



Reliability of Single-Level Equality-Constrained Inverse Optimal Control

Filip Becanovic, Kosta Jovanović, Vincent Bonnet

► To cite this version:

Filip Becanovic, Kosta Jovanović, Vincent Bonnet. Reliability of Single-Level Equality-Constrained Inverse Optimal Control. 2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids), Nov 2024, Nancy, France. pp.623-630, 10.1109/Humanoids58906.2024.10769923 . hal-04931691

HAL Id: hal-04931691

<https://hal.science/hal-04931691v1>

Submitted on 6 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Reliability of Single-Level Equality-Constrained Inverse Optimal Control

Filip Bećanović¹, Kosta Jovanović¹, Vincent Bonnet²

Abstract— Inverse optimal control (IOC) allows the retrieval of optimal cost function weights, or behavioral parameters, from human motion. The literature on IOC uses methods that are either based on a slow bilevel process or a fast but noise-sensitive minimization of optimality condition violation. Assuming equality-constrained optimal control models of human motion, this article presents a faster but robust approach to solving IOC using a single-level reformulation of the bilevel method and yields equivalent results. Through numerical experiments in simulation, we analyze the robustness to noise of the proposed single-level reformulation to the bilevel IOC formulation with a human-like planar reaching task that is used across recent studies. The approach shows resilience to very large levels of noise and reduces the computation time of the IOC on this task by a factor of 15 when compared to a classical bilevel implementation.

I. INTRODUCTION

Human motion generation is a complex task that requires coordination of the human's many degrees of freedom [1]. A common hypothesis is that there exists a general principle underlying human motor coordination and that this principle can in essence be represented through an objective function within an Optimal Control Problem (OCP) [2].

Determining the underlying optimal principles was done for specific classes of human motion such as reaching [2, 3], grasping [4], or gait [5, 6]. For instance, simulation studies have shown that trajectories resulting from minimum-energy optimal control models for human gait coincide well with actual data on human gait [7]. For grasping, minimum-angular-jerk or minimum-joint-torque-change optimal control models have been shown to coincide best with data [8, 9].

Numerous, objective functions have been proposed to describe human motion [10]. However, the idea that a combination of different objectives could be more accurate than any single one in predicting the measured behavior of a system emerged first in the field of reinforcement learning [11]. It was applied to route planning for outdoor mobile robots, where the desired behavior is often clear, but specifying the reward functions that produce this behavior is much more challenging [12]. Within human motion analysis with the model-based paradigm, Mombaur et al. [5] popularized the so-called Inverse Optimal Control (IOC) approach as a way to learn human-like locomotion, but also as a means of transferring said locomotion laws to robots.

¹Filip Bećanović and Kosta Jovanović are with the School of Electrical Engineering, University of Belgrade, Bul. Kralja Aleksandra 73, 11000 Belgrade, Serbia (filip.becanovic@gmail.com, kostaj@etf.rs).

²Vincent Bonnet is with the Laboratory for Analysis and Architecture of Systems, 7 Avenue du Colonel Roche, 31400 Toulouse, France (vincent.bonnet@laas.fr).

Though many studies focus on determining a stationary [3, 5, 6, 13–15] and deterministic objective function underlying a particular movement, it is not generally believed that this is the case. The objective function may vary through time [16, 17] and may vary across different instances of the motion being performed [18]. However, studies are focusing on this particular problem due to the extremely time-consuming computational effort associated with identifying even a deterministic and stationary objective that would best fit the measured data [19].

The problem of IOC is essentially bilevel: the inner problem represents the optimal-control model of the human motion generation, and the outer problem represents the optimization of the parameters of the motion-generation objective function to minimize differences between the generated and measured motion [15, 20].

Bilevel problems are infamously difficult to solve: even linear-linear bilevel problems are NP-hard [21]. Bilevel problems have existed since the inception of the Stackelberg game in 1936, and can also be expressed as mathematical programs with equilibrium constraints [22]. Most solution methods focus on more manageable bilevel problems: linear-linear bilevel problems, or bilevel problems where the solution map of the inner problem is a well-defined function, rather than a set map as it is in the general case [23].

The bilevel approach has been criticized in the IOC literature [24–26] for taking an extremely long time to compute. An approach, most consistently referred to as Inverse Karush-Kuhn-Tucker (IKKT) [27, 28], had become very popular following its publication [29]. IKKT is essentially equivalent to removing the inner loop and putting the norm of the residual of its KKT conditions as the loss function (9a). The IKKT approach provides a polynomial-time algorithm when the data is not noisy and there exists a behavioral parameter, as defined in Section II-B, that perfectly describes the system's behavior, *i.e.* when the optimal loss function (9a) value is 0. Despite numerous papers using and advocating IKKT [4, 16–18, 24–33], the retrieved optimal weights are rarely used to re-solve the OCP and validate the fitting of human data, with due respect to the exceptions [32].

More recently, studies started showing that IKKT is not robust to noise or modeling error [34–37]. Even a very small amount of noise can lead to a completely different set of optimal recovered weights, as demonstrated practically in the context of IOC with 2 degree-of-freedom planar robot trajectories [34]. Noise is inherent to human motion and it is now clear in the community that the IKKT approach should not be used as it is formulated currently [34–37].

From a theoretical perspective, the statistical inconsistency of this approach in the presence of noisy data was also proven [38, 39].

Research on IKKT drove researchers to examine rank deficiencies in the Jacobian matrix of the KKT conditions with respect to the parameters and dual variables [17, 33, 34], since a rank deficiency would indicate that the IOC problem could be solved. Most recently, a reformulation of the bilevel IOC was proposed using this idea: the inner loop of the bilevel problem is replaced with the condition that the determinant of the Gram matrix of the KKT-Jacobian is equal to zero, which enforces the rank deficiency [35]. Thus, model weights are removed from the search and the noisy human observations are projected onto the optimal solution set of the OCP. Thereafter, the corresponding OCP solution is input into the IKKT approach to retrieve the set of optimal weights.

