Time Dependent Inverse Optimal Control using Trigonometric Basis Functions

Rahel Rickenbach RRAHEL@ETHZ.CH
Elena Arcari EARCARI@ETHZ.CH
Melanie N. Zeilinger MZEILINGER@ETHZ.CH
ETH Zurich

Editors: N. Matni, M. Morari, G. J. Pappas

Abstract

The choice of objective is critical for the performance of an optimal controller. When control requirements vary during operation, e.g. due to changes in the environment with which the system is interacting, these variations should be reflected in the cost function. In this paper we consider the problem of identifying a time dependent cost function from given trajectories. We propose a strategy for explicitly representing time dependency in the cost function, i.e. decomposing it into the product of an unknown time dependent parameter vector and a known state and input dependent vector, modelling the former via a linear combination of trigonometric basis functions. These are incorporated within an inverse optimal control framework that uses the Karush–Kuhn–Tucker (KKT) conditions for ensuring optimality, and allows for formulating an optimization problem with respect to a finite set of basis function hyperparameters. Results are shown for two systems in simulation and evaluated against state-of-the-art approaches.[1]

Keywords: Inverse Optimal Control, Trigonometric Basis Functions, Time Dependence.

1. Introduction

The performance of an optimization-based controller significantly depends on the ability to encode the desired goal in the cost function. This can be challenging in several scenarios, e.g. in the context of autonomous driving Kuderer et al. (2015), or for biomedical applications Khodaei et al. (2020). Particularly for the latter, the ability to specify changes in the control requirements that reflect, e.g., variations of the human body in time, is an essential condition for successfully fulfilling the task at hand. For instance, maintaining the basal insulin requirement is a concrete example that shows both increased variability during day and night time, and time-varying profiles over the day Ruan et al. (2016); Scheiner and Boyer (2005). Among the non-biomedical applications, another example is given by the optimal control with respect to continuously changing electricity prices Mbungu et al. (2017). Motivated by these examples, in this paper we tackle the problem of specifying a cost function that explicitly depends on time. To this end, we build on the inverse optimal control (IOC) approach presented in Menner et al. (2019), which encodes optimality conditions by exploiting the Karush–Kuhn–Tucker (KKT) conditions Kuhn and Tucker (2014) and Bellman's Principle of Optimality Bellman (1966). Within this framework, we incorporate the learning of *time features* that we model as trigonometric basis functions, taking inspiration from Lázaro-Gredilla et al. (2010)

[1] The datasets generated and/or analysed during the current study are available in the eth research collection repository, https://doi.org/10.3929/ethz-b-000611670.

and Arcari et al. (2021): their inherent nonlinearity provides a sufficiently rich description of the objective's time dependency, while preserving the efficiency of parametric models.

The objective is assumed to decompose as the product of an unknown continuous-time parameter vector, and a known state and input dependent vector of arbitrary structure. The unknown time vector is defined as a linear combination of trigonometric basis functions, which allows for formulating the time feature extraction as an optimization problem with respect to a finite set of hyperparameters, e.g. sinusoidal frequencies. These are jointly optimized with the Lagrangian variables arising in the IOC problem formulation. Consequently, the proposed algorithm solves the Lagrangian optimization problem regarding the unknown cost function parameters, as well as a line search over a regularization term for obtaining a sparse solution. Due to non-convexity of the optimization problem, a grid search over initial conditions is performed. Results are shown for two simulation examples, i.e. a multi-layer spring-damper system, and an inverted double pendulum, for which performance is evaluated against state-of-the-art approaches. Furthermore, the reliability of our estimates is investigated by varying both sampling times and horizon lengths of the forward control problem.

The preliminaries are summarized in Section 2, and the problem description is given in Section 3. This is followed by an explanation of the proposed approach in Section 4; while the first subsection is devoted to the introduction of the considered learning approach, the final algorithm follows in Section 4.2. Results are gathered in Section 5, and we conclude the paper with a discussion in Section 6.

1.1. Related Work

The use of optimal demonstrations to learn the parameters of an initially unknown objective has been widely addressed in the literature Lin et al. (2021), both from a reinforcement learning and an optimal control perspective Ab Azar et al. (2020). Inverse reinforcement learning (IRL) is often formulated for discrete state and action spaces, and mainly focuses on the reconstruction of either a reward function or a policy from expert demonstrations, and therefore is also considered in the framework of *learning from demonstrations* or *imitation learning* Ng et al. (2000), Ramachandran and Amir (2007), Arora and Doshi (2021). On the other hand, IOC exploits the structure of the forward optimization problem to formulate the corresponding cost function learning approach by typically using stability and/or optimality conditions. Regarding the latter, a large variety of objective estimation techniques were developed: for example, the works of Priess et al. (2014) and Menner and Zeilinger (2018) are inspired by the solution of an LQR problem, while Li et al. (2011) relies on the Hamiltonian Jacobi Bellman equation to estimate the unknown objective. Overall, these approaches focus on infinite-horizon optimization problems without considering the presence of potential constraints. An IOC approach including inequality constraints was presented in Englert et al. (2017) by exploiting KKT optimality conditions. Building on this idea, the method presented in Menner et al. (2019) additionally exploits Bellman's Principle of Optimality in order to solve the original infinite-horizon formulation using finite-length demonstration trajectories. Other extensions focus on the consideration of uncertain data and noise, e.g. in Menner and Zeilinger (2020), and also on the estimation of time-varying objectives Jin et al. (2019), Westermann et al. (2020), Lin et al. (2016). While in these works time dependency is integrated by either separating the considered time horizon into windows with constant parameters, or by averaging over multiple windows, the method presented in this paper explicitly models time as a cost function feature, while preserving the consideration of constraints. The approach of estimating individual cost parameters at each time step, as well as the non-parametric KKT approach, both presented in Englert et al. (2017) also allow for learning time-varying cost functions in the finite-horizon setting. However, time dependency is incorporated via a time-varying vector of parameters, whose length matches the duration of the task at hand, and can therefore potentially become prohibitively long. In this work, the use of time features bypasses this issue since it directly provides a *model* for the relation between cost function and time to be used in the infinite-horizon setting.

