Sequential Inverse Optimal Control of Discrete System

Cao, Sheng

Luo, Zhi-Wei

Quan, Changqin

(Citation)
2023 9th International Conference on Control, Decision and Information Technologies (CoDIT):1924-1929

(Issue Date)
2023-10-24

(Resource Type)
conference paper

(Version)
Accepted Manuscript

Sequential Inverse Optimal Control of Discrete System

1st Sheng Cao
Graduate School of System Informatics
Kobe University
Kobe, Japan

2nd Zhiwei luo
Graduate School of System Informatics
Kobe University
Kobe, Japan

3rd Changqin Quan
Graduate School of System Informatics
Kobe University
Kobe, Japan

***Abstract*—This paper suggests the sequential inverse optimal control (SIOC) approach for discrete-time system. It determines the unknown weight vectors of the cost function in real-time utilizing the input and output of a discrete-time system that is optimally controlled. It systematically calculates the possible solution spaces and their intersections until the intersection space's dimension is reduced to one. The remaining one-dimensional vector at the intersection of the possible solution space is the solution to the IOC problem. In this method, we clarify the conditions on the decrease of the dimension of the intersection space follow with the tackling method of noisy data. The simulation results illustrate the high calculation speed and effective noise-tackling results of our method.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

In contrast to the traditional optimal control problem of solving for optimal control inputs, the problem of IOC is concerned with solving for the cost function based on the data from the observed system state and control inputs.

Optimality of system behavior has been investigated via inverse optimal control (IOC) approach in various applications, including robotics ([1]), biological systems([2] and [3]) and marketing systems ([4]), etc. [1] proposed a method to estimate the cost function of human in order to model the human behavior and therefore, achieve effective human-robot interaction control. In ([5]), IOC has been utilized for analysis of the cost combination of human's motion. Additionally, the biological behavior has been modeled as the problem of Inverse Linear Quadratic Regulator (ILQR) and [6] proposed an adaptive method for modelling and analyzing human's reach-to-grasp behavior. In the traffic research area, IOC method has been utilized in ([7]) to analyze the route choices of taxi drivers.

IOC problem has been considered for simple dynamical models in ([8]) by treating the IOC problem as a special bi-level optimization problem ([9]), where the lower level is the conventional forward optimal control problem and the upper level is the inversion problem. Inspired by this idea, [3] suggested a bi-level optimization method based on IOC method to analyze the locomotion motions. In ([10]), also for the purpose of analyzing locomotion, mathematical programs with complementary constraints in context of inverse optimal control method has been discussed . In order to simplify the problem, [11] transferred the bi-level optimization problem into a single level one and a globally optimal solution has been computed.

In contrast to the hierarchy structure of the bi-level optimization method, a method of inverse optimal control based on minimizing how much observed trajectories violate the first-order necessary conditions was proposed by [12]. Based on the discrete-time minimum principle, [13] proposed a method for recovering the unknown objective-function parameters of discrete-time with finite horizon. Furthermore, in ([14]), one method has been proposed to solve the online calculation of discrete-time IOC in both finite and infinite horizons, however, it requires the necessary invertibility condition of a Jacobian. In addition, the convergence of the cost weights has not been theoretically studied.

This paper suggests a novel sequential inverse optimal control technique to address these issues. The method clarifies the conditions for the convergence of the weight estimation in every step, which may be very helpful in applications. On the other hand, this article also takes into account the issues with noisy data. We propose a strategy for selecting the possible solution space in noisy cases.

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, $x_k = [x_k^1, ..., x_k^n]^T \in R^n$ represents system states and $u_k = [u_k^1, ..., u_k^m]^T \in \mathcal{U}$ denotes control input of the system belonging to a closed, bounded and convex constrained set $\mathcal{U} \subseteq R^m$. We use $x_{[0,K]}$ to denote the state sequence $\{x_k : 0 \leq k \leq K\}$ and $u_{[0,K]}$ to denote control input sequence $\{u_k : 0 \leq k \leq K\}$.