Unfortunately, computing the determinant constraint [35] and its gradient are also time-consuming, forcing them to reduce the size of the inner-loop search space. Doing so requires representing human trajectories as polynomial functions with the smallest degree required to obtain an over-determined system [35]. So besides computation time, it prevents diverse representations of human trajectories and excludes the use of classical direct-transcriptions of OCPs which lend themselves to efficient solvers [40].

In this context, this paper proposes a single-level reformulation (Section II-C) of the IOC problem that maintains the same properties as the bilevel approach, *i.e.* it will give the same solution and be robust to noisy data [38], but can be solved much more efficiently. The proposed approach was tested in noisy simulations of a 2-degree-of-freedom planar model for a reaching task using human-like cost functions [3] that was also used in recent IOC papers [34, 35].

II. METHODS

A. Optimal Control

A general OCP is given in the Equation (1). This paper explicitly excludes control limits (1d) and path constraints (1e) as they induce edge cases when computing the sensitivity of the OCP solution with respect to parameters of the OCP, which is a central theme in Section II-C. For conciseness, parameters are not mentioned in Equation (1) but are introduced in Section II-B.

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_{t_0}^{t_f} \ell(t, \mathbf{x}(t), \mathbf{u}(t)) dt \quad (1a)$$

$$\text{subject to } \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \quad \forall t \in [t_0, t_f] \quad (1b)$$

$$\mathbf{b}(\mathbf{x}(t_0), \mathbf{x}(t_f)) = 0 \quad (1c)$$

$$\mathbf{u}_{lb} \leq \mathbf{u}(t) \leq \mathbf{u}_{ub} \quad \forall t \in [t_0, t_f] \quad (1d)$$

$$\mathbf{c}(t, \mathbf{x}(t), \mathbf{u}(t)) \leq 0 \quad \forall t \in [t_0, t_f] \quad (1e)$$

Many methods exist to numerically solve OCPs. Here we opt for the simplest direct transcription using Euler integration, where both the state and control trajectories are

considered piece-wise constant,

$$\begin{aligned} t_k &= t_0 + \frac{k}{N}(t_f - t_0) & k \in \{0..N\} \\ \mathbf{x}(t) &= \mathbf{x}_k, \quad t \in [t_k, t_{k+1}] & k \in \{0..N\} \\ \mathbf{u}(t) &= \mathbf{u}_k, \quad t \in [t_k, t_{k+1}] & k \in \{0..N-1\} \end{aligned}$$

and the dynamics equation (1b) becomes a difference equation

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k), \quad k \in \{0..N-1\}. \quad (3)$$

By stacking samples of the states and control in a finite-dimensional vector variable \mathbf{z} , as in (4),

$$\mathbf{z} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N, \mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{N-1})^T \quad (4)$$

stacking boundary conditions (1c) and the newfound dynamics difference equations (3) into a vector constraint, as in (5),

$$\mathbf{h}(\mathbf{z}) = \begin{bmatrix} \mathbf{f}_0(\mathbf{x}_0, \mathbf{u}_0) - \mathbf{x}_1 \\ \vdots \\ \mathbf{f}_{N-1}(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}) - \mathbf{x}_N \\ \mathbf{b}(\mathbf{x}_0, \mathbf{x}_N) \end{bmatrix} \quad (5)$$

and with a slight abuse of notation reformulating the cost functional (1a) into a cost function $f(\mathbf{z})$, we can reduce the OCP (1) to a Nonlinear Programming (NLP) problem (6) with the finite-dimensional vector variable \mathbf{z} .

$$\min_{\mathbf{z}} f(\mathbf{z}) \quad \text{subject to } \mathbf{h}(\mathbf{z}) = 0 \quad (6)$$

B. Inverse Optimal Control

Suppose that there is an NLP (6) that encodes the way a human generates motion, but that $f(\mathbf{z})$ and possibly $\mathbf{h}(\mathbf{z})$ are unknown. As the space of possible functions f and \mathbf{h} is too large to search through, we must introduce parameters θ that encode this behavior into the functions and then search for the best estimate $\hat{\theta}$.

IOC focuses on learning the behavior parameters θ of a system by measuring it. We might have measured the behavior of the system M times, denoting the measured trajectories of this system as $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(M)}$, with $\dim \mathbf{y} = \dim \mathbf{z}$. Moreover, our measurements might have different starting and end states encoded in the boundary conditions (1c), or might in general contain differences in the way the environment is set up. To account for those differences, we introduce a parameter $\mathbf{x}^{(i)}$, which contains aspects of the environment not already captured in the system's state-control trajectory $\mathbf{y}^{(i)}$. Be wary of this abuse of notation - $\mathbf{x}^{(i)}$ is entirely distinct from $\mathbf{x}(t)$ from Section II-A.

We adopt this notation so that it coincides with standard notation from the field of Machine Learning (ML), where input-output pairs are often denoted as $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, and parameters of the model are denoted with θ . The IOC problem is then not distinct from a supervised ML problem. Our NLP system-behavior model (6) now becomes parametrized by behavioral parameters θ and input/environment parameters

x , and our predictions \hat{y} become functions of those, as expressed in Equation (7).

$$\hat{y}(\theta, x) = \underset{z}{\operatorname{argmin}} f(z; \theta, x) \quad (7a)$$

$$\text{subject to } h(z; \theta, x) = 0 \quad (7b)$$

As is the case in supervised ML, we attempt to learn a mapping between the inputs x and outputs y from looking at our data points $(x^{(i)}, y^{(i)})_{i=1}^M$, by minimizing a cumulative loss function (8) that expresses the discrepancy between the model predictions and the measured data. The sample loss function ℓ is often the L_2 distance or the RMSE in IOC studies [15, 19].

$$\varepsilon \left(\hat{y}^{(1)}, \hat{y}^{(2)}, \dots, \hat{y}^{(M)} \right) = \frac{1}{2N} \sum_{i=1}^M \ell(\hat{y}^{(i)}, y^{(i)}) \quad (8)$$

The behavioral learning problem that we call IOC is then naturally expressed as the problem in Equation (9).