1.2. Notation

Throughout the paper $\|\cdot\|$ indicates the Euclidean norm, while $|\cdot|$ indicates the 1-norm. When applied to a matrix, the latter refers to the sum over all absolute values of its elements. The set of all non-negative real numbers is indicated with $\mathbb{R}_+$.

2. Preliminaries

In this work, we build upon the *shortest path* IOC (spIOC) approach developed by Menner et al. (2019), considering $D > 0$ finite-length observations provided by a demonstrator. These are assumed to be optimal trajectory segments of the original infinite-horizon constrained optimal control problem, and consist of state and input measurements at time instances $k \in \mathbb{N}$, with $x_d^*(k) \in \mathbb{R}^n$ and $u_d^*(k) \in \mathbb{R}^m$, $d \in [1, D]$, collected for different initial conditions $x_d^*(k)$ over a horizon $N \in \mathbb{N}$. The resulting sequences, indicated by $\mathcal{X}_d^* = \{x_d^*(k), \dots, x_d^*(k+N)\}$ and $\mathcal{U}_d^* = \{u_d^*(k), \dots, u_d^*(k+N-1)\}$, obey the potentially nonlinear but known dynamics $x(k+1) = f(x(k), u(k))$. Furthermore, they (at least locally) optimally solve the following problem

$$\min_{u_i} \sum_{i=0}^{N-1} \ell(F_i(U, x_0), u_i, L) \tag{1a}$$

$$g_p(F_i(U, x_0), u_i) \leq 0, \quad p = 1, \dots, P \tag{1b}$$

$$x_0 = x_d^*(k) \tag{1c}$$

$$x_N = x_d^*(k+N), \tag{1d}$$

whose objective $\ell(\cdot, \cdot, L)$ includes an unknown, but constant parameter-vector L. It can also include P known inequality constraints g. The terminal equality constraint (1d), in accordance with Bellman's Principle of Optimality, allows to formulate the infinite-horizon problem as a shortest path problem of finite length N. Additionally, the operator $F_i(U, x_0)$, with $U = [u_0, \dots, u_{N-1}]$, is defined as

$$F_i(U, x_0) = \begin{cases} x_0 & if\ i = 0, \\ f(F_{i-1}(U, x_0), u_{i-1}) & if\ i \geq 1, \end{cases} \tag{2}$$

which allows for elimination of the associated equality constraint at each time-step, and an expression of the optimization problem in terms of U. Introducing the Lagrange multipliers $\lambda_i \in \mathbb{R}^p$ and $\upsilon \in \mathbb{R}^n$, as well as replacing x_0 with $x^*(k)$, the Lagrangian of problem (1) is given by

$$\begin{aligned} \mathcal{L}(U, \lambda_i, \upsilon, L, x^*(k), x^*(k+N)) &= \upsilon^\top \cdot (F_N(U, x^*(k)) - x^*(k+N)) \\ &+ \sum_{i=0}^{N-1} \ell(F_i(U, x^*(k)), u_i, L) + \lambda_i^\top \cdot g(F_i(U, x^*(k)), u_i). \end{aligned} \tag{3}$$

Finally, to allow for the consideration of potentially sub-optimal data due to noisy observations, the following KKT-based optimization problem is solved with respect to the unknown parameter L

$$\begin{aligned} &\min_{L,\lambda_{d,i},\upsilon_d} \sum_{d=1}^{D} \|\nabla_U \mathcal{L}(U,\lambda_{d,i},\upsilon_d,L,x^*(k),x^*(k+N))|_{U=\mathcal{U}_d^*}\|^2 \\ &\quad \lambda_{d,i,p}\cdot g_p(F_i(\mathcal{U}_d^*,x_d^*(k)),u_d^*(i)) = 0,\;\; p=1,\dots,P,\; d=1,\dots,D \\ &\quad \lambda_{d,i,p} \geq 0,\;\; p=1,\dots,P,\; d=1,\dots,D. \end{aligned} \tag{4}$$