In a typical optimal control issue, we construct the optimum control input $u^*_{[0,K]}$ and obtain a series of optimal states $x^*_{[0,K]}$ to minimize the following cost function, subject to the dynamics (1). (Upper-script $*$ stands for the meaning of optimal.)

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

Identify applicable funding agency here. If none, delete this.

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

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

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

For the purposes of our problem, we assume that both $x_{[0,K]}$ and $u_{[0,K]}$ represent a solution to minimize (2) for the system dynamics (1). The goal of this study is to implement online estimation of the cost weight vector q in the cost function (2), which entails computing the vector q using system state x_k and control input u_k provided step-by-step without the storage and batch processing, as well as prior knowledge of the time horizon K.

III. Sequential inverse optimal control

The maximum principle in finite and infinite horizon optimal control problems is introduced in this section. This principle is valid when any of the conditions in Assumption 1 is encountered.

Based on introduced maximum principle, we suggest our sequential inverse optimal control strategy.

A. Maximum Principle in finite and infinite horizon optimal control

Considering (1) and (2), the Hamiltonian function associated with the optimal control issue is defined as

$$\begin{aligned} H_k(x_{k-1}, u_{k-1}, \lambda_k, q) &= q^T F(x_{k-1}, u_{k-1}) \\ &+ \lambda_k^T f(x_{k-1}, u_{k-1}), \end{aligned} \tag{4}$$

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

The partial derivatives of the Hamiltonian with regard to x_{k-1} and u_{k-1} are represented by $\nabla_x H_k(x_{k-1}, u_{k-1}, \lambda_k, q) \in R^n$ and $\nabla_u H_k(x_{k-1}, u_{k-1}, \lambda_k, q) \in R^m$, respectively.

Assumption 1: For all $k \geq 0$, it is assumed in the following studies that the partial derivative of the dynamics satisfies the following assumptions: $\dfrac{\partial f}{\partial x_k}$ is invertible.

In addition, as the method proposed by [14], the inactive constraint times of the control input is defined as:

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

$$\kappa_k \triangleq \{0 \leq k \leq l : u_k \in \mathrm{int}(\mathcal{U})\} \tag{5}$$

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

As previously indicated in ([14]), ([15]), ([16]) and ([17]), the following lemma holds for the aforementioned assumptions and definition.

Lemma 1: Suppose that the optimal control problem defined by (1) and (2) is solved by trajectories $x_{[0,K]}$ and $u_{[0,K]}$ and if assumption of (A) or (B) holds, then there exist costate vectors $\lambda_0, \ldots, \lambda_K$ that satisfy the combined Pontryagin's maximum principle (PMP) as

$$\bar{F}_{x(k-1)}^T q + \bar{f}_{x(k-1)}^T \lambda_k = \lambda_{k-1} \tag{6}$$

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

$$\bar{F}_{u(k-1)}^T q + \bar{f}_{u(k-1)}^T \lambda_k = 0 \tag{7}$$

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

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

Proof:
In general, (6) and (7) can be determined by computing the gradients of (2) for (1) using (4) with respect to the vectors of x_k and u_k, respectively. The brief proof of this lemma for the assumption is given in [14](Lemma 1) by using the results of [16] (Proposition 3.3.2) and [17] (Theorem 2). □

B. Construction of the Sequential Inverse Optimal Control Method

If the optimal control problem of (1) and (2) is solved by ($x_{[T_0,T_f]}, u_{[T_0,T_f]}$) which is the solution to (6) and (7), then by considering (7) in step k and $k-1$ and substituting (6) into it in step k, we have

$$H_k s_k = 0, \tag{8}$$

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 $s_k = \begin{bmatrix} \lambda_{k+1} \\ q \end{bmatrix}$.

