Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Maximum Likelihood Methods for Inverse Learning of Optimal Controllers ⋆

Marcel Menner * **and Melanie N. Zeilinger** *

* *Institute for Dynamic Systems and Control, ETH Zurich, Zurich, Switzerland (e-mail: {mmenner,mzeilinger}@ethz.ch)*

Abstract: This paper presents a framework for inverse learning of objective functions for constrained optimal control problems, which is based on the Karush-Kuhn-Tucker (KKT) conditions. We discuss three variants corresponding to different model assumptions and computational complexities. The first method uses a convex relaxation of the KKT conditions and serves as the benchmark. The main contribution of this paper is the proposition of two learning methods that combine the KKT conditions with maximum likelihood estimation. The key benefit of this combination is the systematic treatment of constraints for learning from noisy data with a branch-and-bound algorithm using likelihood arguments. This paper discusses theoretic properties of the learning methods and presents simulation results that highlight the advantages of using the maximum likelihood formulation for learning objective functions.

Keywords: Learning for control, data-based control, constrained control.

1. INTRODUCTION

Objective functions used for control design do not necessarily correspond to the actual performance specifications for a dynamical system, which may comprise complex or sparse targets. Instead, they are often chosen to facilitate gradient-based numerical optimization, which, in turn, makes their design not very intuitive and their calibration can require a tedious manual engineering effort to meet the performance specifications. Inverse learning concepts such as inverse optimal control offer an attractive design paradigm for learning objective functions from data to avoid their manual tuning. In this context, the data can originate, e.g., from a human actor, who demonstrates how to optimally operate the dynamical system being considered. Learning from demonstrations, however, necessarily implies that the data are subject to noise and other sources of sub-optimalities, which have to be taken into account.

In this paper, we present and contrast three variants of an inverse optimal control approach that leverages the Karush-Kuhn-Tucker optimality conditions, cf. Karush (1939); Kuhn and Tucker (1951), to learn objective functions of optimal controllers, e.g., for linear quadratic or model predictive control, from noisy data. The first method is based on a convex relaxation of the KKT conditions to allow for noisy data, which is similar to the formulation in Englert et al. (2017); Menner et al. (2019b) and included in this paper as the benchmark. The main contribution of this paper is the proposition of two inverse optimal control methods that combine the KKT conditions with a maximum likelihood estimation algorithm, which offer the key benefit of systematically dealing with state and input constraints in the presence of noisy data. The underlying assumption is that the data are samples from a distribution (rather than expecting deterministic, optimal data). Maximum likelihood estimation is enabled by an algorithm that uses branch-and-bound-type ideas based on the likelihoods of active constraints. The second contribution is a theoretical and simulative analysis of the properties of the three methods. In theory, we analyze the learning results of the inverse optimal control methods for unconstrained, linear dynamical systems and a quadratic cost function. In simulation, we present learning results for both constrained, linear and nonlinear systems.

Related work Inverse optimal control methods typically model data as deterministic and resulting from an optimal control problem [Hewing et al. (2020)], whereas we explicitly consider the data as stochastic. In Kalman (1964), Menner and Zeilinger (2018), and Mombaur et al. (2010), inverse optimal control methods for linear, unconstrained systems are presented. Englert et al. (2017) and Menner et al. (2019b) use a formulation similar to the first method presented in this paper, which is based on the relaxation of the KKT conditions. Chou et al. (2018); Chou et al. (2020) address a related problem by learning constraints. The method in Menner et al. (2019a) considers a non-deterministic model, but does not consider constraints in the learning procedure. The closest to the proposed likelihood estimation methods is Esfahani et al. (2018), where the main difference lies in the proposed formulation using likelihood arguments offering the key advantage of dealing with constraints using a branch-and-bound algorithm.

Inverse reinforcement learning methods typically model data by means of a Markov decision process, cf. Ziebart et al. (2008); Levine and Koltun (2012); Finn et al. (2016). As a result, these methods can deal with noise by construction, but constraints are typically not considered. Compared to inverse reinforcement learning methods, we base our algorithm on the KKT conditions in order to explicitly consider constraints and noisy data.

⋆ This work was supported by the Swiss National Science Foundation under grant no. PP00P2_157601 / 1.

2. PROBLEM STATEMENT

We consider discrete-time dynamical systems of the form

$$\boldsymbol{x}(k+1) = \boldsymbol{f}(\boldsymbol{x}(k), \boldsymbol{u}(k)), \tag{1}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^n$ is the state at time k, $\boldsymbol{u}(k) \in \mathbb{R}^m$ is the input, and $\boldsymbol{f}$ is, in general, a nonlinear function.

Control Model We consider optimal controllers of the form

$$\begin{aligned} \boldsymbol{v}_k^\star = \arg\min_{\boldsymbol{v}_k, \boldsymbol{z}_k \ \forall k} \ & \boldsymbol{\theta}^T \boldsymbol{\phi}\left(\boldsymbol{v}_0, ...\boldsymbol{v}_N, \boldsymbol{z}_0, ...\boldsymbol{z}_{N+1}\right) \\ \text{s.t. } & \boldsymbol{z}_{k+1} = \boldsymbol{f}(\boldsymbol{z}_k, \boldsymbol{v}_k) \quad \forall k = 0, ...N \\ & \boldsymbol{g}(\boldsymbol{z}_k, \boldsymbol{v}_k) \leq \boldsymbol{0} \qquad \forall k = 0, ...N \\ & \boldsymbol{z}_0 = \boldsymbol{z}(0), \end{aligned} \tag{2}$$

where $\boldsymbol{v}_k^\star$ are optimal inputs at time k, $\boldsymbol{z}_k$ are the predicted states given inputs $\boldsymbol{v}_k$, and the initial condition is $\boldsymbol{z}(0)$. The minimizers of (2), i.e., the nominal states $\boldsymbol{z}_k^\star$ and inputs $\boldsymbol{v}_k^\star$, express a motion plan and do not necessarily coincide with the measured states $\boldsymbol{x}(k)$ and inputs $\boldsymbol{u}(k)$. The objective function is defined by $\boldsymbol{\phi}$, which is weighted by the parameters $\boldsymbol{\theta}$. The function $\boldsymbol{g}$ defines constraints and N is the prediction horizon. We assume that $\boldsymbol{\phi}$, $\boldsymbol{f}$, and $\boldsymbol{g}$ are known and continuously differentiable.

Assumption on the data In expectation, the data are assumed to be the solution to (2), i.e., the demonstration, e.g. from a human agent, is modeled as an optimal controller. Due to noise and other sources of sub-optimalities, we assume a probability distribution for the data:

i) We model the initial condition, denoted $\boldsymbol{x}(0)$, as uncertain and assume
$$\boldsymbol{x}(0) \sim \mathcal{N}(\boldsymbol{z}(0), \boldsymbol{\Sigma}_0), \tag{3a}$$
i.e., $\boldsymbol{x}(0)$ is Gaussian distributed with mean $\boldsymbol{z}(0)$ and covariance $\boldsymbol{\Sigma}_0$.

ii) We model the observed inputs, denoted $\boldsymbol{u}(k)$, as suboptimal and assume
$$\boldsymbol{u}(k) \sim \mathcal{N}(\boldsymbol{v}_k^\star, \boldsymbol{\Sigma}_k^u) \quad \forall k = 0, ..., N, \tag{3b}$$
where $\boldsymbol{v}_k^\star$ are the minimizers of (2).

In the context of learning from data generated by a human agent, eq. (3a) and eq. (3b) model that a human agent may be uncertain about the true initial state $\boldsymbol{x}(0)$ and may not execute the intended motion plan optimally.

Objective In this paper, we learn the parameters $\boldsymbol{\theta}$ of the optimal controller in (2) from data represented in the form of state $\boldsymbol{x}(k)$ and input $\boldsymbol{u}(k)$ measurements satisfying (1) generated, e.g., by a human actor modeled as in (3).

