le j \le i-1$ can be represented as

$$H_j\bar{M}_{j+1:i} = \begin{bmatrix} \mathbb{H}_{(j)1} & \mathbb{H}_{(j)2} \\ \mathbb{H}_{(j)3} & \mathbb{H}_{(j)4} \end{bmatrix} \tag{64}$$

where

$$\mathbb{H}_{(j)1} = \bar{f}_{u(j-1)}^T\bar{f}_{x(j)}^T\cdots\bar{f}_{x(i)}^T$$

$$\mathbb{H}_{(j)2} = \bar{F}_{u(j-1)}^T + \sum_{l=j}^{i}(\bar{f}_{u(\bar{l})}^T\prod_{\bar{l}=j-1}^{l-1}\bar{f}_{x(\bar{l}-1)}^T)\bar{F}_{x(l)}^T$$

$$\mathbb{H}_{(j)3} = \bar{f}_{u(j)}^T\bar{f}_{x(j+1)}^T\cdots\bar{f}_{x(i)}^T$$

$$\mathbb{H}_{(j)4} = \bar{F}_{u(j)}^T + \sum_{l=j+1}^{i}(\bar{f}_{u(\bar{l})}^T\prod_{\bar{l}=j}^{l-1}\bar{f}_{x(\bar{l}-1)}^T)\bar{F}_{x(l)}^T.$$

From the structure of $H_i, H_{i-1}M_i, \ldots, H_h\bar{M}_{h+1:i}$, it is known that $\mathbb{H}_{(i)1} = \mathbb{H}_{(i-1)3}, \mathbb{H}_{(i)2} = \mathbb{H}_{(i-1)4}$ and for any $j > h$, we always have $\mathbb{H}_{(j)1} = \mathbb{H}_{(j-1)3}, \mathbb{H}_{(j)2} = \mathbb{H}_{(j-1)4}$.

Equation (62) can be satisfied if and only if there exist a matrix $\bar{\xi}$ that can satisfy the equation below:

$$\begin{bmatrix}\bar{\xi} & I_i\end{bmatrix}\bar{\mathbb{H}}_{h:i} = 0_{N_z(i)\times N_p(i)} \tag{65}$$

where $\bar{\mathbb{H}}_{h:i} = \begin{bmatrix} \mathbb{H}_{(h)3} & \mathbb{H}_{(h)4} \\ \vdots & \vdots \\ \mathbb{H}_{(j)3} & \mathbb{H}_{(j)4} \\ \vdots & \vdots \\ \mathbb{H}_{(i-1)3} & \mathbb{H}_{(i-1)4} \\ \mathbb{H}_{(i)3} & \mathbb{H}_{(i)4} \end{bmatrix}$ and $[\bar{\xi} \ I_i] = [\bar{\xi}_h \ \cdots \ \bar{\xi}_j \ \cdots \ \bar{\xi}_{i-1} \ I_i]$. It is also known that dimension of null space of column vectors in $\bar{\mathbb{H}}_{h:i}$ should be at least $N_s(i)$.

Here, right hand side of $\bar{\mathbb{H}}_{h:i}$ can be rewritten as one form as

$$\begin{bmatrix} \mathbb{H}_{(h)4} \\ \vdots \\ \mathbb{H}_{(j)4} \\ \vdots \\ \mathbb{H}_{(i-1)4} \\ \mathbb{H}_{(i)4} \end{bmatrix} = \bar{\mathbb{H}}_{u_{h:i}}\bar{\mathbb{H}}_{x_{h:i}}$$