3. Problem Description

For the remainder of this paper, the available demonstrations are separated into training and validation data. For this purpose, we define $\mathcal{S}_t^* = \{(\mathcal{X}_1^*,\mathcal{U}_1^*),\dots,(\mathcal{X}_{D_t}^*,\mathcal{U}_{D_t}^*)\}$ and $\mathcal{S}_v^* = \{(\mathcal{X}_{D_t+1}^*, \mathcal{U}_{D_t+1}^*),\dots,(\mathcal{X}_{D_t+D_v}^*,\mathcal{U}_{D_t+D_v}^*)\}$ as training and validation data sets respectively, with D_d and D_v indicating the number of sequences in each set. All of the collected sequences are assumed to be the result of an optimization problem with a time dependent objective, which we express by defining the parameter L in (1) as a function depending continuously on time $t \in \mathbb{R}$. Furthermore, we make the following assumption regarding the structure of the objective:

Assumption 1 *It is assumed that the partially unknown objective function ℓ is convex for each fixed time instance t, and can be decomposed as*

$$\ell(x_i,u_i,L(t)) = \Theta(t)\cdot\phi(x_i,u_i), \tag{5}$$

where $\Theta(t) = [\theta_1(t),\dots,\theta_q(t)] \in \mathbb{R}^{1\times q}$ describes a q dimensional row vector of unknown continuous, time dependent functions $\theta_1(t),\dots,\theta_q(t)$. The column vector $\phi(x_i,u_i) \in \mathbb{R}^q$ is of equal dimension and known.

Remark 1 *The structure in (5) allows for flexible cost function choices. Convexity of $\phi(x_i,u_i)$ can ease the formulation of the associated IOC problem, but the assumption is not a strict requirement (see Remark 3). Note that when $\Theta(t)$ is a constant vector and $\phi(x_i,u_i)$ includes squares of states and inputs, the standard quadratic cost function is obtained.*

The time dependent formulation of the shortest path IOC optimization problem presented in Section 2 results in

$$\min_{x_i,u_i} \sum_{i=0}^{N-1} \Theta(t)\cdot\phi(F_i(U,x_0),u_i) \tag{6a}$$

$$g(F_i(U,x_0),u_i) \leq 0 \tag{6b}$$

$$x_0 = x_d^*(k) \tag{6c}$$

$$x_N = x_d^*(k+N), \tag{6d}$$

for which we assume to know all involved constraints. Similarly to (4), we can obtain an estimate of $\Theta(t)$ by solving the following optimization problem using the training data in $\mathcal{S}_t^*$

$$\begin{aligned} &\min_{\Theta(t),\lambda_{d,i},\upsilon_d} \sum_{d=1}^{D_t} \|\nabla_U \mathcal{L}(U,\lambda_{d,i},\upsilon_d,\Theta(t),x_d^*(k),x_d^*(k+N))|_{U=\mathcal{U}_d^*}\|^2 \\ &\quad \lambda_{d,i,p}\cdot g_p(F_i(\mathcal{U}_d^*,x_d^*(k)),u_d^*(i)) = 0,\;\; p=1,\dots,P,\; d=1,\dots,D_t \\ &\quad \lambda_{d,i,p} \geq 0,\;\; p=1,\dots,P,\; d=1,\dots,D_t, \end{aligned} \tag{7}$$

with its Lagrangian defined as

$$\mathcal{L}(U, \lambda_{d,i}, \upsilon_d, \Theta(t), x_d^*(k), x_d^*(k+N)) = \upsilon_d^\top \cdot (F_N(U, x_d^*(k)) - x_d^*(k+N)) + \\ \sum_{i=0}^{N-1} \Theta(t) \cdot \phi(F_i(U, x_d^*(k)), u_i) + \lambda_{d,i}^\top \cdot g(F_i(U, x_d^*(k)), u_i). \tag{8}$$

Optimizing over a vector of unknown continuous, time dependent functions $\theta_1(t), \dots, \theta_q(t)$ results in an intractable problem. For this reason, in the following section, we define a model for $\Theta(t)$ consisting of a linear combination of trigonometric basis functions, so that problem (7) can be reformulated as an optimization with respect to a finite set of hyperparameters, i.e. the sinusoids' frequencies. This allows for overcoming the intractability of estimating a (potentially very long) sequence of time-varying parameters and provides an efficient framework for identifying the relation between cost and time.

The quality of the estimated parameter $\hat{\Theta}(t)$ is evaluated by solving the forward optimization control problem fixing $\hat{\Theta}(t)$, for each initial state $x_d^*(k),\ d \in [D_t+1, D_t+D_v]$, in the validation set $\mathcal{S}_v^*$. The optimized sequences are indicated with $\hat{\mathcal{X}}_d = \{\hat{x}_d(k), \dots, \hat{x}_d(k+N)\}$ and $\hat{\mathcal{U}}_d = \{\hat{u}_d(k), \dots, \hat{u}_d(k+N-1)\}$, and collected in $\hat{\mathcal{S}}_v = \{(\hat{\mathcal{X}}_{D_t+1}, \hat{\mathcal{U}}_{D_t+1}), \dots, (\hat{\mathcal{X}}_{D_t+D_v}, \hat{\mathcal{U}}_{D_t+D_v})\}$. Consequently, the validation error is defined as an averaged root mean square error between the optimized sequences in $\hat{\mathcal{S}}_v$ and the original validation sequences in $\mathcal{S}_v^*$

$$e_v(\hat{\mathcal{S}}_v, \mathcal{S}_v^*) = \frac{1}{D_v} \sum_{d=D_t+1}^{D_t+D_v} \Big(\frac{1}{N} \sum_{k=0}^{N-1} \|\hat{x}_d(k) - x_d^*(k)\|^2 + \|\hat{u}_d(k) - u_d^*(k)\|^2\Big)^{\frac{1}{2}}. \tag{9}$$