Notation & Preliminaries

In order to ease exposition, we vectorize sequences

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}(0) \\ \boldsymbol{u}(1) \\ \vdots \\ \boldsymbol{u}(N) \end{bmatrix}, \ \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_0 \\ \boldsymbol{z}_1 \\ \vdots \\ \boldsymbol{z}_{N+1} \end{bmatrix}, \ \boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_0 \\ \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_N \end{bmatrix}$$

and use the function $\boldsymbol{F}$ relating the vectorized sequences as in (1): $\boldsymbol{Z} = \boldsymbol{F}(\boldsymbol{V}, \boldsymbol{z}_0)$. We use $\boldsymbol{\phi}\left(\boldsymbol{v}_0, ...\boldsymbol{v}_N, \boldsymbol{z}_0, ...\boldsymbol{z}_{N+1}\right) = \boldsymbol{\phi}\left(\boldsymbol{V}, \boldsymbol{Z}\right)$, as well as $\boldsymbol{g}\left(\boldsymbol{v}_0, ...\boldsymbol{v}_N, \boldsymbol{z}_0, ...\boldsymbol{z}_{N+1}\right) = \boldsymbol{g}\left(\boldsymbol{V}, \boldsymbol{Z}\right)$ equivalently. Further, $\boldsymbol{U} \sim \mathcal{N}(\boldsymbol{V}, \boldsymbol{\Sigma})$ implies $\boldsymbol{u}(k) \sim \mathcal{N}(\boldsymbol{v}_k^\star, \boldsymbol{\Sigma}_k^u)$ for all $k = 0, ..., N$, i.e., $\boldsymbol{\Sigma} \in \mathbb{R}^{mN \times mN}$ is block-diagonal with blocks $\boldsymbol{\Sigma}_k^u$ and we define $\|\boldsymbol{x}\|_{\boldsymbol{X}} = \boldsymbol{x}^T \boldsymbol{X} \boldsymbol{x}$.

Let idx, $\neg$idx $\in \{0,1\}^s$ with idx+$\neg$idx $= \{1\}^s$. For a vector $\boldsymbol{\lambda} \in \mathbb{R}^s$, we define $\boldsymbol{\lambda}_{\text{idx}}$ selecting all elements λ_i for which $\text{idx}_i = 1$ ($\boldsymbol{\lambda}_{\neg\text{idx}}$ selecting λ_i for which $\text{idx}_i = 0$). $\boldsymbol{\delta}^i$ is a unit vector with $\delta_j^i = 1$ if $i = j$ and $\delta_j^i = 0$ if $i \neq j$.

Consider the optimization problem

$$\begin{aligned} \boldsymbol{V}^\star = \arg\min_{\boldsymbol{V}} \ & \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{V}, \boldsymbol{x}) \\ \text{s.t. } & \boldsymbol{g}(\boldsymbol{V}, \boldsymbol{x}) \leq \boldsymbol{0}. \end{aligned} \tag{4}$$

The KKT conditions of (4) are given by

$$\text{KKT}_{\boldsymbol{\theta}}(\boldsymbol{V}^\star, \boldsymbol{x}) = \begin{cases} \nabla_{\boldsymbol{V}} \mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{V}, \boldsymbol{x})|_{\boldsymbol{V}=\boldsymbol{V}^\star} = \boldsymbol{0} \\ \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V}, \boldsymbol{x})) = 0 \\ \boldsymbol{g}(\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V}, \boldsymbol{x})) \leq \boldsymbol{0} \\ \boldsymbol{\lambda} \geq \boldsymbol{0} \end{cases} \tag{5}$$

with the dual variables $\boldsymbol{\lambda}$ and the Lagrangian

$$\mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{V}, \boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V}, \boldsymbol{x})) + \boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V}, \boldsymbol{x})).$$

The KKT conditions are necessary for constrained optimization (first-order derivative tests), i.e., any $\boldsymbol{V}^\star$ locally minimizing (4) satisfies (5). For more details, the reader is referred, e.g., to Boyd and Vandenberghe (2004).

Proposition 1. Consider

$$\begin{aligned} f^1 = \max_{\boldsymbol{x} \in \mathbb{R}^n} \ & f(\boldsymbol{x}) \\ \text{s.t. } & g_1(\boldsymbol{x}) \leq 0 \end{aligned} \qquad \begin{aligned} f^2 = \max_{\boldsymbol{x} \in \mathbb{R}^n} \ & f(\boldsymbol{x}) \\ \text{s.t. } & g_1(\boldsymbol{x}) \leq 0, g_2(\boldsymbol{x}) \leq 0 \end{aligned}$$

and let $\{\boldsymbol{x} | g_1(\boldsymbol{x}) \leq 0\}$ be a non-empty set and $f(\boldsymbol{x})$ be bounded. Then, $f^1 \geq f^2$.

3. INVERSE LEARNING METHODS

This section presents three methods for inverse learning of the objective function, which utilize the KKT conditions in (5) as follows: Suppose $\boldsymbol{V}^\star$ is the result of (2) for some true parameters $\boldsymbol{\theta} = \boldsymbol{\theta}_{\text{t}}$ and initial condition $\boldsymbol{z}(0)$. Then, $\text{KKT}_{\boldsymbol{\theta}}(\boldsymbol{V}^\star, \boldsymbol{z}(0))$ hold for $\boldsymbol{\theta} = \boldsymbol{\theta}_{\text{t}}$. Hence, the KKT conditions can be used to learn $\boldsymbol{\theta}_{\text{t}}$ given $\boldsymbol{V}^\star$. Method 1 is based on a relaxation of the KKT conditions in (5) to allow for noisy data. Methods 2 and 3 are based on maximum likelihood estimation and use the distribution in (3). The three methods vary in computational complexity and model assumptions in the form of approximations.

Method 1 This method uses a relaxation of the KKT conditions to directly relate the data $\boldsymbol{U}, \boldsymbol{x}(0)$ with $\boldsymbol{\theta}$:

$$\begin{aligned} \hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}, \boldsymbol{\lambda}} \ & \|\nabla_{\boldsymbol{V}} \mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{V}, \boldsymbol{x}(0))|_{\boldsymbol{V}=\boldsymbol{U}}\|_I \\ \text{s.t. } & \boldsymbol{\lambda} \geq \boldsymbol{0} \\ & \boldsymbol{\lambda}_{\neg\text{idx}} = \boldsymbol{0}, \end{aligned} \tag{6a}$$

where $\neg$idx indexes inactive constraints, $g_i(\boldsymbol{U}, \boldsymbol{x}(0))$, with

$$\neg\text{idx}_i = \begin{cases} 1 & \text{if } g_i(\boldsymbol{U}, \boldsymbol{x}(0)) < 0 \\ 0 & \text{else.} \end{cases}$$

The main advantage is that (6a) is a convex optimization problem. Compared to (5), $\nabla_{\boldsymbol{V}} \mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{V}, \boldsymbol{x}(0))|_{\boldsymbol{V}=\boldsymbol{U}} \neq \boldsymbol{0}$ as well as $\boldsymbol{g}(\boldsymbol{U}, \boldsymbol{F}(\boldsymbol{U}, \boldsymbol{x})) \neq \boldsymbol{0}$ due to noisy data. Therefore, we minimize $\nabla_{\boldsymbol{V}} \mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{V}, \boldsymbol{x}(0))|_{\boldsymbol{V}=\boldsymbol{U}}$, where $\boldsymbol{\lambda}^T \boldsymbol{g}(\boldsymbol{V}, \boldsymbol{F}(\boldsymbol{V}, \boldsymbol{x})) = 0$ with $\boldsymbol{\lambda} \geq \boldsymbol{0}$ is relaxed to $\boldsymbol{\lambda} \geq \boldsymbol{0}$ and $\boldsymbol{\lambda}_{\neg\text{idx}} = \boldsymbol{0}$.

