Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 10601–10607

Inverse Optimal Control for Dynamic Systems with Inequality Constraints

Z.Chen * **T.Baček** ** **D.Oetomo** ** **Y.Tan** ** **D.Kulić** *

* *Department of Electrical and Computer Systems Engineering, Monash University, Clayton, Australia (e-mail: {zhongxiang.chen,dana.kulic}@monash.edu).*

** *School of Electrical, Mechanical and Infrastructure Engineering, The University of Melbourne, Parkville, Australia (e-mail: {tomislav.bacek,doetomo,yingt}@unimelb.edu.au).*

Abstract: Inverse optimal control (IOC) algorithms can be used to reveal underlying objectives. Existing algorithms commonly estimate the objectives by assuming that the cost function can be represented as a weighted sum of features, and use optimality criteria to estimate the weights. However, the existing literature rarely discusses the recovery of cost functions in the presence of state or control constraints, which often exist due to the limited ranges of actuators and sensors. In this work, an optimisation problem is formulated to find the best values of weights and Lagrange multipliers of constraints to satisfy the optimality conditions, given a segment of an optimal trajectory. The maximum and minimum observed state and control variables are hypothesised as potential box constraints and validated by the associated Lagrange multipliers. In addition, this paper also introduces a method to dynamically choose the window size of the observation, or identify that not enough information was provided for an accurate estimation. The proposed approach is validated using simulated results generated with a two link serial arm. The results show the proposed approach can recover the cost function when box constraints are active, and the Lagrange multiplier value can indicate when and which constraints are present.

Keywords: Inverse optimal control (IOC), state and control constraints, Lagrange multipliers.

1. INTRODUCTION

Inverse optimal control (IOC) aims to study the underlying objective function of observed behaviors from an optimal control(OC) system (Ng and Russell, 2000). It has been broadly applied in human behaviour analysis, learning from demonstration, and rehabilitation and assistive robotics (Kulić et al., 2016). Most existing IOC studies assume that the observed system trajectory is generated from a cost function, which is constructed as a linear combination of given features (or basis functions) with unknown weights (Lin et al., 2021).

Recent works have also explored the application of IOC to analyze motion with constraints. In (Clever and Mombaur, 2016; Reiter et al., 2022), the constrained IOC problem was formulated as a Bi-level optimisation problem. The upper level optimises the unknown weights, parameters and constraints, while the lower level computes forward OC trajectories repetitively. The main drawback of the Bi-level structure is the computational cost, especially when model and task complexity increase. On the other hand, (Menner et al., 2021; Molloy et al., 2020) solved the IOC problem in one step by exploiting the Karush–Kuhn–Tucker(KKT) conditions. In (Molloy et al., 2020), the control constraint was assumed known to the IOC algorithm, and the recovery of the cost function is temporarily paused during the segment of motion when the control constraint is active. However, this algorithm could not handle state constraints. In (Menner et al., 2021), the constraints were formulated using the Lagrange method, and the *Candidate Constraint Set* has to be processed from the motion segment before recovering the unknown feature weights and constraints. More importantly, the algorithm solved the inverse of an infinite-horizon optimal control problem, and the optimal cost of the motion data points behind the observed segment was computed and included in the IOC formulation. This method is more computationally efficient but require preprocessing of the entire motion, which hinders the application in online applications. In (Chou et al., 2020), an algorithm that can learn parametric constraints from locally-optimal demonstrations was proposed. The constraint learning problem was formulated as a mixed integer linear program (MILP) using the KKT conditions, focusing on finding a guaranteed safe set of states and generating novel movements. The issue with the algorithm is its post-processing nature which requires multiple and well-informed demonstrations

⋆ This work was supported by the Australian Research Council, Project scheme (DP190100916). D. Kulić was supported by the Australian Research Council Future Fellowship (FT200100761).

10.1016/j.ifacol.2023.10.681

that cover all constraints to recover the relevant cost and constraints.

In this paper, we present an analysis framework for determining both the objective criterion and box constraints of an observed optimal motion, focusing on closing the gap in cost and constraint recovery in online applications. The proposed framework works in the finite-time horizon setting, where the cost function is parameterizable and time-independent and only the most common inequality constraint, box constraint, are considered. The bounds of the box constraints are extracted from the observed motion segment, then an optimisation problem based on the optimality conditions is formulated to solve for the cost function and active constraints simultaneously. The problem is formulated as a convex optimisation problem which is more computationally efficient compared to the Bi-level and MILP formulation. The presence of active constraints is identified by the value of the inequality constraint multipliers. The proposed algorithm works with incomplete observations and the recovery accuracy increases with window size which also increases computational delay in online applications. Therefore, the proposed algorithm also include a method to choose window size dynamically, which balances the trade-off between estimation accuracy and computational demand. The accuracy of the cost and constraint recovery of the proposed approach are evaluated in simulation with a dynamical model of a two-link planar arm.

2. PROBLEM FORMULATION

Let $\mathbb{R}$ and $\mathbb{R}^n$ denote the set of real numbers and an n-dimensional Euclidean space respectively. Let $\boldsymbol{I}$ be the identity matrix and $\mathbf{0}$ be the zero matrix or the zero vector with appropriate dimensions.

2.1 Optimal Control

An OC problem for a class of discrete-time nonlinear dynamic systems consists of the nonlinear dynamics, the cost function, and the constraints.

The nonlinear dynamics have the following form:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \qquad \boldsymbol{x}_0 \in \mathbb{R}^n, \tag{1}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the state at the k^{th} sampling instant. The control input at the k^{th} sampling instant is denoted as $\boldsymbol{u}_k \in \mathbb{R}^m$. The nonlinear mapping $\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is assumed to be smooth.