The Intuition of Using a "Sliding-window" Approach and Where it Fails

In the previous section, we provided a raw formulation of the time dependent spIOC in (7), and discussed its intractability when optimized as a batch problem. A naive approach for overcoming this consists in solving (7) sequentially, i.e. using a linear Kalman filter to estimate the time dependent parameters $\Theta(t)$, exploiting the fact that the result in Menner et al. (2019) allows for partitioning the demonstrator sequences in small windows. The idea is to use a "sliding window" approach, in which we iteratively re-estimate a portion $\theta_M \in \mathbb{R}^{1\times M}$ of the vector $\Theta(t)$. We assume that the estimation window θ_M stays constant over the window length M, and choose its length in order to keep the estimation problem well-defined. The following example aims at offering an intuitive explanation of when this approach fails, therefore motivating the choice of learning a model for time dependency. In particular, we consider a one-element spring-damper system. As shown in Figure 1, as the order of time-dependency in $\Theta(t)$ increases, the validation error computed as in (9) grows due to the inability of the sliding window approach to capture the increasing degree of nonlinearity in time.

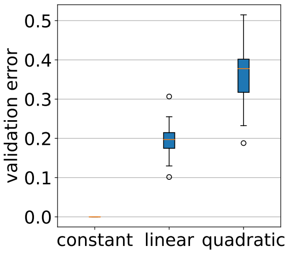

Figure 1: the effect of increasing nonlinearity of $\Theta(t)$

4. Trigonometric Time Dependent IOC

In the following section we present a strategy to address the limitations of a recursive parameter estimation discussed in Section 3, by explicitly introducing a model for time dependency. The idea is to construct a model using a linear combination of trigonometric basis functions and to optimize their hyperparameters within an IOC framework. By learning a model of the cost function's relation with time, we can exploit its predictive capabilities for evaluation on unseen time instances. We refer to this approach as trigonometric time dependent IOC approach (TTD-IOC).

4.1. Trigonometric Basis Functions as Time Features

The model we consider for each time dependent cost function parameter is

$$\begin{aligned}\theta_m(t) =& \alpha_{1,m} 1 + \alpha_{2,m} \cos(\omega_1 t) + \alpha_{3,m} \sin(\omega_1 t) + \\ & \ldots + \alpha_{2E,m} \cos(\omega_E t) + \alpha_{2E+1,m} \sin(\omega_E t),\end{aligned} \tag{10}$$

consisting of $2E$ basis functions, defined by the set of frequencies $\mathcal{W} = \{\omega_1, \ldots, \omega_E\}$. Potential offsets are modelled via an additional constant bias. We define the basis functions vector $\Omega(\mathcal{W}, t) \in \mathbb{R}^{1\times(2E+1)}$ as a row vector consisting of $2E+1$ elements

$$\Omega(\mathcal{W}, t) = [1, \cos(\omega_1 t), \sin(\omega_1 t), \ldots, \cos(\omega_E t), \sin(\omega_E t)], \tag{11}$$

together with the matrix $A \in \mathbb{R}^{(2E+1)\times q}$ consisting of parameters that linearly combine the basis functions

$$A_{(2E+1)\times q} = \begin{bmatrix} \alpha_{11} & \ldots & \alpha_{1q} \\ \vdots & \ddots & \vdots \\ \alpha_{(2E+1)1} & \ldots & \alpha_{(2E+1)q} \end{bmatrix}, \tag{12}$$

resulting in the following model for the time dependent vector $\Theta(t)$ as

$$\Theta(t) = \Omega(\mathcal{W}, t)A. \tag{13}$$

Remark 2 *The model in* (10) *assumes that all cost parameter time dependencies can be approximated with the same set of frequencies $\mathcal{W}$. Considering individual frequencies for each time dependent cost parameter is possible, however, it increases the number of unknown parameters and, accordingly, the amount of required data.*