Method 2 This method is based on maximum likelihood estimation and uses the expected value of the initial condition in (3a) with $\boldsymbol{x}(0) = \boldsymbol{z}(0)$. Using the distribution of

the control inputs in (3b), the parameters $\boldsymbol{\theta}$ are estimated to maximize the probability of observing $\boldsymbol{U}$:

$$\begin{aligned}\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \; & p(\boldsymbol{V}|\boldsymbol{U},\boldsymbol{\Sigma}) \\ \text{s.t. } \boldsymbol{V} = \arg\min_{\tilde{\boldsymbol{V}}} \; & \boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\boldsymbol{V}},\boldsymbol{F}(\tilde{\boldsymbol{V}},\boldsymbol{x}(0))) \\ \text{s.t. } & \boldsymbol{g}(\tilde{\boldsymbol{V}},\boldsymbol{F}(\tilde{\boldsymbol{V}},\boldsymbol{x}(0))) \leq \boldsymbol{0}.\end{aligned} \tag{6b}$$

Method 3 This method considers the uncertainty about the initial condition explicitly. The method additionally optimizes over the initial condition with $\boldsymbol{x}(0) \sim \mathcal{N}(\boldsymbol{z}(0), \boldsymbol{\Sigma}_0)$ and the model for the inverse learning problem is given by

$$\begin{aligned}\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta},\boldsymbol{z}(0)} \; & p(\boldsymbol{V}|\boldsymbol{U},\boldsymbol{\Sigma})p(\boldsymbol{z}(0)|\boldsymbol{x}(0),\boldsymbol{\Sigma}_0) \\ \text{s.t. } \boldsymbol{V} = \arg\min_{\tilde{\boldsymbol{V}}} \; & \boldsymbol{\theta}^T\boldsymbol{\phi}(\tilde{\boldsymbol{V}},\boldsymbol{F}(\tilde{\boldsymbol{V}},\boldsymbol{z}(0))) \\ \text{s.t. } & \boldsymbol{g}(\tilde{\boldsymbol{V}},\boldsymbol{F}(\tilde{\boldsymbol{V}},\boldsymbol{z}(0))) \leq \boldsymbol{0}.\end{aligned} \tag{6c}$$

Both maximum likelihood estimation methods (6b) and (6c) yield bi-level optimization problems that are solved as described in Section 4.

4. ALGORITHM FOR MAXIMUM LIKELIHOOD ESTIMATION

In the following, we outline the algorithm for learning $\boldsymbol{\theta}$ using Method 2. The algorithm for Method 3 follows analogously. We first replace the likelihood $p(\boldsymbol{V}|\boldsymbol{U},\boldsymbol{\Sigma})$ by its logarithmic likelihood $\log p(\boldsymbol{V}|\boldsymbol{U},\boldsymbol{\Sigma})$ (log-likelihood) and the lower level optimization problem in (6b) by its KKT conditions in (5). This way, the bi-level optimization problem is replaced by a combinatorial problem due to the complementary slackness condition $\boldsymbol{\lambda}^T\boldsymbol{g}(\boldsymbol{V},\boldsymbol{F}(\boldsymbol{V},\boldsymbol{x})) = 0$:

$$\begin{aligned}p = \max_{\boldsymbol{\theta},\boldsymbol{V},\boldsymbol{\lambda},\text{idx}} \; & \log p\left(\boldsymbol{V}|\boldsymbol{U},\boldsymbol{\Sigma}\right) \\ \text{s.t. } & \text{KKT}_{\boldsymbol{\theta},\text{idx}}(\boldsymbol{V},\boldsymbol{x}(0)) \\ & \text{idx} \in \{0,1\}^s\end{aligned} \tag{7}$$

with idx selecting which of the s constraints are active, i.e.

$$\text{KKT}_{\boldsymbol{\theta},\text{idx}}(\boldsymbol{V},\boldsymbol{x}) = \begin{cases}\nabla_{\tilde{\boldsymbol{V}}}\mathcal{L}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{V}},\boldsymbol{x})|_{\tilde{\boldsymbol{V}}=\boldsymbol{V}} = \boldsymbol{0} \\ \boldsymbol{g}(\boldsymbol{V},\boldsymbol{F}(\boldsymbol{V},\boldsymbol{x})) \leq \boldsymbol{0} \\ \boldsymbol{g}(\boldsymbol{V},\boldsymbol{F}(\boldsymbol{V},\boldsymbol{x}))_{\text{idx}} = 0 \\ \boldsymbol{\lambda} \geq \boldsymbol{0} \\ \boldsymbol{\lambda}_{\neg\text{idx}} = \boldsymbol{0}.\end{cases}$$

Solving the combinatorial optimization problem in (7) directly is computationally intensive. However, using likelihood arguments with branch-and-bound-type ideas, (7) becomes practically feasible as outlined in the following.

Algorithm 1 summarizes the procedure to solve (7), which is based on systematically enumerating candidate solutions and is conceptually similar to active-set methods, cf., Murty and Yu (1988). The algorithm aims at reducing the number of times (7) has to be solved for a fixed combination of active constraints, denoted $\text{idx}^j \in \{0,1\}^s$, where we use j to index the specific combination of active constraints. First (Line 1 in Alg. 1), we solve

$$\begin{aligned}p^0 = \max_{\boldsymbol{\theta},\boldsymbol{V}} \; & \log p\left(\boldsymbol{V}|\boldsymbol{U},\boldsymbol{\Sigma}\right) \\ \text{s.t. } & \text{KKT}_{\boldsymbol{\theta},\text{idx}^0=\{0\}^s}(\boldsymbol{V},\boldsymbol{x}(0)),\end{aligned} \tag{8}$$

where p^0 is the log-likelihood of observing $\boldsymbol{U}$ and no active constraints ($\text{idx}^0 = \{0\}^s$). Next (Line 2), we compute an upper bound on the log-likelihood of constraint i's activeness given the data $\boldsymbol{U}$ as

$$\begin{aligned}\bar{p}^i = \max_{\boldsymbol{V}} \; & \log p(\boldsymbol{V}|\boldsymbol{U},\boldsymbol{\Sigma}) \\ \text{s.t. } & \boldsymbol{g}(\boldsymbol{V},\boldsymbol{F}(\boldsymbol{V},\boldsymbol{x}(0))) \leq \boldsymbol{0} \\ & \boldsymbol{g}(\boldsymbol{V},\boldsymbol{F}(\boldsymbol{V},\boldsymbol{x}(0)))_{\boldsymbol{\delta}^i} = \boldsymbol{0}.\end{aligned} \tag{9}$$

From Proposition 1, constraint i is not likely to be active if $\bar{p}^i \leq p^0$ and consequently, constraint i does not need to be enumerated, which is the first key component of the algorithm's efficiency as the number of possible constraint combinations, denoted c, can be reduced significantly.

Next (Line 3), we compute upper bounds for the log-likelihood p in (7) for the fixed combinations of the active constraints idx^j (excluding the discarded constraints):

$$\begin{aligned}\tilde{p}^j = \max_{\boldsymbol{V}} \; & \log p(\boldsymbol{V}|\boldsymbol{U},\boldsymbol{\Sigma}) \\ \text{s.t. } & \boldsymbol{g}(\boldsymbol{V},\boldsymbol{F}(\boldsymbol{V},\boldsymbol{x}(0))) \leq \boldsymbol{0} \\ & \boldsymbol{g}(\boldsymbol{V},\boldsymbol{F}(\boldsymbol{V},\boldsymbol{x}(0)))_{\text{idx}^j} = \boldsymbol{0}\end{aligned} \tag{10}$$

for all $j = 1, ...c$. As a result, we obtain c possible candidate constraint combinations as well as their upper bounds:

$$\mathcal{D} = \left\{\left\{\text{idx}^1, \tilde{p}^1\right\}, \left\{\text{idx}^2, \tilde{p}^2\right\}, ..., \left\{\text{idx}^c, \tilde{p}^c\right\}\right\}. \tag{11}$$

For ease of exposition, $\mathcal{D}$ is ordered so that $\tilde{p}^j \geq \tilde{p}^{j+1}$. The log-likelihoods $\tilde{p}^j$ are the second key component for the algorithm's efficiency and are used as the stopping criteria, i.e., (7) is solved for idx^j starting with $j = 1$ until $\tilde{p}^j \leq \max\{p^0, ..., p^{j-1}\}$ (Line 5–9).