The cost function indicates how much a certain state-action trajectory costs. In particular, the standard finite-horizon free-end OC with the following cost is defined:

$$J(\bar{\boldsymbol{u}}) = \sum_{k=0}^{N-1} L(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{2}$$

where the optimization variable $\bar{\boldsymbol{u}}$ is defined as

$$\bar{\boldsymbol{u}} = \left[\, \boldsymbol{u}_0 \;\; \boldsymbol{u}_1 \;\; \cdots \;\; \boldsymbol{u}_{N-1} \,\right]^T, \tag{3}$$

and N is the total number of discrete time steps. Here J is the total cost and $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the running cost at each time step. It is assumed that the running cost $L(\cdot)$ satisfies the following assumption:

Assumption 1. The running cost can be parameterized as follows

$$\begin{aligned} L(\boldsymbol{x}_k, \boldsymbol{u}_k) &= \boldsymbol{\omega}^{\mathsf{T}} \boldsymbol{\phi}^*(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ &= [\omega_1 \;\; \omega_2 \;\; \cdots \;\; \omega_p] \begin{bmatrix} \phi_1(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ \phi_2(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ \vdots \\ \phi_p(\boldsymbol{x}_k, \boldsymbol{u}_k) \end{bmatrix}, \end{aligned} \tag{4}$$

where the nonlinear mapping $\boldsymbol{\phi}^* \colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ is smooth, and both $\boldsymbol{\phi}^*(\cdot, \cdot)$ and $\boldsymbol{\omega}$ are known.

Constraints are an essential part in the OC problem formulation when it is implemented in engineering applications as both actuators and sensors have limited capabilities. This includes equality and inequality constraints:

$$\boldsymbol{h}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}) = \mathbf{0}_{q_1}, \tag{5}$$

$$\boldsymbol{g}(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}) \geq \mathbf{0}_{q_2}, \tag{6}$$

where $\bar{\boldsymbol{u}}$ is defined in (3) and $\bar{\boldsymbol{x}}$ is defined as

$$\bar{\boldsymbol{x}} = \left[\, \boldsymbol{x}_1 \;\; \boldsymbol{x}_0 \;\; \cdots \;\; \boldsymbol{x}_N \,\right]^T. \tag{7}$$

The mappings $\boldsymbol{h} \colon \mathbb{R}^{n \times N} \times \mathbb{R}^{m \times N} \to \mathbb{R}^{q_1}$ and $\boldsymbol{g} \colon \mathbb{R}^{n \times N} \times \mathbb{R}^{m \times N} \to \mathbb{R}^{q_2}$ are the **known** collection of equality constraints and inequality constraints respectively.

In this paper, the most common inequality constraints, box constraints, are considered:

$$\boldsymbol{x}_{\text{lb}} \leq \boldsymbol{x}_k \leq \boldsymbol{x}_{\text{ub}}, \qquad k = 1, \ldots, N, \tag{8}$$

$$\boldsymbol{u}_{\text{lb}} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\text{ub}}, \qquad k = 0, \ldots, N-1, \tag{9}$$

where $\boldsymbol{x}_{\text{lb}}, \boldsymbol{x}_{\text{ub}} \in \mathbb{R}^n$ and $\boldsymbol{u}_{\text{lb}}, \boldsymbol{u}_{\text{ub}} \in \mathbb{R}^m$ are the box constraints of state and control variables respectively.

Consequently, the OC can be formulated as

$$\begin{aligned} &\min_{\bar{\boldsymbol{u}} \in \mathbb{R}^{n \times N}} J(\bar{\boldsymbol{u}}) \\ &s.t. \quad \begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), & \boldsymbol{x}_0 \in \mathbb{R}^m, \\ \boldsymbol{x}_{\text{lb}} \leq \boldsymbol{x}_k \leq \boldsymbol{x}_{\text{ub}}, & k = 1, \ldots, N, \\ \boldsymbol{u}_{\text{lb}} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\text{ub}}, & k = 0, \ldots, N-1 \end{cases} \end{aligned} \tag{10}$$

where $J(\cdot)$ is defined in (2), $\boldsymbol{f}(\cdot, \cdot)$ is defined in (1).

Assume that the optimal solution of (10) is $\bar{\boldsymbol{u}}^*$. The associated optimal state is $\bar{\boldsymbol{x}}^*$. The optimal trajectories can be represented as

$$\bar{\boldsymbol{\xi}} = \{\boldsymbol{\xi}_k, \boldsymbol{x}_N^* \,|k = 0, \cdots, N-1\}, \tag{11}$$

where

$$\boldsymbol{\xi}_k := (\boldsymbol{x}_k^*, \boldsymbol{u}_k^*), \tag{12}$$

for $k = 0, \ldots, N-1$.

2.2 Optimality Condition based Inverse Optimal Control

Conversely from OC, in IOC, it is assumed that the optimal trajectory $\bar{\boldsymbol{\xi}}$ is observed, and the problem can be formulated as recovering the weights and/or terms in the cost function J. Given a cost function structure defined in (4), IOC methods commonly assume that the nonlinear mapping $\boldsymbol{\phi}^*(\cdot, \cdot)$ is known and the parameter $\boldsymbol{\omega}$ is unknown. By observing the optimal solution, IOC tries to identify the parameter $\boldsymbol{\omega}$ from the observed optimal solution. As mentioned in (Jin et al., 2021), it is beneficial to infer objective functions from incomplete trajectories when sensor failures or occlusion causes loss of measurements;

the computational cost for complete trajectory with high dimensional complex systems could be high; the capability of learning objective function with incomplete observation might address other problems such as learning time-varying objective functions. An incomplete observation of the motion can be denoted as

$$\boldsymbol{\xi}_{s:s+\ell} := \{\boldsymbol{\xi}_k : s \leq k \leq s+\ell\} \subset \bar{\boldsymbol{\xi}}, \tag{13}$$

for some $s \in [0, N-1]$ satisfying $s+\ell \in [s, N-1]$. Here $\bar{\boldsymbol{\xi}}$ and $\boldsymbol{\xi}_k$ are defined in (11) and (12) respectively. Thus, the proposed algorithm objective is to estimate the $\boldsymbol{\omega}$ and the active inequality constraints $\boldsymbol{x}_{\text{lb}}, \boldsymbol{x}_{\text{ub}}, \boldsymbol{u}_{\text{lb}}$ and $\boldsymbol{u}_{\text{ub}}$ coming from (8) and (9) respectively, given a measured incomplete optimal trajectory $\boldsymbol{\xi}_{s:s+\ell}$.