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{2N} \sum_{i=1}^M \ell(\hat{y}^{(i)}, y^{(i)}) \quad (9a)$$

$$\text{subject to } \hat{y}^{(i)} = \underset{z}{\operatorname{argmin}} f(z; \theta, x^{(i)}) \quad (9b)$$

$$\text{subject to } h(z; \theta, x^{(i)}) = 0$$

C. Bilevel Optimization and Single-Level Reformulation

The IOC problem (9) is what we call a bilevel optimization problem. Even the simplest classes of bilevel optimization problems are NP-hard [21]. Early literature in IOC used this type of formulation and solved the bilevel IOC problem (9) by using a derivative-free solver to minimize the outer-loop cost function (9a) while generating model predictions in the inner-loop (9b) with a gradient-based solver. This procedure looked somewhat like Algorithm 1, where SOLVE-NG is a generic "gradient-less" solver and SOLVE is a generic gradient-based NLP solver.

Algorithm 1 Bilevel IOC.

Precondition:

Training set: $\mathcal{D} = (x^{(i)}, y^{(i)})_{i=1}^M$

Direct model: $f(z; \theta, x), h(z; \theta, x)$

z -Jacobians: $\partial_z f(z; \theta, x), \partial_z h(z; \theta, x)$

z -Hessians: $\partial_{zz}^2 f(z; \theta, x), \forall j \partial_{zz}^2 h_j(z; \theta, x)$

```

1: function BILEVELIO( $\mathcal{D}$ )
2:    $\hat{\theta} \leftarrow \text{SOLVE-NG}(\text{INNERLOOP}(\theta, \mathcal{D}))$ 
3:   return  $\hat{\theta}$ 
4: end function

5: function INNERLOOP( $\theta, \mathcal{D}$ )
6:    $\varepsilon \leftarrow 0$ 
7:   for  $i \leftarrow 1$  to  $M$  do
8:      $\hat{y}^{(i)}, \hat{v}^{(i)} \leftarrow \text{SOLVE}(f, \partial_z f, \partial_{zz}^2 f, h, \partial_z^2 h, \partial_{zz} h)$ 
9:   end for
10:  return  $\varepsilon$ 
11: end function

```

Single-level reformulations of bilevel optimization problems are well-known in the applied mathematics literature [21–23]. The most convenient single-level reformulation for the bilevel IOC problem (9) is the reformulation based on the KKT conditions of the inner-loop (9b), and consists of replacing the inner-loop (9b) with its KKT conditions, as it allows the implicit computation of the sensitivity of the OCP solution to the behavioral parameters. If the inner-loop had inequality constraints, *i.e.* contained $g(z; \theta, x^{(i)}) \leq 0$, this reformulation would result in a mathematical program with equilibrium constraints [22, 23]. Therefore, we conveniently excluded control limits (1d) and path constraints (1e) from consideration in (1), yielding an NLP (6) and a prediction model (7) without inequality constraints. This is done in almost all the literature on IOC for human motion as it is generally accepted that humans do not operate at their limits during normal motions [32]. To establish the KKT conditions of the inner-loop (9b), we go back to the index-free notation of our prediction model (7). The Lagrangian of the prediction model (10) requires the introduction of Lagrangian multipliers ν for the equality constraints.

$$\mathcal{L}(z, \nu; \theta, x) = f(z; \theta, x) + h(z; \theta, x)^T \nu \quad (10)$$

The first-order necessary conditions of optimality for the prediction model (7) with Lagrangian (10) are given in equation (11).

$$\partial_z \mathcal{L}(z, \nu; \theta, x)^T = 0 \quad (11a)$$

$$h(z; \theta, x) = 0 \quad (11b)$$

We express this compactly as $\mathcal{K}(z, \nu; \theta, x) = 0$ where we define the KKT-vector \mathcal{K} as in equation (12).

$$\mathcal{K}(z, \nu; \theta, x) = \begin{bmatrix} \partial_z f(z; \theta, x)^T + \partial_z h(z; \theta, x)^T \nu \\ h(z; \theta, x) \end{bmatrix} \quad (12)$$

By replacing the inner-loop (9b) by a condition on its KKT-vector (12), the NLP solver of the single-level reformulation will implicitly compute the sensitivity of the optimal solution $\hat{y}^{(i)}$ with respect to the behavioral parameters θ , as is succinctly described in [41]. We will briefly recall this result with a given point $(\bar{z}, \bar{\nu}, \bar{\theta}, \bar{x})$ where $\mathcal{K}(\bar{z}, \bar{\nu}; \bar{\theta}, \bar{x}) = 0$. We want to find the neighboring points θ where this holds

$$\partial_\theta \mathcal{K}(\bar{z}, \bar{\nu}; \bar{\theta}, \bar{x}) = 0. \quad (13)$$

By applying the implicit function theorem,

$$\partial_{z, \nu} \mathcal{K}(\bar{z}, \bar{\nu}; \bar{\theta}, \bar{x}) \partial_\theta(z, \nu) + \partial_\theta \mathcal{K}(\bar{z}, \bar{\nu}; \bar{\theta}, \bar{x}) = 0 \quad (14)$$

and rearranging the terms, we can obtain the sensitivity (15) of the primal-dual solution (z, ν) with respect to the behavioral parameters θ .

$$\partial_\theta(z, \nu) = - (\partial_{z, \nu} \mathcal{K}(\bar{z}, \bar{\nu}; \bar{\theta}, \bar{x}))^{-1} \partial_\theta \mathcal{K}(\bar{z}, \bar{\nu}; \bar{\theta}, \bar{x}) \quad (15)$$

The Jacobian $\partial_{z, \nu} \mathcal{K}(z, \nu; \theta, x)$ of the KKT-vector with respect to primal-dual variables is given in Equation (16),

$$\begin{bmatrix} \partial_{zz}^2 f(z; \theta, x) + \sum_j \nu_j \partial_{zz}^2 h_j(z; \theta, x) & \partial_z h(z; \theta, x)^T \\ \partial_z h(z; \theta, x) & 0 \end{bmatrix} \quad (16)$$

while the Jacobian $\partial_{\theta}\mathcal{K}(\mathbf{z}, \boldsymbol{\nu}; \boldsymbol{\theta}, \mathbf{x})$ of the KKT-vector with respect to behavioral parameters is given in Equation (17).