Algorithm 1 Overall algorithm

1: $\hat{\boldsymbol{\theta}}, p^0 \leftarrow$ Solve (8) with $\text{idx} = \{0\}^s$
2: For each $i = 1, ..., s$, compute $\bar{p}^i$ using (9) and discard constraint i if $\bar{p}^i \leq p^0$
3: Compute $\mathcal{D}$ in (11) using (10)
4: $j = 1, \hat{p} = p^0$
5: **while** $\tilde{p}^j > \max\{p^0, ..., p^{j-1}\} = \hat{p}$ ▷ end if less likely
6: $\quad\boldsymbol{\theta}^j, p^j \leftarrow$ Solve (7) with fixed active constraints idx^j
7: $\quad$**if** $p^j \geq \hat{p}$
8: $\quad\quad\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^i$, $\hat{p} \leftarrow p^i$
9: $\quad j \leftarrow j + 1$

Remark 1. We implemented a projected gradient method that uses backtracking line search [Armijo (1966)] to solve both (6a) for Method 1 and (7) with the fixed active constraints idx^j for Method 2 and Method 3. Section 6 details the computation times of the three inverse learning methods, which show that the proposed algorithm is computationally feasible.

Fig. 1 illustrates the concept of the upper bounds on the constraint likelihoods (Line 2 in Algorithm 1). In the given example, constraint 4 and constraint 5 do not have to be considered for learning, i.e., do not need to be enumerated. The resulting candidate constraint combinations are

$$\mathcal{D} = \left\{\left\{\begin{bmatrix}1\\0\\0\\0\\0\end{bmatrix}, \tilde{p}^1\right\}, \left\{\begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix}, \tilde{p}^2\right\}, \left\{\begin{bmatrix}1\\1\\0\\0\\0\end{bmatrix}, \tilde{p}^3\right\}, \left\{\begin{bmatrix}0\\0\\1\\0\\0\end{bmatrix}, \tilde{p}^4\right\}\right\}$$

Note that $\tilde{p}^1 = \bar{p}^1$, $\tilde{p}^2 = \bar{p}^2$, $\tilde{p}^3 = \bar{p}^2$, and $\tilde{p}^4 = \bar{p}^3$.

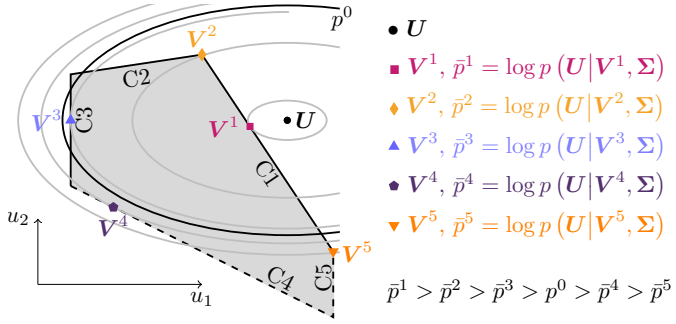

Fig. 1. Illustration of upper bounds on likelihoods $\bar{p}^i$ as computed in (9) for polytopic constraints $i = 1, ...5$. The five upper bounds $\bar{p}^1$–$\bar{p}^5$ (gray ellipses) for the five constraints (C1–C5) as well as p^0 (black ellipse) are displayed as level sets. For Ci, $\boldsymbol{V}^i$ denotes the corresponding, projected input sequence.

5. ANALYSIS FOR LINEAR SYSTEMS AND QUADRATIC COST FUNCTION

In this section, we present properties of the learning methods for a common class of dynamical systems and cost functions. Suppose the system in (1) is unconstrained and linear (time-invariant or time-varying), i.e.,

$$\boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k, \tag{12a}$$

and the cost function is of the form

$$\boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{V}, \boldsymbol{Z}) = \sum_{k=0}^{N} \boldsymbol{z}_k^T\boldsymbol{Q}_{\boldsymbol{\theta}}\boldsymbol{z}_k + \boldsymbol{v}_k^T\boldsymbol{R}_{\boldsymbol{\theta}}\boldsymbol{v}_k \tag{12b}$$

with $\boldsymbol{Q}_{\boldsymbol{\theta}}, \boldsymbol{R}_{\boldsymbol{\theta}} \succ \boldsymbol{0}$. Then, the stationarity condition can be written as

$$\nabla_{\boldsymbol{V}}\mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{V}, \boldsymbol{z}(0)) = \boldsymbol{M}_{\boldsymbol{\theta}}\boldsymbol{V} + \boldsymbol{N}_{\boldsymbol{\theta}}\boldsymbol{z}(0),$$

where both $\boldsymbol{M}_{\boldsymbol{\theta}}$ and $\boldsymbol{N}_{\boldsymbol{\theta}}$ depend on the system dynamics, i.e., $\boldsymbol{A}_k, \boldsymbol{B}_k$, and are linear in their parameters $\boldsymbol{\theta}$, i.e., $\boldsymbol{M}_{\mu\boldsymbol{\theta}_1+\boldsymbol{\theta}_2} = \mu\boldsymbol{M}_{\boldsymbol{\theta}_1} + \boldsymbol{M}_{\boldsymbol{\theta}_2}$ and $\boldsymbol{N}_{\mu\boldsymbol{\theta}_1+\boldsymbol{\theta}_2} = \mu\boldsymbol{N}_{\boldsymbol{\theta}_1} + \boldsymbol{N}_{\boldsymbol{\theta}_2}$ with the scalar μ.

Let $\boldsymbol{\theta}_\mathrm{t}$ be the true parameters and, without loss of generality, $\|\boldsymbol{\theta}_\mathrm{t}\|_2 = 1$ (scale-invariance of the cost function), i.e., $\nabla_{\boldsymbol{V}}\mathcal{L}_{\boldsymbol{\theta}_\mathrm{t}}(\boldsymbol{V}, \boldsymbol{z}(0)) = \boldsymbol{M}_{\boldsymbol{\theta}_\mathrm{t}}\boldsymbol{V} + \boldsymbol{N}_{\boldsymbol{\theta}_\mathrm{t}}\boldsymbol{z}(0) = \boldsymbol{0}$. The goal of the learning methods is thus to estimate $\boldsymbol{\theta}_\mathrm{t}$ or any scaled version $\hat{\boldsymbol{\theta}} = \mu\boldsymbol{\theta}_\mathrm{t}$ with $\mu > 0$ from data. Desirable properties of the learning method are that $\boldsymbol{\theta}_\mathrm{t}$ results in expectation and that all $\hat{\boldsymbol{\theta}} = \mu\boldsymbol{\theta}_\mathrm{t}$ with $\mu > 0$ are equally likely.

Theorem 1 shows that the expected value of Method 3 is the true parameter vector $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_\mathrm{t}$ and that Method 3 is indifferent toward the cost function's scale, i.e., any $\hat{\boldsymbol{\theta}} = \mu\boldsymbol{\theta}_\mathrm{t}$ with $\mu > 0$ are equally likely (in expectation). Method 2 is equally indifferent toward the parameters' scale but the expected parameters are only $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_\mathrm{t}$ if $\boldsymbol{x}(0) = \boldsymbol{z}(0)$ (proof omitted as it can be similarly derived). Theorem 2 shows that the expected parameters $\hat{\boldsymbol{\theta}}$ of Method 1 are not necessarily $\boldsymbol{\theta}_\mathrm{t}$ and that Method 1 is not indifferent toward the parameters' scale.