Next we will revisit how to use $\boldsymbol{\xi}_{s:s+\ell}$ for some $s+\ell \in [0, N-1]$ to estimate the weight $\boldsymbol{\omega}$ without any constraints by using the recovery matrix proposed in (Jin et al., 2021). A new cost function from the given interval $[s, s+\ell]$ is introduced, considering the dynamics as the equality constraint by using the standard Lagrange multiplier:

$$\begin{aligned} &J_L\left(\boldsymbol{x}_{s:s+\ell}, \boldsymbol{u}_{s:s+\ell-1}, \boldsymbol{\lambda}_{s:s+\ell}\right) \\ &= \sum_{k=s}^{s+\ell-1} \left(\boldsymbol{\omega}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{\lambda}_k (\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \boldsymbol{x}_{k+1})\right), \end{aligned} \tag{14}$$

with the optimal parameters to minimize the cost function J_L defined in (14), i.e.,

$$\underset{\substack{\boldsymbol{u}_{s:s+\ell-1},\\ \boldsymbol{\lambda}_{s:s+\ell}}}{\arg\min} \quad J_L\left(\boldsymbol{x}_{s:s+\ell}, \boldsymbol{u}_{s:s+\ell-1}, \boldsymbol{\lambda}_{s:s+\ell}\right). \tag{15}$$

With the optimal solutions $\boldsymbol{\xi}_{s:s+\ell}$ (no additional equality constraints and inequality constraints) and the known feature set, we have the following by applying Pontryagin's maximum principle (Doya et al., 2006),

$$\boldsymbol{\lambda}_{k-1} - \frac{\partial \boldsymbol{f}'}{\partial \boldsymbol{x}_k^*}\boldsymbol{\lambda}_k - \frac{\partial \boldsymbol{\phi}'}{\partial \boldsymbol{x}_k^*}\boldsymbol{\omega} = \boldsymbol{0}, s \leq k \leq s+\ell \tag{16}$$

$$\frac{\partial \boldsymbol{f}'}{\partial \boldsymbol{u}_k^*}\boldsymbol{\lambda}_k + \frac{\partial \boldsymbol{\phi}'}{\partial \boldsymbol{u}_k^*}\boldsymbol{\omega} = \boldsymbol{0}, s \leq k \leq s+\ell \tag{17}$$

where $\boldsymbol{\lambda}_{k-1}$ and $\boldsymbol{\lambda}_k$ is the costate for the OC system. The optimality conditions described in (16) and (17) were the basis of the recovery matrix in (Jin et al., 2021, Definition 1).

3. PROPOSED APPROACH

In this paper, we propose to solve the IOC problem as an optimization problem and incorporate the inequality constraints into the problem formulation using the Lagrange method. This contrasts the common approaches in the literature that formulate the IOC problem as Bi-level optimization problem (Clever and Mombaur, 2016; Reiter et al., 2022) or mixed-integer linear programming (Chou et al., 2020), and closes the gap in optimality condition-based approaches with the presence of unknown state and control box constraints (Jin et al., 2021; Molloy et al., 2020). The proposed method maintains the computational benefit offered by exploiting the optimality conditions. The proposed algorithm works in the finite-time horizon setting and deals with incomplete observations. The minimum window size needed for successful recovery is proposed. Additionally, the value of inequality constraint multipliers indicates the presence of active constraints. The overall algorithm is summarised in Algorithm 1.

Firstly, we modify (14) to incorporate state and control constraints (8) and (9) leading to the following new cost function,

$$\begin{aligned} &\hat{J}_L\left(\boldsymbol{x}_{s:s+\ell}, \boldsymbol{u}_{s:s+\ell-1}, \boldsymbol{\lambda}_{s:s+\ell}, \boldsymbol{\nu}_{\text{sub},k}, \boldsymbol{\nu}_{\text{slb},k}, \boldsymbol{\nu}_{\text{cub},k}, \boldsymbol{\nu}_{\text{clb},k}\right) \\ &= \sum_{k=s}^{s+\ell-1} \left(\boldsymbol{\omega}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{\lambda}_k (\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) - \boldsymbol{x}_{k+1})\right. \\ &\qquad + \boldsymbol{\nu}_{\text{sub},k}(\boldsymbol{x}_{\text{ub}} - (\boldsymbol{\epsilon}_{\text{sub},k} + \boldsymbol{x}_k)) \\ &\qquad + \boldsymbol{\nu}_{\text{slb},k}(\boldsymbol{x}_k - (\boldsymbol{\epsilon}_{\text{slb},k} + \boldsymbol{x}_{\text{lb}})) \\ &\qquad + \boldsymbol{\nu}_{\text{cub},k}(\boldsymbol{u}_{\text{ub}} - (\boldsymbol{\epsilon}_{\text{cub},k} + \boldsymbol{u}_k)) \\ &\qquad \left. + \boldsymbol{\nu}_{\text{clb},k}(\boldsymbol{u}_k - (\boldsymbol{\epsilon}_{\text{clb},k} + \boldsymbol{u}_{\text{lb}}))\right), \end{aligned} \tag{18}$$

where $\boldsymbol{\nu}_{\text{sub},k}$, $\boldsymbol{\nu}_{\text{slb},k}$, $\boldsymbol{\nu}_{\text{cub},k}$, and $\boldsymbol{\nu}_{\text{clb},k}$ are the costates associated with upper and lower bounds of state and control variables at each time step respectively, and $\boldsymbol{\epsilon}_{\text{sub},k}$,$\boldsymbol{\epsilon}_{\text{slb},k}$,$\boldsymbol{\epsilon}_{\text{cub},k}$,and $\boldsymbol{\epsilon}_{\text{clb},k}$ are slack variables of the inequality constraints. The slack variables are postive and costates are $\boldsymbol{0}$ when the inequality constraint is inactive. Then, (18) can be simplified to (14). However, slack variables are $\boldsymbol{0}$ and the costates are positive when the inequality constraints are active.