$$\left[\begin{array}{l} \partial_{\theta}(\partial_{\mathbf{z}}f(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x}))^T + \sum_j \nu_j \partial_{\theta}(\partial_{\mathbf{z}}h_j(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x}))^T \\ \partial_{\theta}\mathbf{h}(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x}) \end{array} \right] \quad (17)$$

By the chain rule, the Jacobian of the loss function with respect to behavioral parameters is easily computed from Equation (18).

$$\partial_{\theta}\ell(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}) = \partial_{\mathbf{z}}\ell(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)})\partial_{\theta}\hat{\mathbf{y}}^{(i)} \quad (18)$$

With this in mind, we reformulate the IOC problem (9) to its single-level variant (19), where one searches for the optimal behavioral parameters, but simultaneously searches for corresponding optimal trajectory predictions $\mathbf{y}^{(i)}$ and Lagrangian multipliers $\boldsymbol{\nu}^{(i)}$.

$$(\hat{\boldsymbol{\theta}}, (\hat{\mathbf{y}}^{(i)}, \hat{\boldsymbol{\nu}}^{(i)})_{i=1}^M) = \quad (19a)$$

$$= \underset{\boldsymbol{\theta}, (\mathbf{z}^{(i)}, \boldsymbol{\nu}^{(i)})_{i=1}^M}{\operatorname{argmin}} \frac{1}{2N} \sum_{i=1}^M \ell(\mathbf{z}^{(i)}, \mathbf{y}^{(i)}) \quad (19b)$$

$$\text{subject to } \mathcal{K}(\mathbf{z}^{(i)}, \boldsymbol{\nu}^{(i)}; \boldsymbol{\theta}, \mathbf{x}^{(i)}) = 0 \quad (19c)$$

Keep in mind that constraint (19c) should hold for all examples $i \in \{1..M\}$. The IOC procedure then reduces to Algorithm 2, where a single-level NLP (19) is solved with objective ε and constraints \mathcal{K} . The algorithm does practically require starting our search from optimal solutions of the inner-loop (9b), so the algorithm contains a call to INNERLOOP from Algorithm 1.

Algorithm 2 Single-level IOC with a gradient-based solver.

Precondition:

Training set: $\mathcal{D} = (\mathbf{x}^{(i)}, \mathbf{y}^{(i)})_{i=1}^M$

Direct model: $f(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x}), \mathbf{h}(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x})$

\mathbf{z} -Jacobians: $\partial_{\mathbf{z}}f(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x}), \partial_{\mathbf{z}}\mathbf{h}(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x})$

\mathbf{z} -Hessians: $\partial_{\mathbf{z}\mathbf{z}}^2 f(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x}), \forall j \partial_{\mathbf{z}\mathbf{z}}^2 h_j(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x})$

$\boldsymbol{\theta}$ -Jacobians: $\partial_{\boldsymbol{\theta}}\mathbf{h}(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x})$

$\partial_{\mathbf{z}}$ -Jacobians: $\partial_{\boldsymbol{\theta}}(\partial_{\mathbf{z}}f(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x}))^T, \forall j \partial_{\boldsymbol{\theta}}(\partial_{\mathbf{z}}h_j(\mathbf{z}; \boldsymbol{\theta}, \mathbf{x}))^T$

```

1: function SINGLELEVELIO( $\boldsymbol{\theta}_0$ ,  $\mathcal{D}$ )
2:    $(\hat{\mathbf{y}}_0^{(i)}, \hat{\boldsymbol{\nu}}_0^{(i)})_{i=1}^M \leftarrow \text{INNERLOOP}(\boldsymbol{\theta}_0, \mathcal{D})$ 
3:    $\hat{\boldsymbol{\theta}}, (\hat{\mathbf{y}}^{(i)}, \hat{\boldsymbol{\nu}}^{(i)})_{i=1}^M \leftarrow \text{SOLVE}(\varepsilon, \mathcal{K})$ 
4:   return  $\hat{\boldsymbol{\theta}}, (\hat{\mathbf{y}}^{(i)}, \hat{\boldsymbol{\nu}}^{(i)})_{i=1}^M$ 
5: end function

```

D. Optimal Control of Robot Arm Trajectory

For a robotic or human arm, the dynamics are expressed in the space of the generalized coordinates, *i.e.*, joint angles \mathbf{q} . These dynamics are usually described by the equation given in (20), where $M(\mathbf{q}_k)$ is the configuration-dependent mass matrix, $C(\mathbf{q}_k, \dot{\mathbf{q}}_k)$ is the vector of centrifugal and Coriolis forces, and $\mathbf{G}(\mathbf{q}_k)$ is the vector of gravitational forces.

$$\tau_k = M(\mathbf{q}_k)\ddot{\mathbf{q}}_k + C(\mathbf{q}_k, \dot{\mathbf{q}}_k) + \mathbf{G}(\mathbf{q}_k) \quad (20)$$

For simplicity's sake, we will equivalently treat the arm as a double integrator like in Equation (21), with τ_k treated

TABLE I: Objective functions considered in this study [3].

Objective Function	Mathematical Expression
Min. joint velocities	$\phi_1 = \sum_{k=0}^{N-2} \dot{\mathbf{q}}_k^T \dot{\mathbf{q}}_k$
Min. joint torques	$\phi_2 = \sum_{k=0}^{N-3} \boldsymbol{\tau}_k^T \boldsymbol{\tau}_k$
Min. end-effector velocity	$\phi_3 = \sum_{k=0}^{N-2} \mathbf{v}_k^T \mathbf{v}_k$
Min. joint jerk	$\phi_4 = \sum_{k=0}^{N-2} \ddot{\mathbf{q}}_k^T \ddot{\mathbf{q}}_k$
Min. joint torque changes	$\phi_5 = \sum_{k=0}^{N-3} \dot{\boldsymbol{\tau}}_k^T \dot{\boldsymbol{\tau}}_k$

as a non-linear function of the state rather than the control signal.