Theorem 1. Consider unconstrained, linear systems of the form (12a) and cost functions (12b). Let $\boldsymbol{U} \sim \mathcal{N}(\boldsymbol{V}, \boldsymbol{\Sigma})$ and $\boldsymbol{x}(0) \sim \mathcal{N}(\boldsymbol{z}(0), \boldsymbol{\Sigma}_0)$. In expectation, Method 3 returns $\boldsymbol{\theta}_\mathrm{t}$ (result 1) and any other parameter realization is necessarily $\hat{\boldsymbol{\theta}} \propto \boldsymbol{\theta}_\mathrm{t}$ (result 2).

Proof. Without loss of generality, define $\boldsymbol{\theta} = \boldsymbol{\theta}_\mathrm{t} + \mu\partial\boldsymbol{\theta}$ with $\mu \in \mathbb{R}$ and $\partial\boldsymbol{\theta} \in \mathbb{R}^p$ such that $\|\partial\boldsymbol{\theta}\|_2 = 1$. The results will be shown by proving the following statements:

Claim 1: For any $\partial\boldsymbol{\theta}$,

$$\hat{\mu} = 0 = \arg\max_{\mu,\boldsymbol{V}} \mathbb{E}\left[-\|\boldsymbol{U} - \boldsymbol{V}\|_{\boldsymbol{\Sigma}^{-1}} - \|\boldsymbol{x}(0) - \boldsymbol{z}(0)\|_{\boldsymbol{\Sigma}_0^{-1}}\right]$$
$$\text{s.t. } \boldsymbol{0} = \boldsymbol{M}_{\boldsymbol{\theta}_\mathrm{t}+\mu\partial\boldsymbol{\theta}}\boldsymbol{V} + \boldsymbol{N}_{\boldsymbol{\theta}_\mathrm{t}+\mu\partial\boldsymbol{\theta}}\boldsymbol{z}(0)$$

Claim 2: For $\partial\boldsymbol{\theta} = \boldsymbol{\theta}_\mathrm{t}$, any $\mu \in \mathbb{R}$ minimizes

$$\hat{\mu} = \arg\max_{\mu,\boldsymbol{V}} \mathbb{E}\left[-\|\boldsymbol{U} - \boldsymbol{V}\|_{\boldsymbol{\Sigma}^{-1}} - \|\boldsymbol{x}(0) - \boldsymbol{z}(0)\|_{\boldsymbol{\Sigma}_0^{-1}}\right]$$
$$\text{s.t. } \boldsymbol{0} = \boldsymbol{M}_{\boldsymbol{\theta}_\mathrm{t}+\mu\partial\boldsymbol{\theta}}\boldsymbol{V} + \boldsymbol{N}_{\boldsymbol{\theta}_\mathrm{t}+\mu\partial\boldsymbol{\theta}}\boldsymbol{z}(0)$$

Result 1 follows readily from Claim 1 as $\boldsymbol{\theta} = \boldsymbol{\theta}_\mathrm{t}$. Result 2 follows from Claim 2 as $\boldsymbol{\theta} = (1 + \hat{\mu})\boldsymbol{\theta}_\mathrm{t} \propto \boldsymbol{\theta}_\mathrm{t}$.

Proofs of Claim 1 and Claim 2. Notice first that $\boldsymbol{M}_{\boldsymbol{\theta}} \succ \boldsymbol{0}$ is invertible. The log-likelihood of Method 3 in (6c) is proportional to

$$-\|\boldsymbol{U} - \boldsymbol{V}\|_{\boldsymbol{\Sigma}^{-1}} - \|\boldsymbol{x}(0) - \boldsymbol{z}(0)\|_{\boldsymbol{\Sigma}_0^{-1}}. \tag{13}$$

Using $\boldsymbol{V} = -\boldsymbol{M}_{\boldsymbol{\theta}}^{-1}\boldsymbol{N}_{\boldsymbol{\theta}}\boldsymbol{z}(0)$, (13) can be written as

$$-\|\boldsymbol{M}_{\boldsymbol{\theta}}\boldsymbol{U} + \boldsymbol{N}_{\boldsymbol{\theta}}\boldsymbol{z}(0)\|_{\left(\boldsymbol{M}_{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}\boldsymbol{M}_{\boldsymbol{\theta}}\right)^{-1}} - \|\boldsymbol{x}(0) - \boldsymbol{z}(0)\|_{\boldsymbol{\Sigma}_0^{-1}}. \tag{14}$$

Then, using linearity ($\boldsymbol{M}_{\boldsymbol{\theta}} = \boldsymbol{M}_{\boldsymbol{\theta}_\mathrm{t}} + \mu\boldsymbol{M}_{\partial\boldsymbol{\theta}}$ and $\boldsymbol{N}_{\boldsymbol{\theta}} = \boldsymbol{N}_{\boldsymbol{\theta}_\mathrm{t}} + \mu\boldsymbol{N}_{\partial\boldsymbol{\theta}}$), $\boldsymbol{U} = \boldsymbol{V} + \partial\boldsymbol{V}$, and $\boldsymbol{x}(0) = \boldsymbol{z}(0) + \partial\boldsymbol{z}$, the expected value of (14) yields

$$\begin{aligned} &-\mu^2\|\boldsymbol{M}_{\partial\boldsymbol{\theta}}\boldsymbol{V} + \boldsymbol{N}_{\partial\boldsymbol{\theta}}\boldsymbol{z}(0)\|_{\left(\boldsymbol{M}_{\boldsymbol{\theta}}^T\boldsymbol{\Sigma}\boldsymbol{M}_{\boldsymbol{\theta}}\right)^{-1}} \\ &-\mathrm{trace}\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\right) - \mathrm{trace}\left(\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_0^{-1}\right) \end{aligned} \tag{15}$$

Therefore, for any $\partial\boldsymbol{\theta}$, $\mu = 0$ maximizes (15), which proves Claim 1. For $\partial\boldsymbol{\theta} = \boldsymbol{\theta}_\mathrm{t}$, $\boldsymbol{M}_{\partial\boldsymbol{\theta}}\boldsymbol{V} + \boldsymbol{N}_{\partial\boldsymbol{\theta}}\boldsymbol{z}(0) = \boldsymbol{0}$ and $\mu \in \mathbb{R}$ minimizes (15), which proves Claim 2. □

Theorem 2. Consider unconstrained linear systems of the form (12a) and cost functions (12b). Let $\boldsymbol{U} \sim \mathcal{N}(\boldsymbol{V}, \boldsymbol{\Sigma})$ and $\boldsymbol{x}(0) \sim \mathcal{N}(\boldsymbol{z}(0), \boldsymbol{\Sigma}_0)$. Then, $\boldsymbol{\theta}_\mathrm{t}$ does not result in expectation from Method 1 (result 1). Further, Method 1 is not scale-invariant (result 2).