Also, by considering (6), a backward recursive relation from s_k to s_{k-1} can be formulated as

$$M_k s_k = s_{k-1}, \tag{9}$$

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

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 with all elements be zero. Noted that, (8) holds only when $k, k-1 \in \kappa_k$ and (9) holds all the time even if $k \notin \kappa_k$.

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

$$\bar{M}_{h:k} s_k = s_h, \tag{10}$$

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

Since the vector s_h comprises both the cost weight vector q and co-state λ_{h+1}, the objective of building online inverse optimal control is then changed to find vector s_h based on finite forward steps of k.

From (8), it is known that s_k locates in the null space of H_k and s_k can be calculated as

$$s_k = (I_N - H_k^+ H_k) r_k, \tag{11}$$

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

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

$$\Phi_{h(i)} r_i = \bar{M}_{h:i} \Theta_i r_i = s_h, \tag{12}$$

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

Here, the column vector space of $\Phi_{h(i)}$ can be denoted as $\Gamma_{\Phi_{h(i)}} = \mathrm{span}(\Phi_{h(i)})$ for $h \leq i \leq k$, where span(:) is denoting the span of a matrix. From (12), we have $s_i \in \Gamma_{\Phi_{h(i)}}$ for $h \leq i \leq k$. Therefore, the spaces $\Gamma_{\Phi_{h(h)}}, \ldots, \Gamma_{\Phi_{h(k)}}$ are the possible solution spaces and the intersection of these possible solution spaces is defined as $\Gamma_{\Omega_{h:k}} = \Gamma_{\Phi_{h(h)}} \cap \cdots \cap \Gamma_{\Phi_{h(k)}}$.

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

In order to obtain s_h, it is necessary to discuss the decrease of the dimension of $\Gamma_{\Omega_{h:k}}$ with the increase of k.

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

$$\dim(\Gamma_{\Omega_{h:i-1}}) > \dim(\Gamma_{\Omega_{h:i}}), \tag{13}$$

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 $\Gamma_{\Phi_{h(j)}}$

$$\dim(\Gamma_{\Phi_{h(i)}} \cap \Gamma_{\Phi_{h(j)}}) \leq \dim(\Gamma_{\Phi_{h(i)}}), \tag{14}$$

where the equality 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

$$\dim(\Gamma_{\Omega_{h:i}}) \leq \dim(\Gamma_{\Omega_{h:i-1}}), \quad h < i < K. \tag{15}$$

The equality 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 be decreased with the increase of k. □

As can be seen from Proposition 1, as the increase of the step, at some step instant k_f, the rank of common intersection subspace of $\Gamma_{\Phi_{h(h)}}, \ldots, \Gamma_{\Phi_{h(k_f)}}$ becomes one and that this unique intersection is s_h.

Here, we give our main result to clarify the condition for the circumstances surrounding the shrinkage of intersection space $\Gamma_{\Omega_{h:i}}$.

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

(a)

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

(b)

- $\mathrm{Dim}(\bar{f}^T_{u(j)}) < m \quad \forall h \leq j \leq i-1$.
- $\bar{F}^T_{u(i)}$ is full rank.
- $$\sum_{j=h}^{i-1} \mathrm{Dim}(\mathrm{null}(\bar{f}^T_{u(j)})) < N_z(i)$$

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

Proof:

Due to space constraints, a comprehensive proof of Theorem 1 will be presented in a forthcoming publication. □

As a result, if either condition (a) or (b) of Theorem 1 is met, the dimension of the intersection of the possible solution space must be reduced at each step.