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \Delta t \dot{\mathbf{q}}_k \quad (21a)$$

$$\dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + \Delta t \ddot{\mathbf{q}}_k \quad (21b)$$

Throughout the paper, we focus on an OCP whose goal is to move the arm from an initial joint configuration $\mathbf{q}_i \in \mathbb{R}^n$, towards an end-effector goal location $\mathbf{p}_g \in \mathbb{R}^2$.

The OCP searches for discretized joint positions \mathbf{q} , velocities $\dot{\mathbf{q}}$, and accelerations $\ddot{\mathbf{q}}$, that satisfy Euler integration constraints (21), *i.e.* (22b)-(22c), initial and terminal conditions (22d)-(22e), and minimize some compound objective function (22a) containing \mathbf{q} , $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, $\boldsymbol{\tau}$ and possibly other derived quantities.

$$\min_{\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}} \sum_{j=1}^r \theta_j \phi_j(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \quad (22a)$$

$$\text{subject to } \mathbf{q}_{k+1} = \mathbf{q}_k + \Delta t \dot{\mathbf{q}}_k \quad k \in \{0..N-1\} \quad (22b)$$

$$\dot{\mathbf{q}}_{k+1} = \dot{\mathbf{q}}_k + \Delta t \ddot{\mathbf{q}}_k \quad k \in \{0..N-2\} \quad (22c)$$

$$\mathbf{q}_0 = \mathbf{q}_i \quad (22d)$$

$$\text{FK}(\mathbf{q}_N) = \mathbf{p}_g \quad (22e)$$

This setup maps over naturally to our general description of the prediction model (7) in Section II-B and the KKT-vector (12) and its Jacobians (16)-(17) in Section II-C by observing the transformations in Equation (23),

$$\mathbf{z} = (\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \quad (23a)$$

$$\mathbf{x} = (\mathbf{q}_i, \mathbf{p}_e) \quad (23b)$$

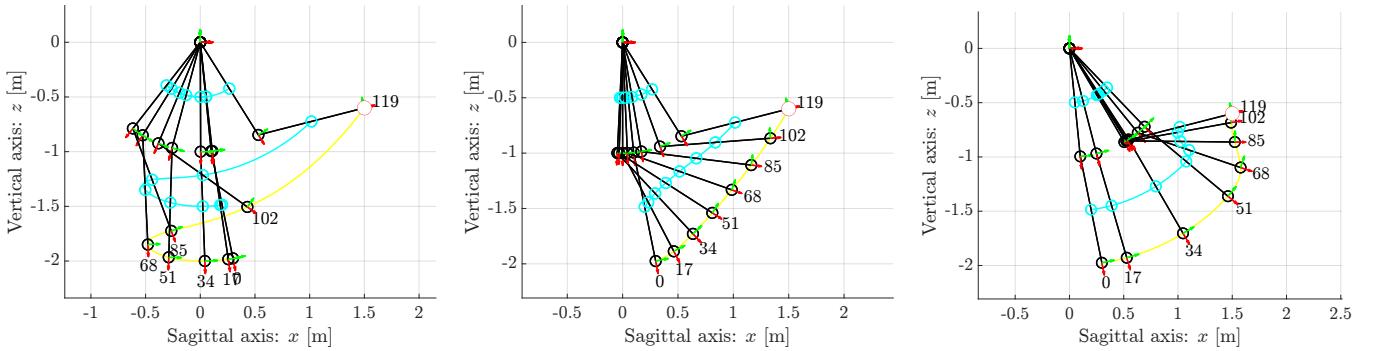
$$\boldsymbol{\theta} = (\theta_1, \dots, \theta_r) \quad (23c)$$

$$f(\mathbf{z}; \boldsymbol{\theta}) = \sum_{j=1}^r \theta_j \phi_j(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \quad (23d)$$

$$\mathbf{h}(\mathbf{z}; \mathbf{x}) = (\dots, \mathbf{q}_0 - \mathbf{q}_i, \text{FK}(\mathbf{q}_N) - \mathbf{p}_g) \quad (23e)$$

where the dots in (23e) stand for the stacking of $\mathbf{q}_{k+1} - \mathbf{q}_k + \Delta t \dot{\mathbf{q}}_k$ for (22b) and $\dot{\mathbf{q}}_{k+1} - \dot{\mathbf{q}}_k + \Delta t \ddot{\mathbf{q}}_k$ for (22c).

Note that the predictive model objective function in (23d) is not a function of the environment parameters \mathbf{x} , and that the equality constraints of the predictive model in (23e) are not functions of the behavioral parameters $\boldsymbol{\theta}$. This setup better reflects the majority of studies in IOC that consider the behavioral parameters θ_j only in the objective function, and in particular as weights of basis objective functions ϕ_j [36, 37].

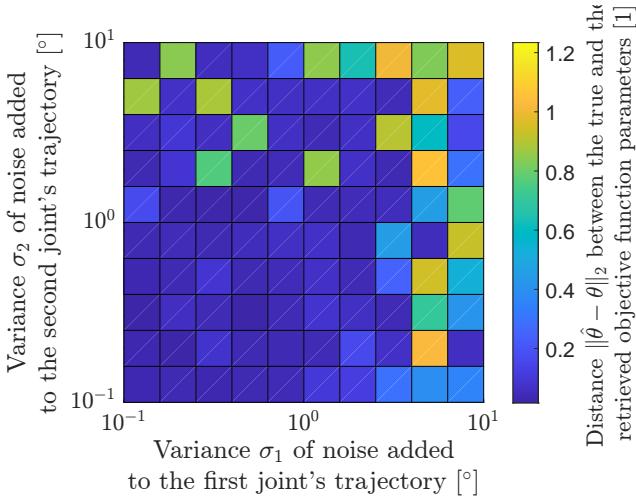


(a) Minimum-torque solution of the robot-arm OCP i.e. $\theta_2 = 1$, $\theta_j = 0$, $j \neq 2$.