Proof. For the considered class of dynamical systems, the cost function in (6a) is

$$\|\boldsymbol{M}_{\boldsymbol{\theta}}\boldsymbol{U} + \boldsymbol{N}_{\boldsymbol{\theta}}\boldsymbol{x}(0)\|_{\boldsymbol{I}}. \tag{16}$$

We define $\boldsymbol{\theta} = \boldsymbol{\theta}_\mathrm{t} + \mu\partial\boldsymbol{\theta}$ with $\mu \in \mathbb{R}$ and $\partial\boldsymbol{\theta} \in \mathbb{R}^p$ and $\|\partial\boldsymbol{\theta}\|_2 = 1$. Then, using linearity ($\boldsymbol{M}_{\boldsymbol{\theta}} = \boldsymbol{M}_{\boldsymbol{\theta}_\mathrm{t}} + \mu\boldsymbol{M}_{\partial\boldsymbol{\theta}}$ and $\boldsymbol{N}_{\boldsymbol{\theta}} = \boldsymbol{N}_{\boldsymbol{\theta}_\mathrm{t}} + \mu\boldsymbol{N}_{\partial\boldsymbol{\theta}}$), $\boldsymbol{U} = \boldsymbol{V} + \partial\boldsymbol{V}$, and $\boldsymbol{x}(0) = \boldsymbol{z}(0) + \partial\boldsymbol{z}$, the expected value of (16) yields

$$\begin{aligned} &\mathbb{E}\left[\|\boldsymbol{M}_{\boldsymbol{\theta}}\boldsymbol{U} + \boldsymbol{N}_{\boldsymbol{\theta}}\boldsymbol{x}(0)\|_{\boldsymbol{I}}\right] = \\ &\mu^2\left(\|\boldsymbol{M}_{\partial\boldsymbol{\theta}}\boldsymbol{V} + \boldsymbol{N}_{\partial\boldsymbol{\theta}}\boldsymbol{z}(0)\|_{\boldsymbol{I}} + \mathrm{t}_{\partial\boldsymbol{\theta},\partial\boldsymbol{\theta}}\right) + 2\mu\ \mathrm{t}_{\boldsymbol{\theta}_\mathrm{t},\partial\boldsymbol{\theta}} + \mathrm{t}_{\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\theta}_\mathrm{t}} \end{aligned} \tag{17}$$

with

$$\begin{aligned} \mathrm{t}_{\boldsymbol{\theta}_\mathrm{t},\boldsymbol{\theta}_\mathrm{t}} &= \mathrm{trace}(\boldsymbol{M}_{\boldsymbol{\theta}_\mathrm{t}}^T\boldsymbol{M}_{\boldsymbol{\theta}_\mathrm{t}}\boldsymbol{\Sigma}) + \mathrm{trace}(\boldsymbol{N}_{\boldsymbol{\theta}_\mathrm{t}}^T\boldsymbol{N}_{\boldsymbol{\theta}_\mathrm{t}}\boldsymbol{\Sigma}_0) \\ \mathrm{t}_{\boldsymbol{\theta}_\mathrm{t},\partial\boldsymbol{\theta}} &= \mathrm{trace}(\boldsymbol{M}_{\boldsymbol{\theta}_\mathrm{t}}^T\boldsymbol{M}_{\partial\boldsymbol{\theta}}\boldsymbol{\Sigma}) + \mathrm{trace}(\boldsymbol{N}_{\boldsymbol{\theta}_\mathrm{t}}^T\boldsymbol{N}_{\partial\boldsymbol{\theta}}\boldsymbol{\Sigma}_0) \\ \mathrm{t}_{\partial\boldsymbol{\theta},\partial\boldsymbol{\theta}} &= \mathrm{trace}(\boldsymbol{M}_{\partial\boldsymbol{\theta}}^T\boldsymbol{M}_{\partial\boldsymbol{\theta}}\boldsymbol{\Sigma}) + \mathrm{trace}(\boldsymbol{N}_{\partial\boldsymbol{\theta}}^T\boldsymbol{N}_{\partial\boldsymbol{\theta}}\boldsymbol{\Sigma}_0). \end{aligned}$$

The optimal μ minimizing (17) is not necessarily 0 but a function of $\partial\boldsymbol{\theta}$, which proves result 1:

$$\mu = -\frac{\mathrm{t}_{\boldsymbol{\theta}_\mathrm{t},\partial\boldsymbol{\theta}}}{\|\boldsymbol{M}_{\partial\boldsymbol{\theta}}\boldsymbol{V} + \boldsymbol{N}_{\partial\boldsymbol{\theta}}\boldsymbol{z}(0)\|_{\boldsymbol{I}} + \mathrm{t}_{\partial\boldsymbol{\theta},\partial\boldsymbol{\theta}}}.$$

For $\partial\boldsymbol{\theta} = \boldsymbol{\theta}_\mathrm{t}$, $\mu = -1$ and $\boldsymbol{\theta} = \boldsymbol{\theta}_\mathrm{t} - \boldsymbol{\theta}_\mathrm{t} = \boldsymbol{0}$, i.e., the estimator is not scale-invariant, which proves result 2. □

6. SIMULATION RESULTS

In this section, we utilize the three discussed methods for learning the cost function's parameters.

6.1 Simulation Setup

System dynamics and constraints We consider one linear system and one nonlinear system with dynamics

$$\boldsymbol{x}(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k) \tag{18a}$$

$$\boldsymbol{x}(k+1) = \begin{bmatrix} 1 & 1-u(k)^2 \\ 0 & 1 \end{bmatrix} \boldsymbol{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k). \tag{18b}$$

The inputs are constrained as $|u(k)| \leq 1$.

Cost function The true cost function is chosen as

$$\boldsymbol{\theta}_{\mathrm{t}}{}^T \boldsymbol{\phi}(\boldsymbol{V}, \boldsymbol{Z})$$

with

$$\boldsymbol{\theta}_{\mathrm{t}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{\phi}(\boldsymbol{V}, \boldsymbol{Z}) = \begin{bmatrix} \sum_{k=0}^{N} z_{1,k}^2 \\ \sum_{k=0}^{N} z_{2,k}^2 \\ \sum_{k=0}^{N-1} (v_{k+1} - v_k)^2 \\ \sum_{k=0}^{N} v_k^2 \end{bmatrix}$$

with $\boldsymbol{z}_k = [z_{1,k}\ z_{2,k}]^T$ and $N = 10$. As the cost function is scale-invariant, we fix one parameter and learn

$$\hat{\boldsymbol{\theta}}^T = \begin{bmatrix} \hat{\theta}_1 & \hat{\theta}_2 & \hat{\theta}_3 & 1 \end{bmatrix}.$$

Data generation In order to generate the data, we sample the initial conditions $\boldsymbol{z}(0)$ from $\boldsymbol{z}(0) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Using $\boldsymbol{z}(0)$, we obtain the optimal input sequence $\boldsymbol{V}$ using (2). Then, the (sub-optimal) demonstration is generated as $\boldsymbol{U} \sim \mathcal{N}(\boldsymbol{V}, \sigma_u^2 \boldsymbol{I}_{10})$ and $\boldsymbol{x}(0) \sim \mathcal{N}(\boldsymbol{z}(0), \sigma_0^2 \boldsymbol{I}_2)$, where σ_u and σ_0 are varied logarithmically as

$$\begin{aligned} \sigma_u, \sigma_0 \in \{&0.0001, 0.000215, 0.000464, 0.001, \\ &0.00215, 0.00464, 0.01, \\ &0.0215, 0.0464, 0.1, \\ &0.215, 0.464, 1\} \end{aligned}$$

Evaluation criterion We evaluate the learned parameters by comparing $\boldsymbol{V}$ resulting from (2) with $\boldsymbol{\theta}_{\mathrm{t}}$ and $\hat{\boldsymbol{V}}$ resulting from (2) with $\hat{\boldsymbol{\theta}}$ as

$$\text{error} = \frac{\|\hat{\boldsymbol{V}} - \boldsymbol{V}\|_2}{\|\boldsymbol{V}\|_2}. \tag{19}$$

6.2 Learning Results

For every tuple $\{\sigma_u, \sigma_0\}$, we repeat the data generation process 1000 times and learn $\hat{\boldsymbol{\theta}}$ using the three inverse learning methods.