C. Calculation of the vector s_h

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

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

$$\Omega_s = \Omega_{h:i} = \mathrm{null}(Y_{h(i)}), \tag{16}$$

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

When $\dim(\Gamma_{\Omega_{h:i}}) = 1$, Ω_s becomes a 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 keep the same direction with V_n or $-V_n$. Since using $-s_h$ instead of s_h results in negative cost function, s_h can be selected as

$$s_h = \begin{cases} V_n a_p, & C(x_k, u_k, \hat{q}) \geq 0, \\ -V_n a_p, & C(x_k, u_k, \hat{q}) < 0, \end{cases} \tag{17}$$

where $\hat{q} = [V_n a_p]_{n+1:n+n_f}$ is the vector constructed by $(n+1)$-th element to $(n+n_f)$-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 $(n+1)$-th element to $(n+n_f)$-th element of V_n and $||q||$ is previous known value. Here, we scale V_n using the known norm of q in order to obtain an unique and accurate cost weight vector.

2) Calculation for Control Constraints: When control constraints are met, we construct the calculation process as

$$\begin{cases} \Gamma_{\Omega_{h:i}} = \Gamma_{\Omega_{h:i-1}} \cap \Gamma_{\Phi_{h(i)}}, & u_i \in \mathrm{int}(\mathcal{U}) \;\&\; u_{i-1} \in \mathrm{int}(\mathcal{U}), \\ \textit{Reinitialize} \quad \Gamma_{\Omega_{h:i}}, & u_i \in \mathrm{int}(\mathcal{U}) \;\&\; u_{i-1} \notin \mathrm{int}(\mathcal{U}), \\ \textit{Skip the Step}, & \textit{Otherwise}. \end{cases} \tag{18}$$

We stop the calculation when the control constraint is active and reinitialize, resume the IOC calculation when it is inactive. The high calculation speed of our method makes it possible to quickly complete the IOC calculation after control constraints active duration so that it is not necessary to store the data of the duration before control constraint's active times ([14]).

IV. Simulation Examples

In this section, we use simulation in two different contexts to demonstrate our methodology. The system dynamics in the simulations are

$$x_{k+1} = Ax_k + Bu_k \tag{19}$$

where $x_k \in \mathbb{R}^3$,$u_k \in \mathbb{R}^2$,$A \in \mathbb{R}^{3\times 3}$ and $B \in \mathbb{R}^{3\times 2}$. The cost function selected in the simulations is following

$$V_1(x,t) = \sum_{k=0}^{k=\infty} \frac{1}{2}(x_k^T Q_x x_k + u_k^T Q_u u_k) \tag{20}$$

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

To repeatedly check the calculation steps in the simulation of the suggested approach, in both of our simulations, we reset the matrix $N_{I_{h:i}}$ with $N_{I_{h:i}} = N_{\Phi_{h(i)}}$ after one result calculated out in step i, letting the intersection space only contain data from step i so as to start a new cycle of calculations for the inverse optimal control.

A. Simulation 1

In setting 1, A and B have been chosen as

$$A = \begin{bmatrix} 0.9654 & 5.4572 & 0 \\ -0.0013 & 0.9545 & 0 \\ -0.0038 & 5.5437 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0.0284 & 0.0142 \\ 0.020 & 0.010 \\ 0.056 & 0.028 \end{bmatrix} \tag{21}$$

The discrete LQR method on infinite horizon generates the optimal control input as well as the system states with 50 steps. The $dlqr()$ function in Matlab 2016b produced the K_s used in calculating optimal control input $u_k = -K_s x_k$.

We contrasted the results of the suggested method with the online IOC method suggested by [14] in two aspects: the number of steps needed to calculate the cost weight vector in each cycle (Fig. 1,Fig. 2)) and the recovery error of cost weights (Fig. 3,Fig. 4).