where

$$\bar{\mathbb{H}}_{u_{h:i}}=\begin{bmatrix} \bar{F}^T_{u(h)} & \bar{f}^T_{u(\bar{h})} & \cdots & \cdots & \cdots & \bar{f}^T_{u(h)}\prod_{\bar{l}=h}^{i-1}\bar{f}^T_{x(\bar{l}-1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \bar{F}^T_{u(j)} & 0 & \cdots & \bar{f}^T_{u(\bar{j})} & \cdots & \bar{f}^T_{u(\bar{j})}\prod_{\bar{l}=j}^{i-1}\bar{f}^T_{x(\bar{l}-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{F}^T_{u(i-1)} & 0 & \cdots & 0 & \cdots & \bar{f}^T_{u(\bar{i}-1)} \\ \bar{F}^T_{u(i)} & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}$$

and $\bar{\mathbb{H}}_{x_{h:i}}=\begin{bmatrix} I \\ \bar{F}^T_{x(h)} \\ \vdots \\ \bar{F}^T_{x(j)} \\ \vdots \\ \bar{F}^T_{x(i-1)} \end{bmatrix}$. From (65), it is known that (65) can be satisfied only if $\begin{bmatrix}\bar{\xi} & I\end{bmatrix}\bar{\mathbb{H}}_{u_{h:i}}\bar{\mathbb{H}}_{x_{h:i}}=0_{N_z(i)\times n}$.

Here, since $\bar{\mathbb{H}}_{u_{h:i}}\bar{\mathbb{H}}_{x_{h:i}}=\begin{bmatrix}\vdots \\ \vdots \\ \bar{F}^T_{u(i)}\end{bmatrix}$ is not a zero matrix, $\begin{bmatrix}\bar{\xi} & I\end{bmatrix}\bar{\mathbb{H}}_{u_{h:i}}\bar{\mathbb{H}}_{x_{h:i}}=0_{N_z(i)\times n}$ only happens when $[\bar{\xi}\ \ I]\bar{\mathbb{H}}_{u_{h:i}}=0$.

Based on the derivation above, it is known that

a) When $\bar{f}_{u(j)}\ \ \forall h\le j\le i-1$ are all full rank that

$$\dim(\bar{f}^T_{u(j)})=m\ \ \forall h\le j\le i-1$$

where dim(:) represents the dimension of the column vector space of the matrix, and $\bar{F}_{u(i)}$ is full rank. From the structure of matrix $\bar{\mathbb{H}}_{u_{h:i}}$, $\bar{\mathbb{H}}_{u_{h:i}}$ is also full rank, indicating that there exist no $\bar{\xi}$ make $[\bar{\xi}\ \ I]\bar{\mathbb{H}}_{u_{h:i}}=0_{N_z(i)\times(n-h+1+m)}$. Finally, it is indicating that ξ_s let (62) is not satisfied and ξ let (58) is not satisfied. Therefore,

$$\Gamma_{\Omega_{h:i-1}}\not\subseteq\Gamma_{\Phi_{h(i)}}$$

in this case.

b) When $\dim(\bar{f}^T_{u(j)})<m\ \ \forall h\le j\le i-1$ and $\bar{F}_{u(i)}$ is full rank, from the structure of matrix $\bar{\mathbb{H}}_{u_{h:i}}$, dimension of null space of column vector space of $\bar{\mathbb{H}}_{u_{h:i}}$ satisfies

$$\dim(\text{null}(\bar{\mathbb{H}}_{u_{h:i}}))=\sum_{j=h}^{i-1}\dim(\text{null}(\bar{f}^T_{u(j)})). \tag{66}$$

Since the dimension of the null space of the column vectors in $\bar{\mathbb{H}}_{h:i}$ should be at least $N_s(i)$, when $\dim(\bar{\mathbb{H}}_{u_{h:i}})$ calculated in (66) satisfies $\dim(\bar{\mathbb{H}}_{u_{h:i}})<N_s(i)$, $\bar{\xi}$ make (65) is not satisfied. Furthermore, ξ_s let (62) is not satisfied and ξ let (58) is not satisfied. Therefore,

$$\Gamma_{\Omega_{h:i-1}}\not\subseteq\Gamma_{\Phi_{h(i)}}$$

in this case. ■

REFERENCES

[1] Y. Li, K. P. Tee, R. Yan, W. L. Chan, and Y. Wu, "A framework of human-robot coordination based on game theory and policy iteration," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1408–1418, Dec. 2016.