Fig. 2 illustrates the median of the error in (19) for the 1000 trials for $\{\sigma_u, \sigma_0\}$ and the three learning methods. For $\sigma_u = \sigma_0 = 0$, error $= 0$ for all three methods. In the presence of noise $\sigma_u, \sigma_0 > 0$, the learning results degrade differently. The error increases for larger standard deviations $\{\sigma_u, \sigma_0\}$ for all three methods. However, it can be seen that with increased noise levels (sub-optimal data), the error increases more quickly for Method 1, whereas the errors remain smaller for Method 2 and Method 3. Now, consider small σ_u. For increased σ_0, the error for

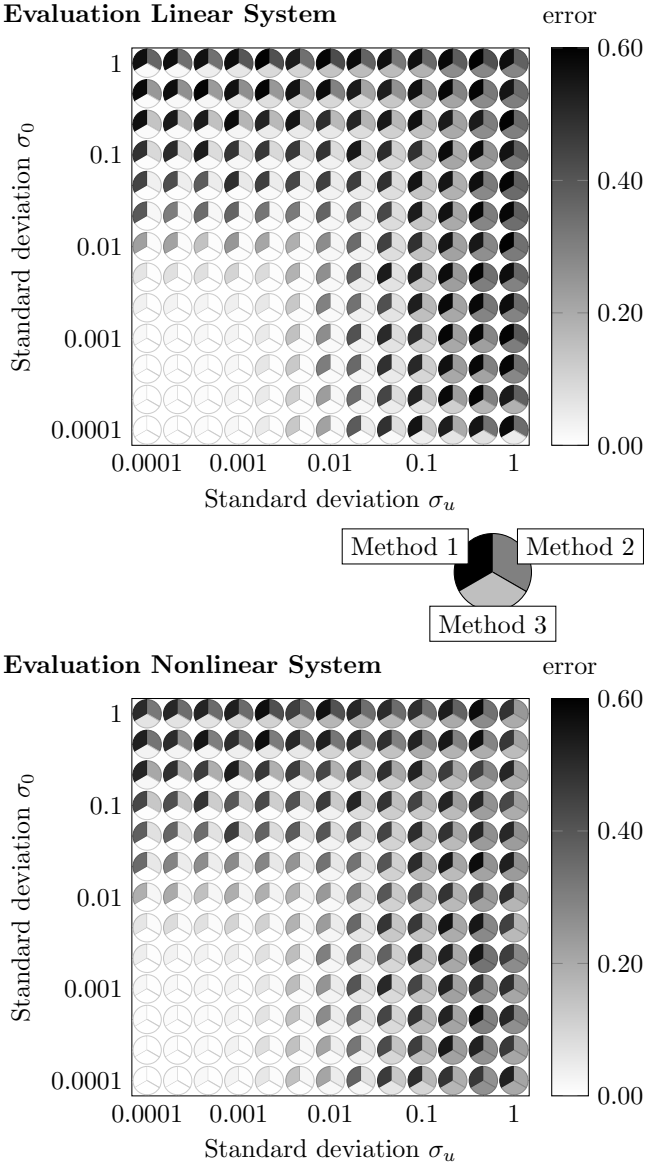

Fig. 2. Median of error $\in [0, 0.6]$ for different standard deviations σ_u and σ_0 (color map from white to black) for the three inverse learning methods.

Method 3 is significantly smaller than Method 2, which is expected since Method 3 optimizes over $\boldsymbol{z}(0)$. For small σ_0 and increasing σ_u, Method 3 also outperforms Method 2, which suggests that optimizing over the initial condition is advantageous even for small uncertainties in the initial conditions. Note that the standard deviations $\sigma_u = \sigma_0 > 0.1$ are unrealistically high as $|u| \leq 1$.

Fig. 3 shows a more detailed statistical evaluation of the error in (19) for the 1000 trials and $\sigma_u = \sigma_0$. First, it can be seen that the estimation with Method 2 and Method 3 have a lower error compared to Method 1. The errors tend to be lower for the nonlinear system compared to the linear system, which can best be seen for $\sigma_0 = \sigma_u < 0.01$. The relatively low errors of Method 2 and Method 3 suggest the superiority of the maximum likelihood formulation over the convex relaxation approach of Method 1 measured with respect to the predictive performance, i.e., (19).

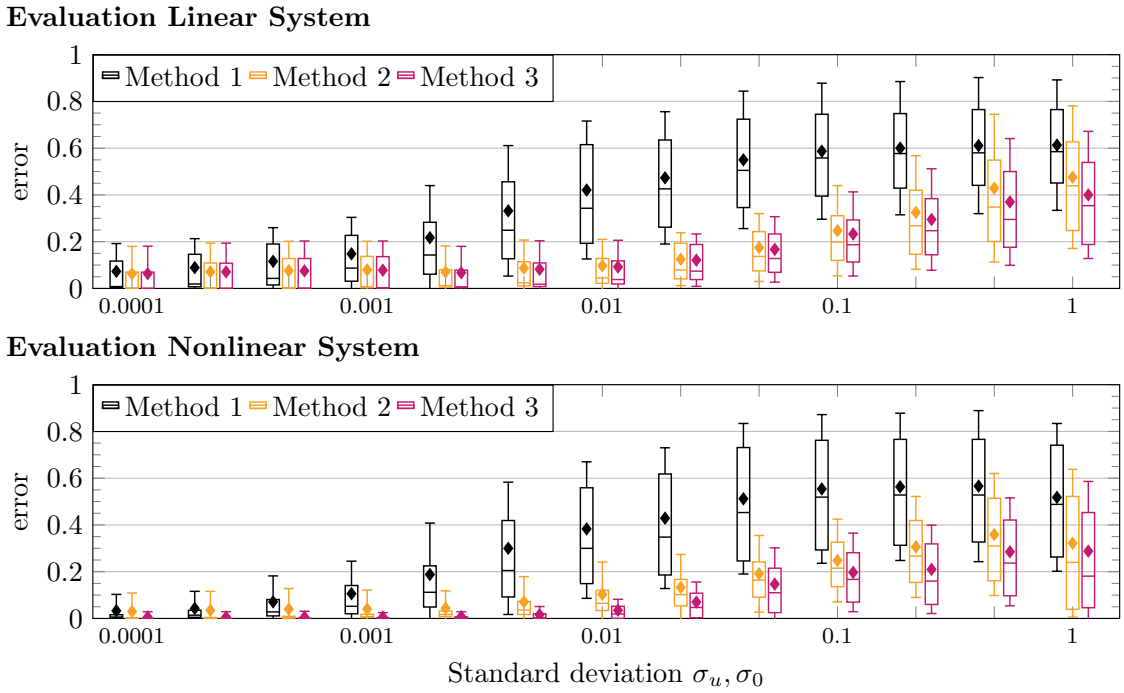

Fig. 3. Evaluation of the error in (19) for the three inverse learning methods for different noise levels $\sigma_u = \sigma_0$. The diamond symbol and vertical line represent the mean and the median, respectively; the box edges represent the 25th and the 75th percentiles; and the whiskers represent the standard deviation.

6.3 Computation Time

Table 1 states the median computation time for all samples of the initial conditions for the three learning methods using MATLAB with the hardware configuration: 3.1 GHz Intel Core i7, 16 GB 1867 MHz DDR3, and Intel Iris Graphics 6100 1536 MB. Method 1 is convex and, therefore, the computation is cheap and requires less than one second. Method 2 is computationally slightly more involved but can still be solved in around one second. Method 3 is more demanding as also the initial condition, $\boldsymbol{z}(0)$, is an optimization variable.