Applying Pontryagin's maximum principle with the additional multipliers $\boldsymbol{\nu}_{\text{sub},s:s+\ell}$, $\boldsymbol{\nu}_{\text{slb},s:s+\ell}$, $\boldsymbol{\nu}_{\text{cub},s:s+\ell}$, and $\boldsymbol{\nu}_{\text{clb},s:s+\ell}$ associated with inequality constraints, (16) and (17) become

$$\boldsymbol{\lambda}_{k-1} - \frac{\partial \boldsymbol{f}'}{\partial \boldsymbol{x}_k^*}\boldsymbol{\lambda}_k - \frac{\partial \boldsymbol{\phi}'}{\partial \boldsymbol{x}_k^*}\boldsymbol{\omega} + \boldsymbol{\nu}_{\text{sub},k} - \boldsymbol{\nu}_{\text{slb},k} = \boldsymbol{0}, \quad s \leq k \leq s+\ell, \tag{19}$$

$$\frac{\partial \boldsymbol{f}'}{\partial \boldsymbol{u}_k^*}\boldsymbol{\lambda}_k + \frac{\partial \boldsymbol{\phi}'}{\partial \boldsymbol{u}_k^*}\boldsymbol{\omega} - \boldsymbol{\nu}_{\text{cub},k} + \boldsymbol{\nu}_{\text{clb},k} = \boldsymbol{0}, \quad s \leq k \leq s+\ell, \tag{20}$$

Considering (19) and (20), together with the properties of the costates, i.e. $\boldsymbol{\nu}_{\text{sub},s:s+\ell}$ and $\boldsymbol{\nu}_{\text{slb},s:s+\ell}$ can not be non-zero at the same time, we propose that $\boldsymbol{\nu}_{\text{sub},s:s+\ell}$ and $\boldsymbol{\nu}_{\text{slb},s:s+\ell}$ can be summed together in $\boldsymbol{\nu}_{\text{sb},k}$. Similarly, $\boldsymbol{\nu}_{\text{cub},s:s+\ell}$ and $\boldsymbol{\nu}_{\text{clb},s:s+\ell}$ could be lumped together as $\boldsymbol{\nu}_{\text{cb},k}$, which lead to the following expressions

$$\boldsymbol{\lambda}_{k-1} - \frac{\partial \boldsymbol{f}'}{\partial \boldsymbol{x}_k^*}\boldsymbol{\lambda}_k - \frac{\partial \boldsymbol{\phi}'}{\partial \boldsymbol{x}_k^*}\boldsymbol{\omega} + \boldsymbol{\nu}_{\text{sb},k} = \boldsymbol{0}, s \leq k \leq s+\ell, \tag{21}$$

$$\frac{\partial \boldsymbol{f}'}{\partial \boldsymbol{u}_k^*}\boldsymbol{\lambda}_k + \frac{\partial \boldsymbol{\phi}'}{\partial \boldsymbol{u}_k^*}\boldsymbol{\omega} + \boldsymbol{\nu}_{\text{cb},k} = \boldsymbol{0}, s \leq k \leq s+\ell. \tag{22}$$

Prior works (Jin et al., 2021; Molloy et al., 2020) eliminated the costates $\boldsymbol{\lambda}_k$ associated with system dynamics by rearranging and substituting (16) into (17). However, the costates $\boldsymbol{\nu}_{\text{sb},k}$ and $\boldsymbol{\nu}_{\text{cb},k}$ associated with the inequality constraints cannot be eliminated because these are variables we wish to recover to identify the presence of constraints. As a result, there are more unknown variables than the number of equations in (21), leading to non-unique solutions. Therefore, we choose to minimise the residual error instead of solving for the necessary KKT condition equations explicitly:

$$\boldsymbol{\lambda}_{k-1} - \frac{\partial \boldsymbol{f}'}{\partial \boldsymbol{x}_k^*}\boldsymbol{\lambda}_k - \frac{\partial \boldsymbol{\phi}'}{\partial \boldsymbol{x}_k^*}\boldsymbol{\omega} + \boldsymbol{\nu}_{\mathrm{sb},k} = \boldsymbol{\varepsilon}_{x,k}, s \le k \le s+\ell, \tag{23}$$

$$\frac{\partial \boldsymbol{f}'}{\partial \boldsymbol{u}_k^*}\boldsymbol{\lambda}_k + \frac{\partial \boldsymbol{\phi}'}{\partial \boldsymbol{u}_k^*}\boldsymbol{\omega} + \boldsymbol{\nu}_{\mathrm{cb},k} = \boldsymbol{\varepsilon}_{u,k}, s \le k \le s+\ell, \tag{24}$$

where $\boldsymbol{\varepsilon}_{x,k}$, and $\boldsymbol{\varepsilon}_{u,k}$ are the residual error calculated at each time step. Similar approaches were presented in (Chou et al., 2020; Englert et al., 2017; Menner et al., 2021) to deal with noisy input data or sub-optimal demonstrations. Here, we use the residual error to enable the identification of constraints. The IOC problem is then formulated as an optimisation problem

$$\begin{aligned} \underset{\substack{\hat{\boldsymbol{\omega}},\hat{\boldsymbol{\lambda}}_{s:s+\ell},\\ \boldsymbol{\nu}_{\mathrm{sb},s:s+\ell},\\ \boldsymbol{\nu}_{\mathrm{cb},s:s+\ell}}}{\arg\min} \quad & \sum_{k=s}^{s+\ell} \boldsymbol{\varepsilon}_{x,k}^2 + \boldsymbol{\varepsilon}_{u,k}^2 \\ \text{s.t.} \quad & \sum_{i=1}^{p} \hat{\boldsymbol{\omega}}_i = 1, \\ & 0 \le \hat{\boldsymbol{\omega}}_i \le 1, \\ & \boldsymbol{\nu}_{\mathrm{sb},s:s+\ell} \in \mathbb{R}^n, \\ & \boldsymbol{\nu}_{\mathrm{cb},s:s+\ell} \in \mathbb{R}^m. \end{aligned} \tag{25}$$

Next we discuss how to identify the presence of active constraints and how to dynamically chose a suitable window size.