4.2. Regularized Optimization Problem for TTD-IOC

In the following, we present the proposed algorithm for optimizing the hyperparameters $\mathcal{W}$ and A and introduce the regularization term for obtaining a sparse solution. For this purpose we include the model (13) into the optimization problem presented in equation (7) and consider that measurements are only available for time instances $t = kT_s$, where T_s indicates the sampling time. Additionally, the objective is extended with a lasso inspired regularizer $\beta \in \mathbb{R}_+$. Defining $\mathcal{L}_{U,d} = \mathcal{L}(U, \lambda_{d,i}, \upsilon_d, \Omega(\mathcal{W}, kT_s)A, x^*_d(k), x^*_d(k+N))$, the regularized, time dependent Lagrangian optimization problem results in

$$\begin{aligned} \hat{\mathcal{W}}, \hat{A} = \arg \min_{\mathcal{W}, A, \lambda_{d,i}, \upsilon_d} & \sum_{d=1}^{D_t} \|(\nabla_U \mathcal{L}_{U,d})|_{U=\mathcal{U}^*_d}\|^2 + \beta |A(2:2E+1, 2:q)| \\ & \lambda_{d,i,p} \cdot g_p(F_i(\mathcal{U}^*_d, x^*_d(k)), u^*_d(i)) = 0, \;\; p = 1, \ldots, P, \; d = 1, \ldots, D_t \\ & \lambda_{d,i,p} \geq 0, \;\; p = 1, \ldots, P, \; d = 1, \ldots, D_t \\ & A(:, 0) = v^*_\alpha. \end{aligned} \tag{14}$$

Algorithm 1 TTD-IOC

```
Define $\beta_i, \beta_f, \beta_s, D_t, D_v, \bar{e}_\beta$
$\beta = \beta_i$
while $\beta \le \beta_f$ do
    $\hat{\mathcal{W}}_\beta, \hat{A}_\beta \leftarrow solve(14)$ with $\beta$
    $\hat{\Theta}_\beta(t) \leftarrow \Omega(\hat{\mathcal{W}}_\beta, t)\hat{A}_\beta$
    $\hat{\mathcal{S}}_v \leftarrow solve(6)\ w.r.t.\ \hat{\Theta}_\beta(t)$
    $e_\beta \leftarrow solve(9)$
    if $e_\beta \le \bar{e}_\beta$ then
        $\hat{\mathcal{W}}, \hat{A} = \hat{\mathcal{W}}_\beta, \hat{A}_\beta$
        $\bar{e}_\beta = e_\beta$
    end
    $\beta = \beta + \beta_s$
end
$\hat{\Theta}(t) \leftarrow \Omega(\hat{\mathcal{W}}, t)\hat{A}$
```

The lasso-inspired regularizer thereby penalizes the sum of the absolute values of the sub-matrix of A to minimize the amount of time-varying cost parameters, offering an automatic selection of the model complexity. Furthermore, the trivial solution of setting all elements of A equal to zero is excluded from the set of potential solutions by adding the equality constraint $A(:,1) = v_\alpha^*$, where $v_\alpha^* \in \mathbb{R}^{2E+1}$ is user-defined. Finally, the estimated feature dependent cost parameter are obtained as $\hat{\Theta}(t) = \Omega(\hat{\mathcal{W}}, t)\hat{A}$. Note that the mentioned predictive capabilities with respect to unseen time instances allow for considering different sampling time instances for further predictions.

Remark 3 *Depending on the cost features $\phi(x_i, u_i)$, further constraints can be added to the proposed optimization problem in (14) to preserve the convexity of the estimated cost function $\ell(\cdot,\cdot,\cdot)$ for each fixed time instance t, e.g. non-negativity constraints on $\Theta(t)$ for convex $\phi(x_i, u_i)$.*

The proposed algorithm, presented in Algorithm 1, consists of a line search over β, between β_i and β_f, with a step-wise increase of β_s. For all values of β the optimization problem in (14) is solved, and the quality of the respective parameter estimate is evaluated with respect to the resulting validation error introduced in (9). The initial value of $\bar{e}_\beta$ is chosen sufficiently high, making sure that it is adjusted in the algorithm's first iteration.

5. Results

In this section, we analyse the proposed algorithm by applying it to two illustrative simulation scenarios: the first is a linear dynamical system, i.e. a three-layer spring-damper system (sys1) , while the second system consists of two inverted pendulums (sys2) connected via a spring-damper element, as an example of nonlinear dynamics (see Figure 2). For both systems, we choose the cost function parameter associated with one input to vary with time and design $\phi(x, u)$ to include squares of all states and inputs. The cost parameters of the considered forward problems from which we obtain training and validation demonstrations are presented in Table 1. For the time dependent cost parameter $\theta_{m,i}$ we define three different continuous functions

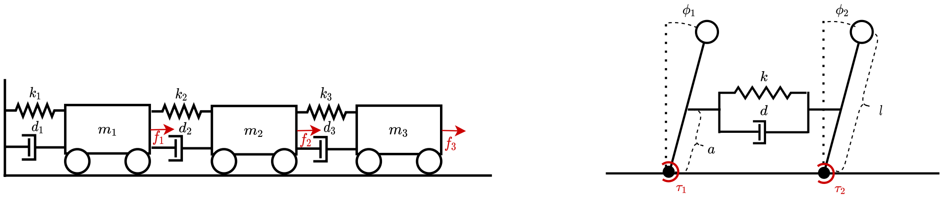

Figure 2: Illustration of considered dynamical systems. The multi-layer spring-damper system is depicted on the left and the double pendulum on the right.

$$\theta_{m,1}(t) = 4 + 1.5\cos(2t) + 1.5\cos(3t)$$
$$\theta_{m,2}(t) = 1.5 + 0.02t^2 - 0.01t$$
$$\theta_{m,3}(t) = 4 + e^{0.2t}$$