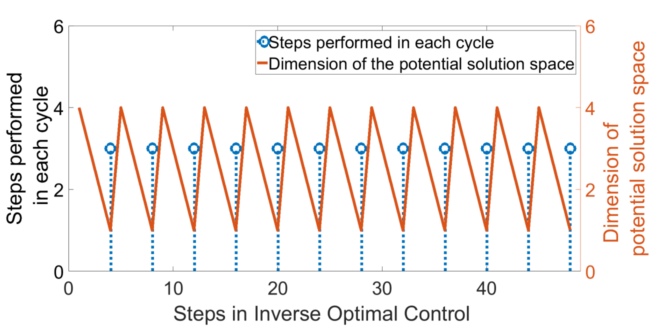

Fig. 1. Steps costed in Setting 1

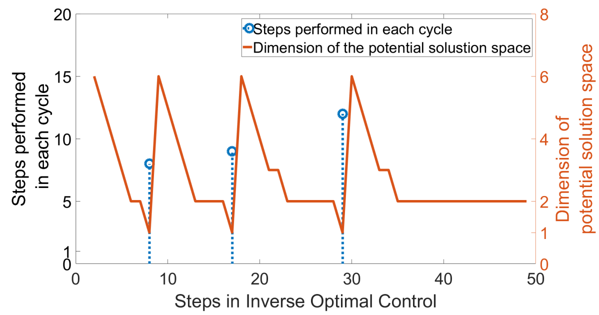

Fig. 2. Steps costed in Setting 1 (Method of [14])

The steps required in our method are shown in Fig. 1 while the steps required in previous online IOC method proposed by [14] is shown in 2. The total number of steps taken during the simulations are shown by the x axis in both figures. The height of the blue dotted line, which is associated to the left y axis, represents the number of steps taken during each IOC cycle. The red line in these two figures, whose height is relating to the right y axis, illustrate the dimension variation of intersection of possible solution's space ($\Sigma_{I_{h:i}}$ in this research and null space of $\bar{Q}_k$ in previous [14]).

The result in Fig. 1 shows that the dimension of $\Sigma_{I_{h:i}}$ is decreasing in every steps in each calculation cycle. The online IOC completes the computation at an incredibly high speed, with the maximum number of steps in one cycle being 4. This result verifies Theorem 1 by considering that Eq. (21) is satisfying condition (2) in Theorem 1.

By compared with the result of the proposed method, with previous method, the dimension of the null space of matrix $\bar{Q}_k$ in Algorithm 1 of [14], is not decreasing all the time. As the result, the steps of one cycle for online IOC's calculation in Fig. 2 is always larger than the suggested method of this paper.

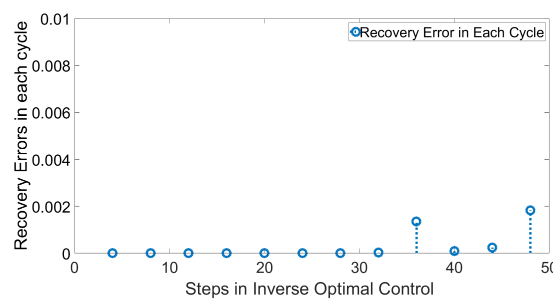

Fig. 3. Estimation Error of Setting 1

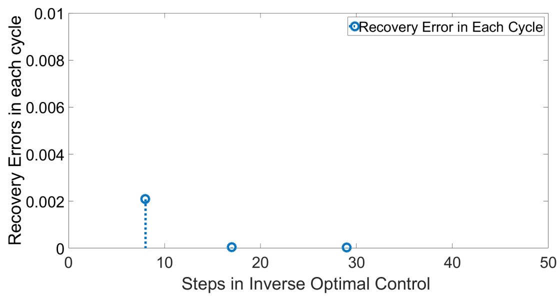

Fig. 4. Estimation Error of Setting 1 (Method of [14])

Fig. 3 as well as Fig. 4 show the recovery error of both methods which is calculated with

$$e = ||\hat{q} - q|| \tag{22}$$

where $\hat{q}$ denotes the estimation vector of q calculated by the proposed online IOC method and previous method. From these two figures, we know that the recovery error of both method are all very small in every calculation cycle.

Therefore, all the above results show that our proposed method in this paper can effectively improve the calculation speed while keep the recovery accuracy of IOC.