[2] N. Aghasadeghi and T. Bretl, "Inverse optimal control for differentially flat systems with application to locomotion modeling," in *Proc. IEEE Int. Conf. Robotics and Automation*, Hong Kong, China, 2014, pp. 6018–6025.

[3] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—An inverse optimal control approach," *Auton. Robots*, vol. 28, no. 3, pp. 369–383, Apr. 2010.

[4] J. Saez-Gallego, J. M. Morales, M. Zugno, and H. Madsen, "A data-driven bidding model for a cluster of price-responsive consumers of electricity," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 5001–5011, Nov. 2016.

[5] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Human behavior modeling with maximum entropy inverse optimal control," in *Proc. AAAI Spring Symp.-Human Behavior Modeling*, Stanford, USA, 2009.

[6] B. Berret, E. Chiovetto, F. Nori, and T. Pozzo, "Evidence for composite cost functions in arm movement planning: An inverse optimal control approach," *PLoS Comput. Biol.*, vol. 7, no. 10, p. e1002183, Oct. 2011.

[7] H. El-Hussieny, A. A. Abouelsoud, S. F. M. Assal, and S. M. Megahed, "Adaptive learning of human motor behaviors: An evolving inverse optimal control approach," *Eng. Appl. Artif. Intell.*, vol. 50, pp. 115–124, Apr. 2016.

[8] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. 23rd Nat. Conf. Artificial Intelligence*, Chicago, USA, 2008, pp. 1433–1438.

[9] S. Dempe, *Foundations of Bilevel Programming*. New York, USA: Springer, 2002.

[10] K. Hatz, J. P. Schlöder, and H. G. Bock, "Estimating parameters in optimal control problems," *SIAM J. Sci. Comput.*, vol. 34, no. 3, pp. A1707–A1728, Jan. 2012.

[11] S. Albrecht and M. Ulbrich, "Mathematical programs with complementarity constraints in the context of inverse optimal control for locomotion," *Optim. Methods Software*, vol. 32, no. 4, pp. 670–698, Aug. 2017.

[12] S. Dempe, F. Harder, P. Mehlitz, and G. Wachsmuth, "Solving inverse optimal control problems via value functions to global optimality," *J. Glob. Optim.*, vol. 74, no. 2, pp. 297–325, Jun. 2019.

[13] A. E. Bryson and Y. C. Ho, *Applied Optimal Control: Optimization, Estimation and Control*. Routledge, 2018.

[14] M. Johnson, N. Aghasadeghi, and T. Bretl, "Inverse optimal control for deterministic continuous-time nonlinear systems," in *Proc. 52nd IEEE Conf. Decision and Control*, Firenze, Italy, 2013, pp. 2906–2913.

[15] T. L. Molloy, J. J. Ford, and T. Perez, "Finite-horizon inverse optimal control for discrete-time nonlinear systems," *Automatica*, vol. 87, pp. 442–446, Jan. 2018.

[16] T. L. Molloy, J. J. Ford, and T. Perez, "Online inverse optimal control for control-constrained discrete-time systems on finite and infinite horizons," *Automatica*, vol. 120, p. 109109, Oct. 2020.

[17] M. Almobaied, I. Eksin, and M. Guzelkaya, "A new inverse optimal control method for discrete-time systems," in *Proc. 12th Int. Conf. Informatics in Control, Automation and Robotics*, Colmar, France, 2015, pp. 275–280.

[18] M. Almobaied, I. Eksin, and M. Guzelkaya, "Inverse optimal controller based on extended Kalman filter for discrete-time nonlinear systems," *Optim. Control Appl. Methods*, vol. 39, no. 1, pp. 19–34, Jan.–Feb. 2018.