Results are obtained using a HP ProBook 440 with an Intel Core i7 processor, while using the Ipopt optimization framework Wächter and Biegler (2006) within the Casadi framework Andersson et al. (2019). The issue of dealing with a nonconvex optimization problem is thereby addressed via a grid search over suitable frequency values (ω_{init}) for the initialization of the applied solver. Its step size and the grid corners can be adjusted by altering the values of ω_i, ω_f, and ω_s. For the subsequent experiments with sys1 they are chosen as $0.5, 2.5$, and 2.0 and as $0.11, 0.11$, and 0.0 for sys2, respectively. The line search parameters are set to $\beta_i = 0.04, \beta_f = 0.06$, and $\beta_s = 0.01$ for sys1 and to $\beta_i = 0.058, \beta_f = 0.061$, and $\beta_s = 0.003$ for sys2. The performance of our proposed approach is evaluated on each system with respect to the validation error defined in (9).

	θ_{x_1/ϕ_1}	$\theta_{\dot{x}_1/\dot{\phi}_1}$	θ_{x_2/ϕ_2}	$\theta_{\dot{x}_2/\dot{\phi}_2}$	θ_{x_3}	$\theta_{\dot{x}_3}$	θ_{f_1/τ_1}	θ_{f_2/τ_2}	θ_{f_3}
sys1	7	5	6	8	6.5	5.5	2	4	$\theta_{m,i}$
sys2	7	5	10	5	-	-	4	$\theta_{m,i}$	-

Table 1: Cost parameters of considered forward problems.

5.1. Multi-layer Spring-Damper System

The considered multi-layer spring-damper system consists of three stacked mass elements, m_1, m_2, and m_3, connected to each other or the wall by a spring-damper pair with spring constants k_1 up to k_3 and damping constants d_1 up to d_3. The input is given by a force vector $F = [f_1, f_2, f_3]$ consisting of three forces that can be exerted on their respective mass. An illustration of the system is given in the left part of Fig. 2 and the dynamics by equation (16).

$$\begin{aligned}
\ddot{x}_1 &= {m_1}^{-1}(-(k_1 + k_2)x_1 - (d_1 + d_2)\dot{x}_1 + k_2 x_2 + d_2 \dot{x}_2 + f_1) \\
\ddot{x}_2 &= {m_2}^{-1}(k_1 x_1 + d_1 \dot{x}_1 - (k_2 + k_3)x_2 - (d_2 + d_3)\dot{x}_2 + k_3 x_3 + d_3 \dot{x}_3 + f_2) \\
\ddot{x}_3 &= {m_3}^{-1}(k_3 x_2 + d_3 \dot{x}_2 - k_3 x_3 - d_3 \dot{x}_3 + f_3)
\end{aligned} \tag{16}$$

The proposed modelling of time dependency within the presented approach enables an improvement with respect to the vanilla spIOC described in Section 2. We measure the improvement in terms of validation error, and confirm the ability of the learned cost to generalize for unseen scenarios, i.e. different initial conditions, as can be observed in the left plots of Figure 3.

5.2. Inverted Double Pendulum

The inverted double pendulum consists of two inverted pendulums of length l, having a mass element m attached at its top. At the height indicated with a they are connected via a spring-damper pair whose constants are indicated by k and d. Each pendulum can be actuated by its individual torque τ_1 or τ_2. An illustration of the system is given in the right picture of Fig. 2 and the dynamics by equation (17). Defining $F = ka(\sin(\phi_2) - \sin(\phi_1)) + da(\cos(\phi_2)\dot{\phi}_2 - \cos(\phi_1)\dot{\phi}_1)$ it follows

$$\begin{aligned}
\ddot{\phi}_1 &= gl^{-1}\sin(\phi_1) + a\cos(\phi_1)F + (ml^2)^{-1}\tau_1 \\
\ddot{\phi}_2 &= gl^{-1}\sin(\phi_2) - a\cos(\phi_2)F + (ml^2)^{-1}\tau_2.
\end{aligned} \tag{17}$$

The resulting validation errors for our considered nonlinear system are given in the right plots of Figure 3. Similarly to the previous example, the proposed approach shows again improved validation errors with respect to spIOC.

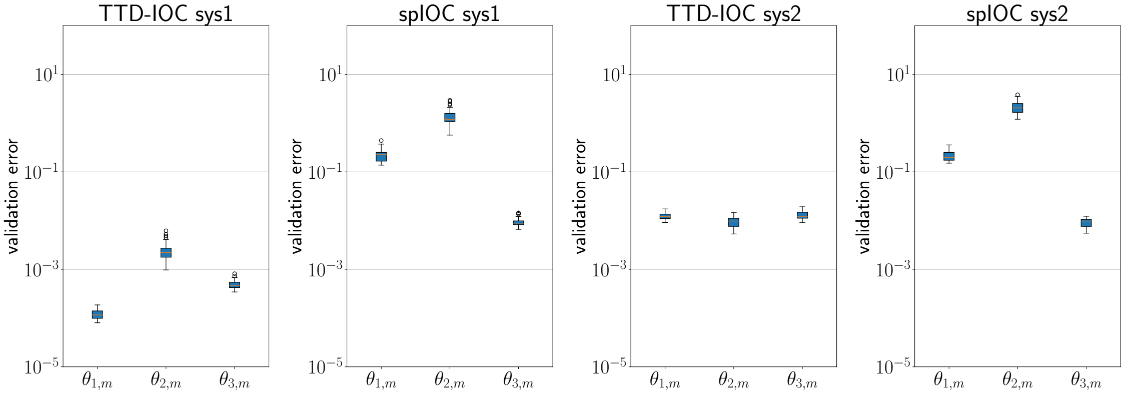

Figure 3: Comparison between the validation errors produced by the trigonometric time dependent IOC (TTD-IOC) and the shortest path IOC (spIOC) for time dependent objectives for both sys1 and sys2. Results are shown on a log scale.