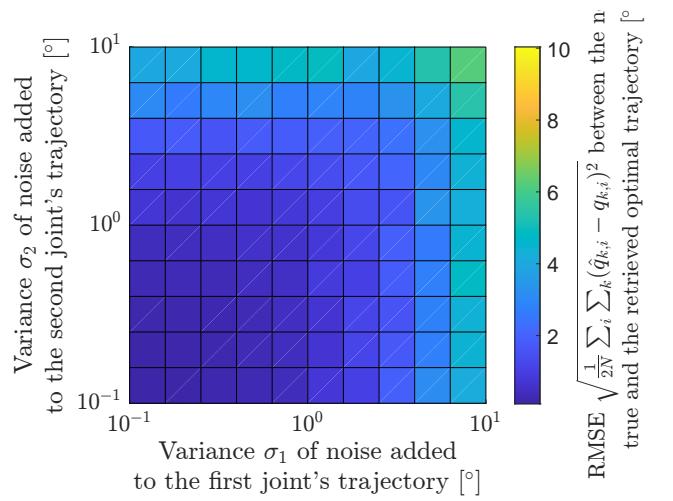
(b) Minimum-end-effector-velocity solution of the robot-arm OCP i.e. $\theta_3 = 1$, $\theta_j = 0$, $j \neq 3$.

(c) Minimum-torque-change solution of the robot-arm OCP i.e. $\theta_5 = 1$, $\theta_j = 0$, $j \neq 5$.

Fig. 1: Solutions of the robot-arm OCP (22) with different objective functions. All important constants defined in Section III-A.



(a) Heatmap of the error in behavioral parameter retrieval.



(b) Heatmap of the error in joint trajectory retrieval.

Fig. 2: Heatmaps of the errors in parameter and trajectory retrieval when solving the single-level IOC problem (19) with noisy data using Algorithm 2.

Many basis objective functions can be found throughout the robotics and biomechanics literature on IOC. Table I uses five of the cost functions proposed by Berret et al. [3] for a similar reaching task.

III. NUMERICAL EXPERIMENTS

A. Optimal Control of Robot Arm Trajectory

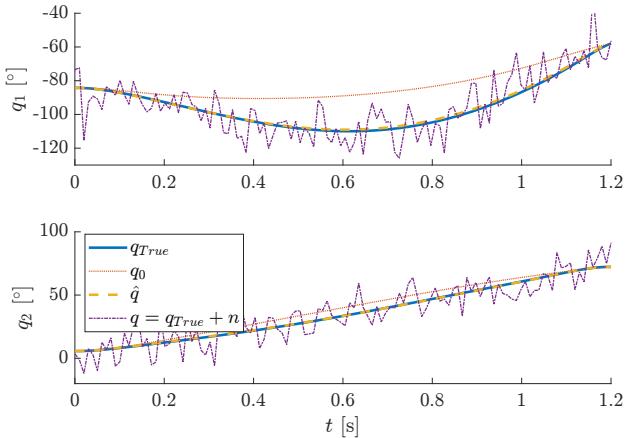
The 2 degrees-of-freedom arm should move from its given initial configuration $q_i = (-\frac{\pi}{2} + 0.1, -0.1)$ rad, analogous to slightly flexed arm hanging from the shoulder, towards a given final configuration $p_e = (1.5, 0.6)$ m, analogous to a position in front of the body where a grasping target may be. The trajectory starts at $t_0 = 0$ s and lasts a fixed amount until $t_f = 1.2$ s. The trajectory is discretized with $N = 120$ samples, so the time-step is equal to $\Delta t = 0.01$ s.

For simplicity, the segments are chosen to be unit-length, unit-mass, uniform-density sticks, so that the geometric and inertial parameters are as simple as possible: the segment

lengths $L = (1, 1)$ m, center-of-mass local-frame positions $C = (0.5, 0.5; 0, 0)$ m, segment masses $M = (1, 1)$ kg, and segment moments of inertia computed at the axis perpendicular to the plane of motion and passing through the center of mass $I = (\frac{1}{12}, \frac{1}{12})$ kg · m².

In Figure 1, the results of solving the OCP given in problem (22) for three different values of θ_j are displayed. Multiple snapshots of the robot arm are shown in various configurations along the trajectory, with sample numbers ranging from 0 to 119. The robot body is drawn in black, while the XY-axes of the reference frames of the robot's joints are represented as red-green arrows. The centers of mass of the segments are depicted as cyan circles, with their trajectory indicated by a cyan line. Finally, the trajectory of the end-effector is shown as a yellow line.

Subfigure 1a represents the minimum-torque trajectory. One can see that the trajectory swings back using gravity to build up acceleration, saving up joint torques. Moreover,



(a) Joint trajectories with the noise level that yielded the maximum trajectory RMSE in the heatmap of Figure 2b, i.e. $\sigma_1 = 10^1$, $\sigma_2 = 10^{-1}$, $\text{RMSE}(\mathbf{q}, \hat{\mathbf{q}}) = 10.03^\circ$.

Fig. 3: Joint trajectories involved in the noisy IOC from Section III-B and the heatmaps from Figure 2 for two different levels of noise.

the centers of mass of the links are brought closer to the vertical line passing through the shoulder joint, aligning the shoulder-to-center of mass line with the gravitational force. This alignment diminishes the torque required to compensate for gravity. Subfigure 1b represents the minimum sum of end-effector velocity trajectory. One can see that the end-effector travels from its initial position to its goal position in a straight line. Subfigure 1c represents the minimum-torque-change trajectory. This trajectory does not produce torque oscillations like the minimum-torque trajectory, while still preventing significant increments in torque usage.