3.1 Constraint Identification

In theory, the inequality constraint multipliers can take different values depending on whether the inequality bound was hit as

$$\boldsymbol{\nu}_{\mathrm{sb},k} \begin{cases} = 0, & x_{\mathrm{lb}} < x_k < x_{\mathrm{ub}} \\ > 0, & x_k = x_{\mathrm{ub}} \\ < 0, & x_k = x_{\mathrm{lb}}. \end{cases} \tag{26}$$

The same rules apply to $\boldsymbol{\nu}_{\mathrm{cb},k}$. In practice, when there are no limitations on the values of $\boldsymbol{\nu}_{\mathrm{sb},s:s+\ell}$ and $\boldsymbol{\nu}_{\mathrm{cb},s:s+\ell}$, and the only objective in (25) is to minimise the sum of residual errors, the additional optimizable variables means the solution is non-unique. Therefore, the following limitations were applied to $\boldsymbol{\nu}_{\mathrm{sb},s:s+\ell}$ and $\boldsymbol{\nu}_{\mathrm{cb},s:s+\ell}$ utilising information from the observed trajectory segment:

$$\boldsymbol{\nu}_{\mathrm{sb},s:s+\ell} \in \begin{cases} [0,0], & \text{if } \hat{x}_{\mathrm{lb}} < x_k < \hat{x}_{\mathrm{ub}} \\ (0,\mathrm{inf}), & \text{if } x_k = \hat{x}_{\mathrm{ub}} \ \&\ \hat{x}_{\mathrm{ub}} = \hat{x}_{\mathrm{lb}} > x_n \\ (-\mathrm{inf},0), & \text{if } x_k = \hat{x}_{\mathrm{lb}} \ \&\ \hat{x}_{\mathrm{ub}} = \hat{x}_{\mathrm{lb}} < x_n, \end{cases} \tag{27}$$

where x_n is the neutral position, $\hat{x}_{\mathrm{ub}}$ and $\hat{x}_{\mathrm{lb}}$ are the hypothesized upper and lower bound of x_k. If x_k is not constant during the analysis segment, $\hat{x}_{\mathrm{ub}}$ and $\hat{x}_{\mathrm{lb}}$ can be extracted directly as the extrema. However, if x_k is constrained for the entire duration of the analysis window, it may not be known whether the constraint is the upper or lower bound. In this case, we utilise a prior, such as knowledge of the neutral position x_n, to assign the constraint to either be upper or lower bound. This prior may be known (e.g., in the case of human motion) or obtained from earlier analysis windows.

With the modified multiplier range, we check the value of $\boldsymbol{\nu}_{\mathrm{sb},s:s+\ell}$ and $\boldsymbol{\nu}_{\mathrm{cb},s:s+\ell}$ to identify whether there is any constraint activated during the observed trajectory segment. For instance, if $\nu_{\mathrm{sb},k}, k \in (s, s+\ell)$ is positive, then the $\hat{x}_{\mathrm{ub}}$ observed is the upper bound of x and an active constraint is detected.

3.2 Dynamic Window Size and Stopping Criteria

The only question left is how to choose a suitable observation length ℓ. Increasing observation length improves the accuracy of cost function estimation, but also increases computational cost and delay. A fixed window size was used and no guidance on the choice of the window size was provided in (Menner et al., 2021). In (Molloy et al., 2020) and (Jin et al., 2021), rank condition was used to select window size. However, (Jin et al., 2021) could not handle any inequality constraints, while (Molloy et al., 2020) only deals with control constraints. We propose to use the residual error, i.e. when $\sum_{k=s}^{s+\ell} \boldsymbol{\varepsilon}_{x,k}^2 + \boldsymbol{\varepsilon}_{u,k}^2$ is greater than the desired residual error level ε_d, to determine the minimum suitable window size. As shown in Algorithm 1, the observation length ℓ is increased until the residual error sum is smaller than ε_d (Successful Recovery, SR) or the end of the trajectories are reached (EoT).

Algorithm 1 IOC with inequality constraint

Input: Trajectory segment $\boldsymbol{\xi}_{s:s+\kappa}$ and feature set $\boldsymbol{\phi}$
Output: Feature weights $\boldsymbol{\omega}$ and active inequality constraint bound values

```
Let ℓ = ℓ_0 and ε_s = ε_0
while ε_s > ε_d & ℓ < κ do
    extract x̂_ub, x̂_lb, û_ub, û_lb from ξ_{s:s+ℓ}
    compute new residual error sum
    ε_s = (Σ_{k=s}^{s+ℓ} ε²_{x,k} + ε²_{u,k}) / ℓ
    ℓ = ℓ + 1
end while
determine active inequality constraints using values of
ν_cb and ν_cb as in Section 3.1
```

4. SIMULATION SETUP

4.1 Multi-rigid-body Dynamics

The dynamic equations of a rigid-body system describe the relationship between the inputs to the system, forces and torques, and the state of the system such as position, velocity and acceleration. Assuming all dynamic parameters of the system are known, a closed-form relationship can be obtained through the Lagrange approach and has the form:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau}, \tag{28}$$

where $\boldsymbol{q} \in \mathbb{R}^m$, $\dot{\boldsymbol{q}} \in \mathbb{R}^m$ and $\ddot{\boldsymbol{q}} \in \mathbb{R}^m$ are the vectors of generalized joint coordinates, velocities and accelerations respectively, $\boldsymbol{M} \in \mathbb{R}^{m\times m}$ is the joint-space inertia matrix, $\boldsymbol{C} \in \mathbb{R}^{m\times m}$ is the Coriolis and centripetal coupling matrix, $\boldsymbol{G} \in \mathbb{R}^m$ is the gravity terms and $\boldsymbol{\tau} \in \mathbb{R}^m$ is the torque vector. The additional torque components caused by friction, backlash and actuators are assumed to be small and neglected, thus not included in (28). Note here, $n = 2 \times m$ for a fully actuated multi-rigid-body dynamics model.

The dynamic equation forms the basis of the state transition equation $\boldsymbol{f}$ in (1) and can be derived as

$$\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{x}_{k+1} = \begin{bmatrix} \boldsymbol{q}_{k+1} \\ \dot{\boldsymbol{q}}_{k+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_k + \dot{\boldsymbol{q}}_k \Delta t \\ \dot{\boldsymbol{q}}_k + \ddot{\boldsymbol{q}}_k \Delta t \end{bmatrix}, \tag{29}$$

where

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{q}_k \\ \dot{\boldsymbol{q}}_k \end{bmatrix}, \tag{30}$$

$$\boldsymbol{u}_k = \boldsymbol{\tau}_k, \tag{31}$$

$$\ddot{\boldsymbol{q}}_k = \boldsymbol{M}(\boldsymbol{q}_k)^{-1}(\boldsymbol{u}_k - \boldsymbol{C}(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k) - \boldsymbol{G}(\boldsymbol{q}_k)), \tag{32}$$

and Δt is the time difference between each time step of the discrete system.