5.3. Comparative Study

In this section, we investigate the effects of varying sampling times and horizon lengths on the validation error. For this purpose, the parameter estimates of sys1, obtained in accordance with training data sequences $\mathcal{S}_t^*$ with a horizon length of $N = 60$ and sampling time $T_s = 0.1$, are used to obtain optimal sequences for an adjusted horizon length or an adjusted sampling time. They are evaluated with respect to new and unseen validation data of given N and T_s. The validation errors resulting from a variation in sampling time are depicted in Figure 5, and those for different horizon lengths in Figure 6. Furthermore, we tested the performance of the approach in the presence of model mismatch, specifically when the number of chosen basis functions E is different from the true model. Assuming that the true cost function is described by 2 trigonometric features, the value E is varied in the set $\{1, 2, 3, 4\}$, and the obtained validation errors are displayed in Figure 4.

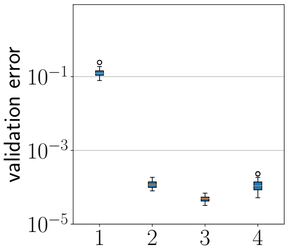

Figure 4: The effect of varying E.

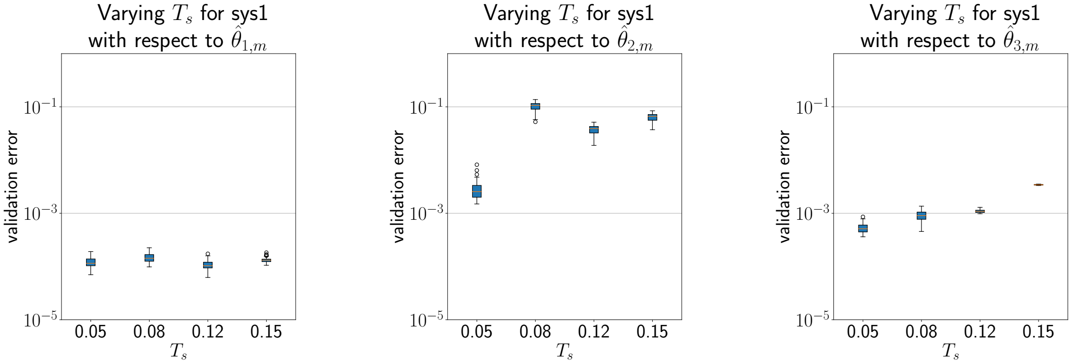

Figure 5: Validation error of using $\hat{\theta}_{1,m}$ up to $\hat{\theta}_{3,m}$ for optimal sequences of sys1 that vary in T_s.

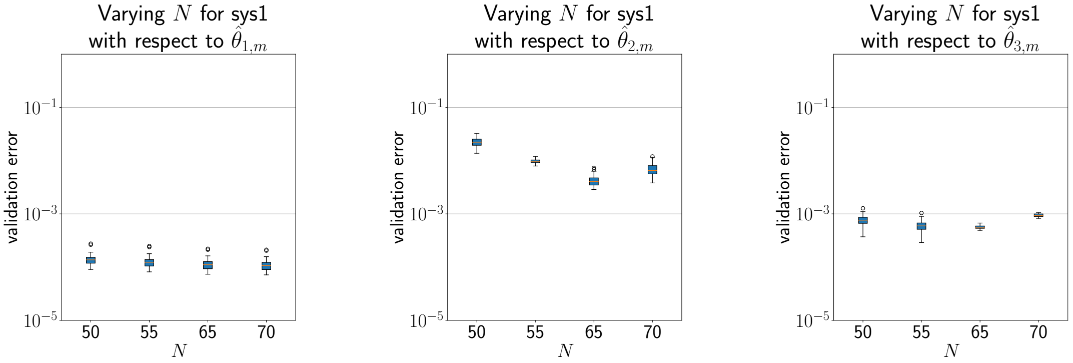

Figure 6: Validation error of using $\hat{\theta}_{1,m}$ up to $\hat{\theta}_{3,m}$ for optimal sequences of sys1 that vary in N.