B. Simulation 2

In simulation 2, A and B of linear system dynamic have been selected as

$$A = \begin{bmatrix} 0.9 & 1.8 & 0 \\ 0.13 & 0.26 & 0 \\ 0.38 & 0.76 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0.0284 & 0.0142 \\ 0.0020 & 0.0010 \\ 0.0056 & -0.0028 \end{bmatrix} \tag{23}$$

where the Jacobian's positivity assumption is guaranteed with the setting of A and the Jacobian's invertibility assumption is not satisfied due to the rank deficient setting of A. It means that the previous method is not applicable to be used to recover the cost weights of such kind of system's optimal behavior.

Moreover, the problem of control constraint is also taken into consideration in the simulation. The control constraints set is selected as $\mathcal{U} \triangleq \{u_{ki} > -0.2\}$ where u_{ki} denotes the ith element of u_k

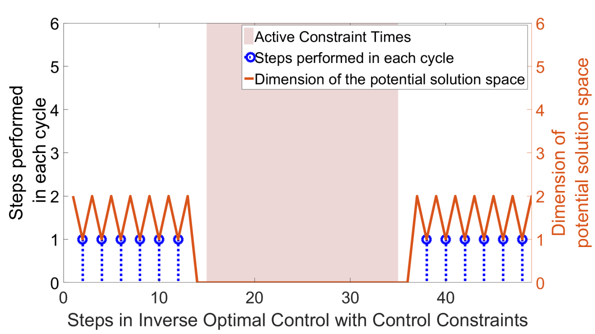

Fig. 5. Steps costed in Setting 2

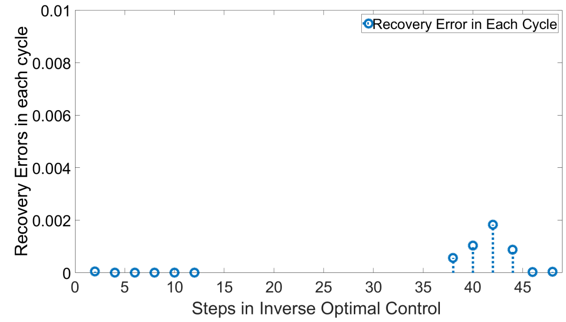

Fig. 6. Estimation Error of Setting 2

Fig. 5 shows the calculation steps in every cycle in simulation 2, which is all 2 in this results and the shaded region (15s to 35s) in Fig. 5 represents the control constraint active area. Here, as same as the result in simulation 1, the dimension's variation of possible solution's space verifies the Theorem 1 and the variation of the dimension of possible solution's space after the control constraints activated shows that the proposed algorithm is effective on handling the problem from control constraints.

Fig. 6 shows the estimation error in this simulation. Even under the condition that A is rank deficient, the extremely small errors in all cycles show that our method can effectively recover the required cost weights with considerable accuracy. Moreover, it is also shown in Fig. 6 that the estimation error also is small after the control constraints activated. Combining both Fig. 6 and Fig. 5, the results of simulation 2 show us that the proposed method can effectively tackle the rank deficient A and this setting should not have any impact on the calculation speed as well as recovery accuracy of our proposed method. The control constraint problem can also be successfully solved using our method.

V. Conclusion

In this paper, we propose an online calculation method of discrete-time IOC which is applicable to recover both finite and infinite horizon optimal control's cost weights.

Firstly, we establish a calculation method for the possible solution space of the IOC and sequentially calculating the intersection of all possible solution spaces in each previous steps. With the necessary steps' accumulated, the dimension of the intersection space shall be decrease to 1 and the remaining vector in the intersection space will be the required solution to IOC.

After that, we then exploit the property of solution's convergence. It is found that if any of the two conditions is satisfied, the intersection space's dimension will be decreased in every step, which satisfies the real time calculation and is meaningful to the IOC for discrete-time optimal control on finite-horizon.