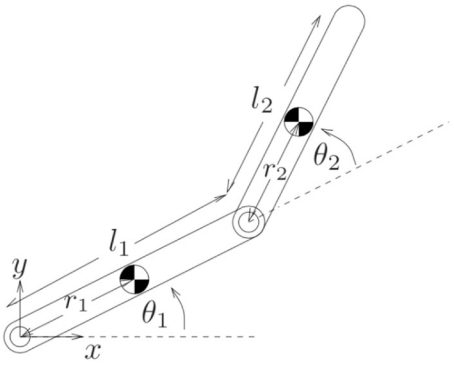

Fig. 1. The schematic of the two-link serial arm model in the simulation (Spong and Vidyasagar, 2004, p.209), the parameters are summarised in Table 1.

Table 1. Parameters of two-link arm model.

m_1	m_2	l_1	l_2	r_1	r^2	I_1	I_2
1	1	1	1	0.5	0.5	0.5	0.5

The units of mass, length and moment of inertia parameters are kg, m, and $kg\ m^2$ respectively.

A two-link serial arm model (Spong and Vidyasagar, 2004, p.209) was used to generate simulated optimal trajectories. The schematic of the model is shown in Fig. 1. The state and control variables of the system are defined as $\boldsymbol{x} = [q_1, q_2, \dot{q}_1, \dot{q}_2]'$ and $\boldsymbol{u} = [\tau_1, \tau_2]'$. Note that, $[\cdot]'$ is the transpose of a vector or matrix.

4.2 Trajectory Generation

The goal of the OC problem is to generate the motion trajectories which get the serial arm from initial state $\boldsymbol{x}_0 = [\pi/2, -\pi/2, 0, 0]'$ to the final position $\boldsymbol{x}_f = [0, 0, 0, 0]'$ in $T = 1$ seconds, and the motion is discretised with $\Delta t = 0.01$. The cost (objective) function takes the form described in (4), with the features set $\boldsymbol{\phi} = [\tau_1^2, \tau_2^2]'$ and the corresponding weights $[0.3, 0.7]'$. The box inequality constraints of states and control variables were applied to generate optimal trajectories with constrained behaviours.

Case 1 No state or control constraints were applied.

Case 2 The state variable $\dot{q}_2$ is bounded within $[-2.5, 2.5]$ radian per second.

Case 3 The state variable $\dot{q}_1$ is bounded within $[-2, 2]$ radian per second in addition to constraints in *Case 2*.

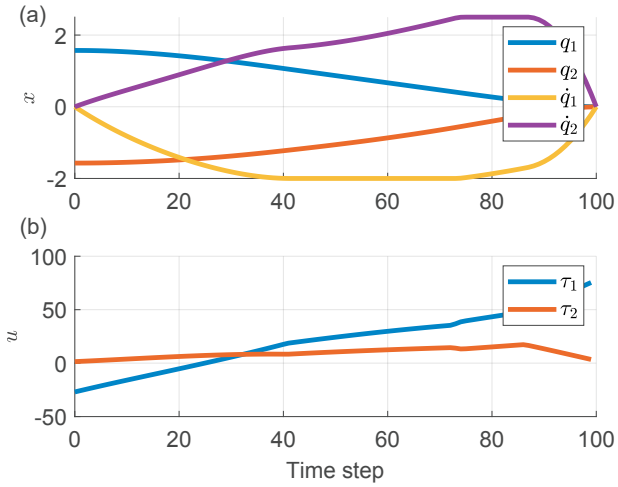

Fig. 2. Optimal solution using the two link model, with state constraints $-2 \leq \dot{q}_1 \leq 2$ and $-2.5 \leq \dot{q}_2 \leq 2.5$.

The optimal trajectories of *Case 3* are shown in Fig. 2.

Three algorithms were implemented to analyse the optimal trajectories generated. Namely, RM, the recovery matrix method (Jin et al., 2021), FW, the proposed algorithm with fixed window size w (similar to the approach proposed in (Menner et al., 2021)), and DW, the proposed algorithm with dynamic window size as shown in Algorithm 1. The optimal trajectories and convex optimisation problem formulated in FW and DW were all solved using CasADi (Andersson et al., 2019).

5. RESULTS

The IOC approach presented in Section 3 was used to recover the objective function weights and constraints assuming the system dynamics and feature set are known. The initial observation length ℓ_0 was set to 10, initial residual error sum ε_0 was set to 1, and desired residual error level ε_d was set to 1e-8. Algorithm 1 was applied repetitively to the motion segments $\boldsymbol{\xi}_{s:s+\kappa}$ where s takes increasing values from 0 to 90, s is the starting time step of the observation and $\kappa = 100 - s$ indicates the end of the trajectory has been reached as $T = 1s$ and $\Delta t = 0.01s$. The estimated weights of $\hat{\boldsymbol{\omega}}$ of the corresponding feature terms $\hat{\boldsymbol{\phi}}$ are shown in Fig. 3. The average of the estimated weights of all feature terms in $\hat{\boldsymbol{\omega}}$ with $\varepsilon_s < \varepsilon_d$ are summarised in Table 2. The comparison with FW with time steps 5 and 20 are shown in Fig. 4 and Fig. 5. The average weights recovered from all algorithms are summarised in Table 2.

In all three cases, the recovery matrix (RM) based IOC algorithm (Jin et al., 2021) can estimate the feature weights when no inequality constraint is active within the observed trajectory segment. This is expected because when inequality constraints are not activated, $\boldsymbol{\nu}_{\text{sub},k}, \boldsymbol{\nu}_{\text{slb},k}, \boldsymbol{\nu}_{\text{cub},k}, \boldsymbol{\nu}_{\text{clb},k}$ are $\mathbf{0}$ and the optimality conditions can be simplified to (16) and (17). However, the RM method (Jin et al., 2021) cannot recover the weights correctly when inequality constraints are active. The number of valid estimates is reduced from 95 for Case 1 to 68 and 47 for Cases 2 and 3 respectively as seen in Table 2.