5.4. Discussion

The results obtained in Section 5.1 and 5.2 show an increased ability to mimic the optimal input and output sequences of underlying continuously time dependent objective functions in comparison to an existing IOC approach that does not model time dependent features (spIOC). Furthermore, in Figure 5 and Figure 6, it can be observed that modeling the time dependency as an explicit feature allows for the consideration of different sampling times and horizon lengths. In general, a better estimate in terms of validation error results in better generalization capabilities. The investigation of varying the number of frequencies in the considered trigonometric feature vector indicates that, given the regularization parameter β, a larger value of E does not lead to a degradation in terms of the validation error. Furthermore, by inspecting the elements of A it can be seen that the parameters associated with unused basis functions are close to zero. While the validation error increases when choosing fewer frequencies, the proposed approach still outperforms the spIOC estimate.

6. Conclusion

This paper has presented a procedure for modeling time dependency as an explicit cost function feature using linear combinations of trigonometric basis functions. Building on previous results, we extended the shortest path IOC approach to include the additional optimization over the time feature hyperparameters, and discussed its performance by analysing two simulation examples. Results show lower validation error compared with the shortest path IOC approach. Furthermore, the use of a model provides an estimate for not only seen but also unseen instances in time, and the proposed method thereby also provides high-quality predictions (i.e. low validation errors) when varying the sampling time, as well as the horizon length in the forward optimization problem. In the future, we plan to examine further the effect of the amount of training data on the validation error and to compare the proposed solution strategy for addressing the nonconvexity of the Lagrangian optimization problem against, e.g., sampling-based routines, as well as test the proposed procedure in real-world problems.

References

Nematollah Ab Azar, Aref Shahmansoorian, and Mohsen Davoudi. From inverse optimal control to inverse reinforcement learning: A historical review. *Annual Reviews in Control*, 2020.

Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2019.

Elena Arcari, Anna Scampicchio, Andrea Carron, and Melanie N Zeilinger. Bayesian multi-task learning using finite-dimensional models: A comparative study. In *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 2021.

Richard Bellman. Dynamic programming. *Science*, 1966.

Peter Englert, Ngo Anh Vien, and Marc Toussaint. Inverse kkt: Learning cost functions of manipulation tasks from demonstrations. *The International Journal of Robotics Research*, 2017.

Wanxin Jin, Dana Kulić, Jonathan Feng-Shun Lin, Shaoshuai Mou, and Sandra Hirche. Inverse optimal control for multiphase cost functions. *IEEE Transactions on Robotics*, 2019.

Mohammad Javad Khodaei, Nicholas Candelino, Amin Mehrvarz, and Nader Jalili. Physiological closed-loop control (pclc) systems: Review of a modern frontier in automation. *Ieee Access*, 2020.

Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. Learning driving styles for autonomous vehicles from demonstration. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

Harold W Kuhn and Albert W Tucker. Nonlinear programming. In *Traces and emergence of nonlinear programming*. Springer, 2014.

Miguel Lázaro-Gredilla, Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Aníbal R Figueiras-Vidal. Sparse spectrum gaussian process regression. *The Journal of Machine Learning Research*, 2010.

Weiwei Li, Emanuel Todorov, and Dan Liu. Inverse optimality design for biological movement systems. *IFAC Proceedings Volumes*, 2011.

Jonathan Feng-Shun Lin, Vincent Bonnet, Adina M Panchea, Nacim Ramdani, Gentiane Venture, and Dana 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.

Jonathan Feng-Shun Lin, Pamela Carreno-Medrano, Mahsa Parsapour, Maram Sakr, and Dana Kulić. Objective learning from human demonstrations. *Annual Reviews in Control*, 2021.

N. Trésor Mbungu, Raj M. Naidoo, and R.C. Bansal. Real-time electricity pricing: Tou-mpc based energy management for commercial buildings. *Energy Procedia*, 2017.

Marcel Menner and Melanie N Zeilinger. Convex formulations and algebraic solutions for linear quadratic inverse optimal control problems. In *2018 European control conference (ECC)*. IEEE, 2018.

Marcel Menner and Melanie N Zeilinger. Maximum likelihood methods for inverse learning of optimal controllers. *IFAC-PapersOnLine*, 2020.

Marcel Menner, Peter Worsnop, and Melanie N Zeilinger. Constrained inverse optimal control with application to a human manipulation task. *IEEE Transactions on Control Systems Technology*, 2019.

Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, 2000.

M Cody Priess, Richard Conway, Jongeun Choi, John M Popovich, and Clark Radcliffe. Solutions to the inverse lqr problem with application to biological systems analysis. *IEEE Transactions on control systems technology*, 2014.

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *IJCAI*, 2007.

Yue Ruan, Hood Thabit, Lalantha Leelarathna, Sara Hartnell, Malgorzata E Willinska, Sibylle Dellweg, Carsten Benesch, Julia K Mader, Manuel Holzer, Harald Kojzar, et al. Variability of insulin requirements over 12 weeks of closed-loop insulin delivery in adults with type 1 diabetes. *Diabetes Care*, 2016.

Gary Scheiner and Bret A. Boyer. Characteristics of basal insulin requirements by age and gender in type-1 diabetes patients using insulin pump therapy. *Diabetes Research and Clinical Practice*, 2005.

Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 2006.

Kevin Westermann, Jonathan Feng-Shun Lin, and Dana Kulić. Inverse optimal control with time-varying objectives: application to human jumping movement analysis. *Scientific reports*, 2020.