B. Noisy Inverse Optimal Control

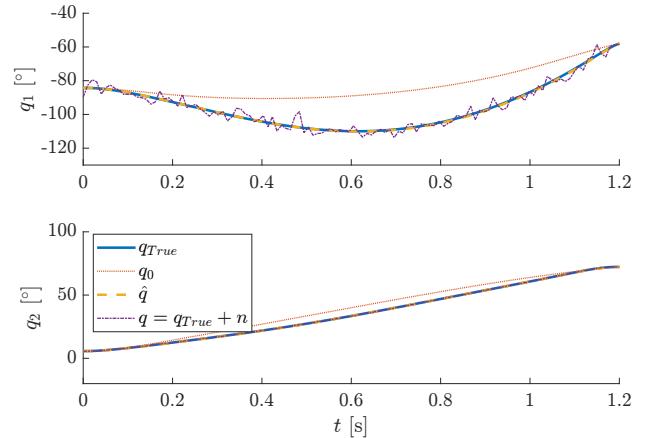
With a given behavioral parameter, and other constants fixed as in Section III-A, we generate an optimal solution $\mathbf{y}^{(1)}$ for the corresponding $\mathbf{x}^{(1)}$ (forming $\mathbf{x}^{(1)}$ described in (23b), numerical values in Section III-A). Select $N_{\text{noise}} = 11$ logarithmically-spaced joint-dependent Gaussian-noise levels and add it to each joint's trajectory. For each noise level, process a single-example data set $(\mathbf{x}^{(1)}, \mathbf{y}^{(1)})$ with Algorithm 2, using a random initial behavioral parameter θ_0 . Obtain a solution to the IOC problem (19) as an estimate $\hat{\theta}$ of the behavioral parameters and $\hat{\mathbf{y}}^{(1)}$ of the measured trajectory.

The IOC loss function (9a) used in this numerical experiment is the quadratic loss in the joint angle space, i.e.

$$\ell(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)}) = \|\hat{\mathbf{q}}^{(i)} - \mathbf{q}^{(i)}\|_2^2. \quad (24)$$

One can extract \mathbf{q} from \mathbf{y} as described in (23a).

Figure 2 displays the results of this analysis. The X-axes of the figures represent the variance of the Gaussian noise added to the trajectory of joint 1, ranging logarithmically from 0.1° to 10° . The Y-axes have the same range but represent the variance of Gaussian noise added to the trajectory of joint 2. Both heatmaps represent measures of the intensity of errors.



(b) Joint trajectories with the noise level that yielded the median trajectory RMSE in the heatmap of Figure 2b, i.e. $\sigma_1 = 10^{0.4}$, $\sigma_2 = 10^{-1}$, $\text{RMSE}(\mathbf{q}, \hat{\mathbf{q}}) = 2.37^\circ$.

In Subfigure 2a, the heatmap represents L_2 distance $\|\hat{\theta} - \theta\|$ between the true behavioral parameter θ and the estimated one $\hat{\theta}$ given the different noise levels, and it is unitless. In Subfigure 2b, the heatmap represents the RMSE, in degrees, between the true trajectory $\mathbf{q}^{(1)}$ and the retrieved trajectory $\hat{\mathbf{q}}$ given the different noise levels. From Subfigure 2a, one can see that few noise levels produce a significant error in behavioral parameter retrieval. From Subfigure 2b, one can see that as the noise levels increase, the error in trajectory retrieval increases linearly.

It is possible to see from comparing Subfigure 2a to Subfigure 2b that for some noise levels the behavioral parameter solving the problem is not unique: the behavioral parameters do not match well but the trajectories do. We attribute this to the poor sensitivity of the optimal solution with respect to parameters when correlated functions are present, e.g. joint velocity and joint jerk. This means that in this region of the behavioral parameter space, varying the parameters does not produce much of a change in the optimal solution, but this small change still fits the noisy data slightly better. This can be considered a case of what is called overfitting in ML. To prevent this, we should train with more than one sample, which we plan to do in future works.

Figure 3 illustrates the joint trajectories of the noisy IOC for two different levels of noise. The full line \mathbf{q}_{True} represents the true target trajectory generated with a given behavioral parameter θ_{True} . The dotted line \mathbf{q}_0 represents the trajectory corresponding to behavioral parameter θ_0 with which we initialize the single-level IOC search, like in Algorithm 2. The dashed-dotted line \mathbf{q} is the noisy target trajectory which we feed into the loss function (9a), obtained from the true target trajectory by adding noise. The dashed line $\hat{\mathbf{q}}$ represents the recovered trajectory from the single-level IOC procedure with Algorithm 2. Figure 3 is meant to complement Figure 2 and strengthen the point that even in the presence of heavy noise, we are able to retrieve a good

fit to the trajectory.

As far as computation time goes, across all noise levels, Algorithm 1 averages 8.240 s to compute the solution while Algorithm 2 averages 0.549 s to compute the solution. The computation time is improved by a factor of 15 with respect to a standard bilevel formulation. In this context, we can conclude that the single-level IOC is fairly robust to noise, contrary to what we referred to earlier as the IKKT procedure [29] which has been shown to have large behavioral parameter and trajectory errors for noise levels equal to 0.1° [34].

IV. CONCLUSION

The objective of this paper was to demonstrate that single-level reformulations of the bi-level IOC work as well in the context of noisy observations while being considerably faster. Moreover, it allows for more general formulations than previous work [35] and is robust to noise contrary to the IKKT method [17, 18]. Though validated in a simulation, the proposed approach is very encouraging as it demonstrated robustness to very large amounts of noise. The only requirement for using this approach, provided the motion-generation OCP model is only equality-constrained, is the C^1 continuity of the motion-generation objective and constraint functions with respect to both the trajectory variables and the behavioral parameters. The huge computational speed-up comes from the fact that the NLP solver in Algorithm 2 can implicitly compute the sensitivity (15) of local solutions of the motion-generation problem with respect to the behavioral parameters and thus reduce the bilevel to a single level problem [41]. Even though inequality constraints are more difficult to handle and they are mostly not active in human motion planning problems, efforts should be made in the direction of the usage of single-level reformulations and algorithms for IOC for inequality-constrained models [15].

Finally, studying the performance of single-level IOC can be extended along multiple axes, most notably for 1) other transcriptions of the OCP than the direct-Euler transcription, 2) other task formulations than the one where the final position of the end-effector is constrained, 3) more extensive sets of basis objective functions than given in I, 4) the use of variable cost function weights as the fast computation time will allow it, and of course 5) the use of real human data.

ACKNOWLEDGEMENT

This work was financially supported by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia under contract number: 451-03-66/2024-03/200103. This work was financially supported by the European Union's Horizon Europe Programme for Research and Innovation through the MUSAE project with grant agreement N° 101070421. This work was supported by the Partenariats Hubert Curien (PHC) Pavle Savić, funded by the French Ministry of Europe and Foreign Affairs and the French Ministry of Higher Education, Research, and Innovation under grant agreement N° 49431WM.