Table 2. Recovered feature weights

		τ_1^2	τ_1^2	*RMSE*	NE
Ground Truth		0.3	0.7		
Case 1	**DW**	0.3000	0.7000	**4.8453e-09**	90(90)
	FW_5	0.3001	0.6999	5.7978e-04	95
	FW_{10}	0.3000	0.7000	8.1602e-09	90
	FW_{15}	0.3000	0.7000	6.5193e-07	85
	FW_{20}	0.3000	0.7000	9.3064e-07	80
	FW_{25}	0.3000	0.7000	4.5925e-08	75
	RM	0.3000	0.7000	**1.1782e-09**	95(95)
Case 2	**DW**	**0.3000**	**0.7000**	**1.7013e-07**	90(90)
	FW_5	0.2976	0.7024	0.0267	95
	FW_{10}	0.2996	0.7004	0.0040	90
	FW_{15}	0.2999	0.7001	0.0014	85
	FW_{20}	0.2999	0.7001	6.5495e-04	80
	FW_{25}	0.3000	0.7000	3.8225e-04	75
	RM	0.3000	0.7000	**1.4939e-08**	68(95)
Case 3	**DW**	**0.3000**	**0.7000**	**1.2236e-05**	90(90)
	FW_5	0.3024	0.6986	0.0659	95
	FW_{10}	0.2993	0.7007	0.0051	90
	FW_{15}	0.2996	0.7004	0.0037	85
	FW_{20}	0.2996	0.7004	0.0027	80
	FW_{25}	0.2996	0.7004	0.0027	75
	RM	0.3000	0.7000	**2.6082e-09**	47(95)

DW: proposed method using dynamic window size;
FW_w: proposed method with fixed window size w;
RM: recovery matrix method
RMSEs were computed between the recovered weights and the ground truth of feature τ_1^2. This is sufficient because the objective function contains only two feature terms.
NE: the number of valid estimates returned. The number in the brackets is the total number of estimates made.

The proposed method with fixed window size provides better estimates for trajectory segments with active constraints. The distinguishable change in $\boldsymbol{\nu}_{\text{sb},k}$ can be used as the indication of an active constraint, as shown in Fig. 4(b), and Fig. 5(b). Note here that the absolute values of $\boldsymbol{\nu}_{\text{sb},k}$ and $\boldsymbol{\nu}_{\text{cb},k}$ were plotted because the vertical axes of Fig. 3(b), Fig. 4(b) and Fig. 5(b) are in logarithmic scale. It is clear that the accuracy of recovered weights is correlated to residual error and that increasing the window size leads to lower residual errors and a lower root mean squared error (RMSE). However, this leads to fewer estimates and has a higher computational cost. For instance, 95 estimates were returned with window size 5 but only 75 for window size 25.

The proposed method with dynamic window size generates accurate estimates of feature weights for trajectory segments with or without active constraints. The proposed approach terminates when the stopping criteria $\varepsilon_s < \varepsilon_d$ is met or the end of the trajectory has been reached. The value of ε_d governs the trade off between accuracy and computational cost and potential loss of valid estimations, as the lower ε_d leads to larger window size. ε_d was set to 1e-8 for the results shown in Table 2 and Fig. 3. The selection of ε_d improves accuracy comparing to fixed window method without large increase of window size from the default 10 as shown in Fig. 3 (d). The proposed method also produces valid estimates for all segments with active constraints compared to the recovery matrix method. It is worthwhile to note that the estimate results generated when the end of the trajectory is reached would be discarded in Fig. 3(a). Those gaps correspond to the *EoT* flags in Fig. 3(d), where the accuracy criteria $\varepsilon_s < \varepsilon_d$ was not met.

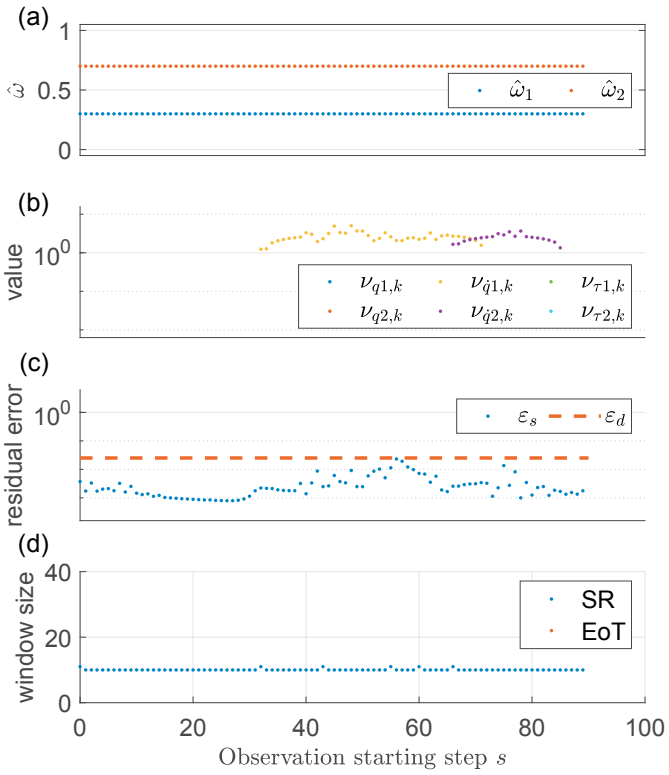

Fig. 3. Recovered feature weights of *Case 3* using proposed IOC method with dynamic window size.