[19] P. Prasanna, J. Jacob, and M. P. Nandakumar, "Inverse optimal control of a class of affine nonlinear systems," *Trans. Inst. Meas. Control*, vol. 41, no. 9, pp. 2637–2650, Feb. 2019.

[20] B. Huang, X. Ma, and U. Vaidya, "Data-driven nonlinear stabilization using Koopman operator," in *The Koopman Operator in Systems and Control: Concepts, Methodologies, and Applications*, A. Mauroy, I. Mezić, and Y. Susuki, Eds. Cham, Switzerland: Springer, 2020, pp. 313–334.

[21] S. M. Khansari-Zadeh and A. Billard, "Learning control Lyapunov

function to ensure stability of dynamical system-based robot reaching motions," *Rob. Auton. Syst.*, vol. 62, no. 6, pp. 752–765, Jun. 2014.

[22] H. Ravanbakhsh and S. Sankaranarayanan, "Learning control Lyapunov functions from counterexamples and demonstrations," *Auton. Robots*, vol. 43, no. 2, pp. 275–307, Feb. 2019.

[23] Y. C. Chang, N. Roohi, and S. Gao, "Neural Lyapunov control," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[24] H. Zhang, J. Umenberger, and X. Hu, "Inverse optimal control for discrete-time finite-horizon linear quadratic regulators," *Automatica*, vol. 110, p. 108593, Dec. 2019.

[25] C. Yu, Y. Li, H. Fang, and J. Chen, "System identification approach for inverse optimal control of finite-horizon linear quadratic regulators," *Automatica*, vol. 129, p. 109636, Jul. 2021.

[26] A. Keshavarz, Y. Wang, and S. Boyd, "Imputing a convex objective function," in *Proc. IEEE Int. Symp. Intelligent Control*, Denver, USA, 2011, pp. 613–619.

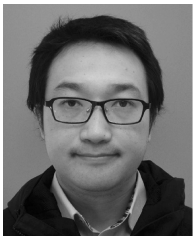
[27] M. Schultheis, D. Straub, and C. A. Rothkopf, "Inverse optimal control adapted to the noise characteristics of the human sensorimotor system," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9429–9442, 2021.

[28] J. Blot, and N. Hayek, *Infinite-Horizon Optimal Control in the Discrete-Time Framework*. New York, USA: Springer, 2014.

[29] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. 1*. 2nd ed. Belmont, USA: Athena Scientific, 2000.

[30] J. Blot and H. Chebbi, "Discrete time Pontryagin principles with infinite horizon," *J. Math. Anal. Appl.*, vol. 246, no. 1, pp. 265–279, Jun. 2000.

[31] W. Jin, D. Kulić, J. F.-S. Lin, S. Mou, and S. Hirche, "Inverse optimal control for multiphase cost functions," *IEEE Trans. Robot.*, vol. 35, no. 6, pp. 1387–1398, Dec. 2019.



Sheng Cao received the Ph.D. degree in 2017 from the Graduate School of System Informatics at the Kobe University. His research interests include robotic rehabilitation, human-robot interaction, biomechanics analysis, robot's safe control, etc. He is an Assistant Professor of Kobe University.

Zhiwei Luo received the B.Eng. degree from Huazhong University of Science and Technology in 1984, and the M.Eng., Dr.Eng. from Nagoya University in 1991 and 1992, respectively. He was an Assistant Professor of Toyohashi Institute of Technology from 1992 to 1994, a Frontier Researcher of RIKEN from 1994 to 1999, an Associate Professor of Yamagata University from 1999 to 2001, a Team Leader of RIKEN from 2001 to 2008, and a Professor of Kobe University since 2006. His research interests include robotics, system control, artificial intelligence and health engineering. He is a Member of SCI, SICE, RSJ. He is honored Fellow of SICE.

Changqin Quan received the Ph.D. degree in 2011 from Faculty of Engineering, University of Tokushima, Japan. She is currently an Associate Professor in Kobe University. Her research interests include natural language processing, affective computing, and artificial intelligence.