REFERENCES

- [1] N. A. Bernstein, *The Coordination and Regulation of Movements*. Oxford: Pergamon Press, 1967.
- [2] N. Sylla, V. Bonnet, G. Venture, N. Armande, and P. Fraisse, "Human arm optimal motion analysis in industrial screwing task," in *5th IEEE RAS/EMBS international conference on biomedical robotics and biomechatronics*. IEEE, 2014, pp. 964–969.
- [3] 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.
- [4] A. M. Panchea and N. Ramdani, "Inverse parametric optimization in a set-membership error-in-variables framework," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6536–6543, 2017.
- [5] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—an inverse optimal control approach," *Autonomous robots*, vol. 28, pp. 369–383, 2010.
- [6] D. Clever and K. D. Mombaur, "An inverse optimal control approach for the transfer of human walking motions in constrained environment to humanoid robots." in *Robotics: Science and systems*, vol. 12, 2016.
- [7] F. C. Anderson and M. G. Pandy, "Dynamic optimization of human walking," *J. Biomech. Eng.*, vol. 123, no. 5, pp. 381–390, 2001.
- [8] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *Journal of neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.
- [9] Y. Uno, M. Kawato, and R. Suzuki, "Formation and control of optimal trajectory in human multijoint arm movement," *Biological cybernetics*, vol. 61, no. 2, pp. 89–101, 1989.
- [10] M. Febrer-Nafria, A. Nasr, M. Ezati, P. Brown, J. M. Font-Llagunes, and J. McPhee, "Predictive multibody dynamic simulation of human neuromusculoskeletal systems: a review," *Multibody System Dynamics*, vol. 58, no. 3, pp. 299–339, 2023.
- [11] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning," in *Icmi*, vol. 1, no. 2, 2000, p. 2.
- [12] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 729–736.
- [13] S. Albrecht, K. Ramirez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz, "Imitating human reaching motions using physically inspired optimization principles," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2011, pp. 602–607.
- [14] S. Albrecht, M. Leibold, and M. Ulbrich, "A bilevel optimization approach to obtain optimal cost functions for human arm movements," *Numerical Algebra, Control and Optimization*, vol. 2, no. 1, pp. 105–127, 2012.
- [15] 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.
- [16] J. F.-S. Lin, V. Bonnet, A. M. Panchea, N. Ramdani, G. Venture, and D. Kulić, "Human motion segmentation using cost weights recovered from inverse optimal control," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 1107–1113.
- [17] W. Jin, D. Kulić, J. F.-S. Lin, S. Mou, and S. Hirche, "Inverse optimal control for multiphase cost functions," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1387–1398, 2019.
- [18] K. Westermann, J. F.-S. Lin, and D. Kulić, "Inverse optimal control with time-varying objectives: application to human jumping movement analysis," *Scientific reports*, vol. 10, no. 1, p. 11174, 2020.
- [19] I. Maroger, O. Stasse, and B. Watier, "Inverse optimal control to model human trajectories during locomotion," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 25, no. 5, pp. 499–511, 2022.
- [20] K. Mombaur and D. Clever, "Inverse optimal control as a tool to understand human movement," *Geometric and numerical foundations of movements*, pp. 163–186, 2017.
- [21] S. Dempe, *Foundations of bilevel programming*. Springer Science & Business Media, 2002.
- [22] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.
- [23] J. F. Bard, *Practical bilevel optimization: algorithms and applications*. Springer Science & Business Media, 2013, vol. 30.

- [24] N. Aghasadeghi, A. Long, and T. Bretl, “Inverse optimal control for a hybrid dynamical system with impacts,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 4962–4967.
- [25] 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.
- [26] A. M. Panchea and N. Ramdani, “Towards solving inverse optimal control in a bounded-error framework,” in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 4910–4915.
- [27] P. Englert, N. A. Vien, and M. Toussaint, “Inverse kkt: Learning cost functions of manipulation tasks from demonstrations,” *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1474–1488, 2017.
- [28] P. Englert and M. Toussaint, “Learning manipulation skills from a single demonstration,” *The International Journal of Robotics Research*, vol. 37, no. 1, pp. 137–154, 2018.
- [29] 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.
- [30] A.-S. Puydupin-Jamin, M. Johnson, and T. Bretl, “A convex approach to inverse optimal control and its application to modeling human locomotion,” in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 531–536.
- [31] 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)*. IEEE, 2014, pp. 6018–6025.
- [32] A. M. Panchea, N. Ramdani, V. Bonnet, and P. Fraisse, “Human arm motion analysis based on the inverse optimization approach,” in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. IEEE, 2018, pp. 1005–1010.
- [33] W. Jin, D. Kulić, S. Mou, and S. Hirche, “Inverse optimal control from incomplete trajectory observations,” *The International Journal of Robotics Research*, vol. 40, no. 6-7, pp. 848–865, 2021.
- [34] J. Colombel, D. Daney, and F. Charpillet, “On the reliability of inverse optimal control,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8504–8510.
- [35] ———, “Holistic view of inverse optimal control by introducing projections on singularity curves,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 240–12 246.
- [36] F. Bečanović, J. Miller, V. Bonnet, K. Jovanović, and S. Mohammed, “Assessing the quality of a set of basis functions for inverse optimal control via projection onto global minimizers,” in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 7598–7605.
- [37] F. Bečanović, V. Bonnet, R. Dumas, K. Jovanović, and S. Mohammed, “Force sharing problem during gait using inverse optimal control,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 872–879, 2022.
- [38] A. Aswani, Z.-J. Shen, and A. Siddiq, “Inverse optimization with noisy data,” *Operations Research*, vol. 66, no. 3, pp. 870–892, 2018.
- [39] A. Aswani, “Statistics with set-valued functions: applications to inverse approximate optimization,” *Mathematical Programming*, vol. 174, no. 1, pp. 225–251, 2019.
- [40] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, “Crocoddyl: An efficient and versatile framework for multi-contact optimal control,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2536–2542.
- [41] S. Barratt, “On the differentiability of the solution to convex optimization problems,” *arXiv preprint arXiv:1804.05098*, 2018.