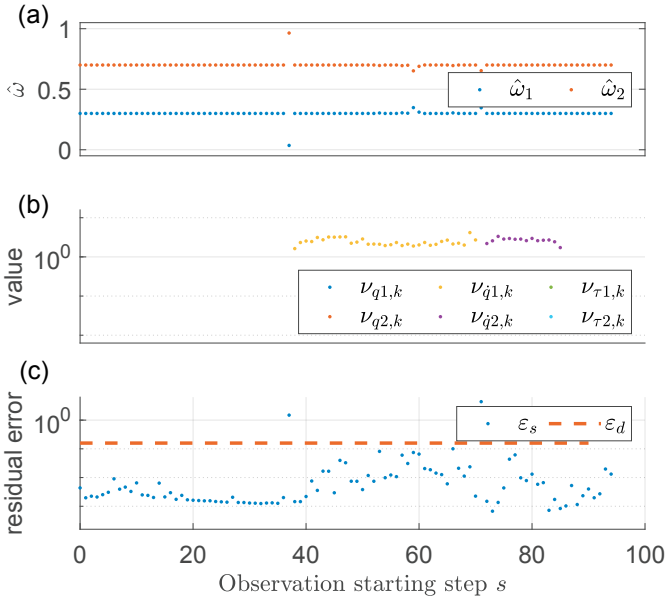

Fig. 4. Recovered feature weights of *Case 3* using proposed IOC method with fixed window size 5.

There are some limitations to the proposed algorithm. Firstly, the box constraint assumption permits a time-decoupled cost function and easy access to the constraint bounds. The proposed approach cannot tackle task space

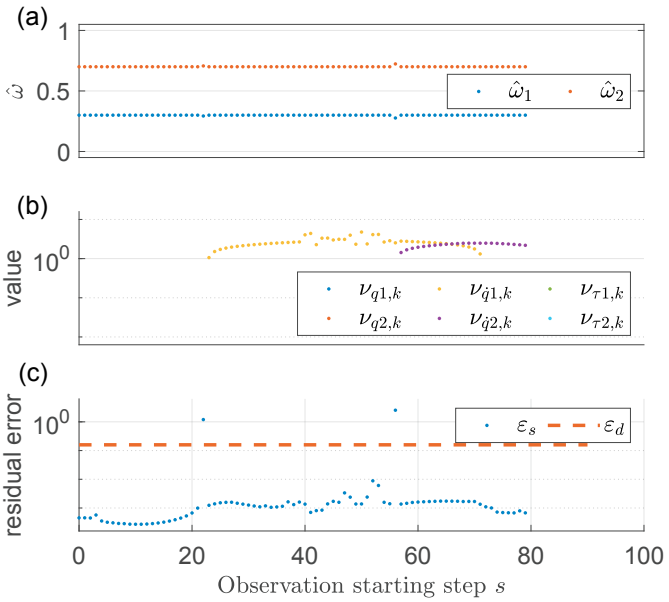

Fig. 5. Recovered feature weights of *Case 3* using proposed IOC method with fixed window size 20.

obstacles which may introduce non-linear constraints and time dependent cost terms as studied in (Chou et al., 2020). Secondly, the performance of the proposed algorithm can be hindered by noise, as with other optimality condition based IOC algorithms (Jin et al., 2021; Molloy et al., 2020). Although the proposed algorithm can handle small magnitudes of added Gaussian noise by increasing observation window size, filtering noise prior to passing the trajectory segment to the algorithm may yield less computational cost. Lastly, the proposed algorithm does not guarantee a unique solution of the cost function and constraints. The dimension of the problem grows linearly with observation length and model complexity, which could result in many local minima that can satisfy the optimality conditions.

6. CONCLUSIONS

In this paper, we developed an inverse optimal control(IOC) algorithm to handle constraints on both state and control variables. Box constraints were incorporated into the IOC problem formulation using the Lagrange method. An optimisation problem was formulated according to optimality conditions of optimal control problems. Normalised residual error was used to automatically select a suitable window size of the incomplete observation. The proposed approach can efficiently recover the objective function and active box constraints from partial trajectories, by formulating a convex optimisation problem. In addition, the adaptive window size can help to balance the trade-off between accuracy and temporal resolution. Our future work will extend the application of the proposed algorithm to analyse trajectories observed in practical settings, which will have more complex dynamical model and suffer from noise.

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36.

Chou, G., Ozay, N., and Berenson, D. (2020). Learning constraints from locally-optimal demonstrations under cost function uncertainty. *IEEE Robotics and Automation Letters*, 5(2), 3682–3690.

Clever, D. and Mombaur, K. (2016). An inverse optimal control approach for the transfer of human walking motions in constrained environment to humanoid robots. In *Robotics: Science and Systems XII*.

Doya, K., Ishii, S., Pouget, A., and Rao, R.P.N. (eds.) (2006). *Bayesian Brain: Probabilistic Approaches to Neural Coding*. Computational Neuroscience Series. MIT Press.

Englert, P., Vien, N.A., and Toussaint, M. (2017). Inverse kkt: Learning cost functions of manipulation tasks from demonstrations. *The International Journal of Robotics Research*, 36(13-14), 1474–1488.

Jin, W., Kulić, D., Mou, S., and Hirche, S. (2021). Inverse optimal control from incomplete trajectory observations. *The International Journal of Robotics Research*, 40(6-7), 848–865.

Kulić, D., Venture, G., Yamane, K., Demircan, E., Mizuuchi, I., and Mombaur, K. (2016). Anthropomorphic movement analysis and synthesis: A survey of methods and applications. *IEEE Transactions on Robotics*, 32(4), 776–795.

Lin, J.F.S., Carreno-Medrano, P., Parsapour, M., Sakr, M., and Kulić, D. (2021). Objective learning from human demonstrations. *Annual Reviews in Control*, 51, 111–129.

Menner, M., Worsnop, P., and Zeilinger, M.N. (2021). Constrained inverse optimal control with application to a human manipulation task. *IEEE Transactions on Control Systems Technology*, 29(2), 826–834.

Molloy, T.L., Ford, J.J., and Perez, T. (2020). Online inverse optimal control for control-constrained discrete-time systems on finite and infinite horizons. *Automatica*, 120, 109109.

Ng, A.Y. and Russell, S.J. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, 663–670. Morgan Kaufmann Publishers Inc.

Reiter, R., Messerer, F., Schratter, M., Watzenig, D., and Diehl, M. (2022). An inverse optimal control approach for trajectory prediction of autonomous race cars. In *2022 European Control Conference (ECC)*, 146–153.

Spong, M.W. and Vidyasagar, M. (2004). *Robot Dynamics and control*. Wiley.