The simulation results illustrate that our sequential IOC algorithm is effective and has high speed of calculation.

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 Transactions on Robotics*, vol. 32, no. 6, pp. 1408–1418, 2016.

[2] N. Aghasadeghi and T. Bretl, "Inverse optimal control for differentially flat systems with application to locomotion modeling," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6018–6025.

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

[4] A. Keshavarz, Y. Wang, and S. Boyd, "Imputing a convex objective function," in *2011 IEEE international symposium on intelligent control*. IEEE, 2011, pp. 613–619.

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

[6] H. El-Hussieny, A. Abouelsoud, S. F. Assal, and S. M. Megahed, "Adaptive learning of human motor behaviors: An evolving inverse optimal control approach," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 115–124, 2016.

[7] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Human behavior modeling with maximum entropy inverse optimal control." in *AAAI spring symposium: human behavior modeling*, vol. 92, 2009.

[8] K. Hatz, J. P. Schl¥"oder, and H. G. Bock, "Estimating parameters in optimal control problems," *SIAM Journal on Scientific Computing*, vol. 34, no. 3, pp. A1707–A1728, 2012.

[9] S. Dempe, *Foundations of bilevel programming*. Springer Science & Business Media, 2002.

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

[11] S. Dempe, F. Harder, P. Mehlitz, and G. Wachsmuth, "Solving inverse optimal control problems via value functions to global optimality," *Journal of Global Optimization*, vol. 74, no. 2, pp. 297–325, 2019.

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

[13] 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, 2018.

[14] ——, "Online inverse optimal control for control-constrained discrete-time systems on finite and infinite horizons," *Automatica*, vol. 120, p. 109109, 2020.

[15] J. Blot and N. Hayek, *Infinite-horizon optimal control in the discrete-time framework*. Springer, 2014.

[16] D. Bertsekas, *Dynamic programming and optimal control: Volume I*. Athena scientific, 2012, vol. 1.

[17] J. Blot and H. Chebbi, "Discrete time pontryagin principles with infinite horizon," *Journal of Mathematical Analysis and Applications*, vol. 246, no. 1, pp. 265–279, 2000.

APPENDIX

Proof:

If $\Gamma_{\Omega_{h:i-1}} \subseteq \Gamma_{\Phi_{h(i)}}$, we have $\text{null}(\Gamma_{\Omega_{h:i-1}}) \supseteq \text{null}(\Gamma_{\Phi_{h(i)}})$, representing that there exist a full rank matrix $\xi = \begin{bmatrix}\xi_h & \dots & \xi_j & \dots & \xi_{i-1}\end{bmatrix} \in R^{N_{z(i)} \times \sum_{j=h}^{i} N_{z(j)}}$ satisfy the equation below, where $N_{z(j)} \quad \forall h \le j \le i$ denotes the dimension of null space of $\Gamma_{\Phi_{h(j)}}$.

$$\xi\bar{\Omega}_{h:i-1} = \text{null}(\Phi_{h(i)})^T \tag{24}$$

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

$$\xi\bar{\Omega}_{h:i-1}\Phi_{h(i)} = 0_{N_z(i)\times N_p(i)} \tag{25}$$

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 orthogonal complement to the column vector space of $\Phi_{h(i)}$.

Here, under the Assumption 1 (A) 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 $\Phi_{h(j)}$. (25) can be satisfied if and only if there exists a nonzero matrix ξ_o satisfies

$$\xi_o\bar{\Omega}'_{h:i-1}\Theta_i = 0_{N_z(i)\times N_p(i)}, \tag{26}$$

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 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

$$rank(\xi_o\bar{\Omega}'_{h:i-1}) + rank(\Theta_i) = N. \tag{27}$$

From (27) and (26), it is known that row vector space of $\xi_o\bar{\Omega}'_{h:i-1}$ is orthogonal complement to the vector space of Θ_i, meaning that there exist a matrix ξ_s satisfies

$$\begin{bmatrix}\xi_s & I_i\end{bmatrix}\begin{bmatrix}H_h\bar{M}_{h+1:i} \\ \vdots \\ H_j\bar{M}_{j+1:i} \\ \vdots \\ H_{i-1}M_i \\ H_i\end{bmatrix} = 0_{N_z(i)\times N} \tag{28}$$