Table 1. Computation time

System	Method	Median over all samples
Linear (18a)	Method 1	$T_{\mathrm{L,M1}} = 0.148\mathrm{s}$
	Method 2	$T_{\mathrm{L,M2}} = 1.19\mathrm{s}$ $(8.04 T_{\mathrm{L,M1}})$
	Method 3	$T_{\mathrm{L,M3}} = 3.07\mathrm{s}$ $(20.8 T_{\mathrm{L,M1}})$
Nonlinear (18b)	Method 1	$T_{\mathrm{NL,M1}} = 0.142\mathrm{s}$
	Method 2	$T_{\mathrm{NL,M2}} = 0.584\mathrm{s}$ $(4.12 T_{\mathrm{NL,M1}})$
	Method 3	$T_{\mathrm{NL,M3}} = 1.44\mathrm{s}$ $(10.2 T_{\mathrm{NL,M1}})$

Fig. 4 shows a more detailed statistical evaluation of the computation times for $\sigma_u = \sigma_0$. The three learning methods have their respective peak computation times at different noise levels, i.e., Method 1's maximum computation times are highest for lower noise levels (peak of median at $\sigma_u = \sigma_0 = 0.0001$); Method 2's maximum times occur for slightly higher noise levels (peak at $\sigma_u = \sigma_0 = 0.00215$); whereas Method 3's peak is for high noise levels (peak at $\sigma_u = \sigma_0 = 0.0215$). For all methods and noise levels, the mean value is higher than the median, which is expected as the median is less susceptible to outliers, i.e., instances with particularly long computation times.

7. CONCLUSION

This paper presented three inverse optimal control methods; one method that uses a convex relaxation of the KKT optimality conditions and two methods that combine the KKT conditions with maximum likelihood estimation. It proposed a branch-and-bound-style algorithm for the maximum likelihood formulation, which is based on likelihood arguments to systematically deal with constraints in the presence of noisy data. A simulation study exemplified the three inverse learning methods with both a constrained, linear and a nonlinear system. The results showed that the likelihood estimation methods can be implemented quite efficiently and yield robust learning results, whereas the convex method is computationally efficient but less robust to noise in the training data.

REFERENCES

Armijo, L. (1966). Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1), 1–3.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Chou, G., Ozay, N., and Berenson, D. (2020). Learning constraints from locally-optimal demonstrations under cost function uncertainty. *IEEE Robot. and Automat. Lett.* doi:10.1109/LRA.2020.2974427.

Chou, G., Berenson, D., and Ozay, N. (2018). Learning constraints from demonstrations. *arXiv:1812.07084*.

Englert, P., Vien, N.A., and Toussaint, M. (2017). Inverse KKT: Learning cost functions of manipulation tasks from demonstrations. *Int. J. Robot. Res.*, 36(13–14), 1474–1488.

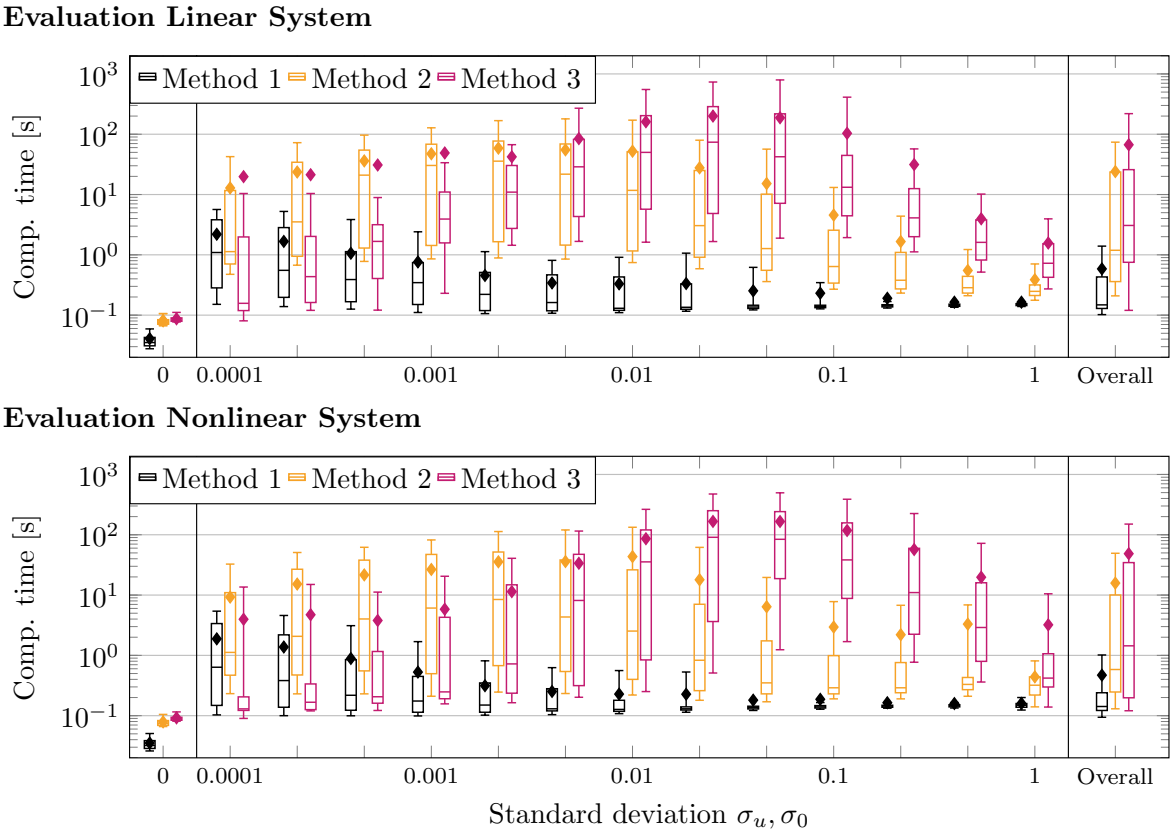

Fig. 4. Statistical evaluation of the computation time for the three inverse learning methods with $\sigma_u = \sigma_0$. The diamond symbol and vertical line represent the mean and the median, respectively; the box edges represent the 75th and the 25th percentiles; and the whiskers represent the 90th and the 10th percentiles.

Esfahani, P.M., Shafieezadeh-Abadeh, S., Hanasusanto, G.A., and Kuhn, D. (2018). Data-driven inverse optimization with imperfect information. *Math. Program.*, 167(1), 191–234.

Finn, C., Levine, S., and Abbeel, P. (2016). Guided cost learning: Deep inverse optimal control via policy optimization. In *33rd Int. Conf. Mach. Learn.*, 49–58.

Hewing, L., Wabersich, K.P., Menner, M., and Zeilinger, M.N. (2020). Learning-based model predictive control: Toward safe learning in control. *Annu. Rev. Control, Robot., and Auton. Syst.*, 3, 269–296.

Kalman, R.E. (1964). When is a linear control system optimal? *J. Basic Eng.*, 86(1), 51–60.

Karush, W. (1939). *Minima of functions of several variables with inequalities as side constraints.* Master's thesis, Dept. of Mathematics, Univ. of Chicago.

Kuhn, H.W. and Tucker, A.W. (1951). Nonlinear programming. In *2nd Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press.

Levine, S. and Koltun, V. (2012). Continuous inverse optimal control with locally optimal examples. In *29th Int. Conf. Machine Learning*, 475–482.

Menner, M., Berntorp, K., Zeilinger, M.N., and Di Cairano, S. (2019a). Inverse learning for human-adaptive motion planning. In *58th IEEE Conf. Decision and Control*, 809–815.

Menner, M., Worsnop, P., and Zeilinger, M.N. (2019b). Constrained inverse optimal control with application to a human manipulation task. *IEEE Trans. Control Syst. Technol.* doi:10.1109/TCST.2019.2955663.

Menner, M. and Zeilinger, M.N. (2018). Convex formulations and algebraic solutions for linear quadratic inverse optimal control problems. In *Eur. Control Conf.*, 2107–2112.

Mombaur, K., Truong, A., and Laumond, J.P. (2010). From human to humanoid locomotion: an inverse optimal control approach. *Auton. Robots*, 28(3), 369–383.

Murty, K.G. and Yu, F.T. (1988). *Linear complementarity, linear and nonlinear programming*, volume 3. Berlin: Heldermann.

Ziebart, B.D., Maas, A., Bagnell, J.A., and Dey, A.K. (2008). Maximum entropy inverse reinforcement learning. In *AAAI Conf. Artif. Intell.*, 1433–1438.