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

(28) 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

$$\begin{aligned} H_i &= \begin{bmatrix} \mathbb{H}_{(i)1} & \mathbb{H}_{(i)2} \\ \mathbb{H}_{(i)3} & \mathbb{H}_{(i)4} \end{bmatrix} \\ &= \begin{bmatrix} \bar{f}^T_{u(i-1)}\bar{f}^T_{x(i)} & \bar{F}^T_{u(i-1)} + \bar{f}^T_{u(i-1)}\bar{F}^T_{x(i)} \\ \bar{f}^T_{u(i)} & \bar{F}^T_{u(i)} \end{bmatrix} \end{aligned} \tag{29}$$

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

$$H_j\bar{M}_{j+1:i} = \begin{bmatrix} \mathbb{H}_{(j)1} & \mathbb{H}_{(j)2} \\ \mathbb{H}_{(j)3} & \mathbb{H}_{(j)4} \end{bmatrix} \tag{30}$$

where

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

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

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

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

From the structure of $H_i, H_{i-1}M_i, \dots, H_h\bar{M}_{h+1:i}$, 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 $\mathbb{H}_{(j)1} = \mathbb{H}_{(j-1)3}, \mathbb{H}_{(j)2} = \mathbb{H}_{(j-1)4}$.

(28) can be satisfied if and only if there exist a matrix $\bar{\xi}$ satisfy the equation below.

$$\begin{bmatrix} \bar{\xi} & I_i \end{bmatrix} \bar{\mathbb{H}}_{h:i} = 0_{N_z(i)\times N_p(i)} \tag{31}$$

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

$$\begin{bmatrix} \bar{\xi} & I_i \end{bmatrix} = \begin{bmatrix} \bar{\xi}_h & \dots & \bar{\xi}_j & \dots & \bar{\xi}_{i-1} & I_i \end{bmatrix}.$$

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

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

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 (31), it is known that (31) 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 $\begin{bmatrix} \bar{\xi} & I \end{bmatrix} \bar{\mathbb{H}}_{u_{h:i}} = 0$.

Based on the derivation above, it is known that

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

$$\text{Dim}(\bar{f}^T_{u(j)}) = m \quad \forall h \leq j \leq i-1$$

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, meaning that there exist no $\bar{\xi}$ make $\begin{bmatrix} \bar{\xi} & I \end{bmatrix} \bar{\mathbb{H}}_{u_{h:i}} = 0_{N_z(i)\times(n-h+1+m)}$. At last, it indicates that there exist no ξ let (28) be satisfied and no ξ let (24) be satisfied.

(b) When $\text{Dim}(\bar{f}^T_{u(j)}) < m$ $\forall h \leq j \leq i-1$ 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

$$\text{Dim}(\text{null}(\bar{\mathbb{H}}_{u_{h:i}})) = \sum_{j=h}^{i-1} \text{Dim}(\text{null}(\bar{f}^T_{u(j)})) \tag{32}$$

Since dimension of null space of column vectors in $\bar{\mathbb{H}}_{h:i}$ should be at least $N_s(i)$, when $\text{Dim}(\bar{\mathbb{H}}_{u_{h:i}})$ calculated in (32) satisfies $\text{Dim}(\bar{\mathbb{H}}_{u_{h:i}}) < N_s(i)$, there exist no $\bar{\xi}$ make (31) be satisfied. Furthermore, there is no ξ_s let (28) be satisfied and no ξ let (24) be satisfied.

□