On a Probabilistic Approach for Inverse Data-Driven Optimal Control

Émiland Garrabé, Hozefa Jesawada, Carmen Del Vecchio, and Giovanni Russo*

Abstract— We consider the problem of estimating the possibly non-convex cost of an agent by observing its interactions with a nonlinear, non-stationary and stochastic environment. For this inverse problem, we give a result that allows to estimate the cost by solving a convex optimization problem. To obtain this result we also tackle a forward problem. This leads to the formulation of a finite-horizon optimal control problem for which we show convexity and find the optimal solution. Our approach leverages certain probabilistic descriptions that can be obtained both from data and/or from first-principles. The effectiveness of our results, which are turned in an algorithm, is illustrated via simulations on the problem of estimating the cost of an agent that is stabilizing the unstable equilibrium of a pendulum.

I. INTRODUCTION

Inferring the intents of an agent by observing its interactions with the environment is crucial to many scientific domains, with applications spanning across e.g., engineering, psychology, economics, management and computer science. Inverse optimal control/reinforcement learning (IOC/IRL) refers to both the problem and the class of methods to infer the cost/reward driving the actions of an agent by observing its inputs/outputs [1]. Tackling this problem is relevant to sequential decision-making [2] and can be useful to design data-driven control systems with humans-in-the-loop as well as incentive schemes in sharing economy settings [3].

In this context, a key challenge in IOC/IRL lies in the fact that the underlying optimization can become ill-posed even when the environment dynamics is linear, deterministic and the cost is convex. Motivated by this, we propose an approach to estimate possibly non-convex costs when the underlying dynamics is nonlinear, non-stationary and stochastic. The approach leverages probabilistic descriptions that can be obtained directly from data and/or from first-principles. Also, the results allow to obtain cost estimates by solving a convex optimization problem.

Related works: we briefly survey a number of works related to the results and methodological framework of this paper and we refer to e.g., [1] for a detailed review on inverse problems across learning and control. As remarked in [4], IOC methods were originally developed to find *control histories* to *produce* desired outputs. More recently, driven by the advances in computational power to process datasets, IOC and IRL methods have gained considerable attention.

E. Garrabé and H. Jesawada are joint first authors. H. Jesawada, and C. Del Vecchio are with the Department of Engineering, University of Sannio, Benevento, Italy, E. Garrabé and G. Russo are with the Department of Information and Electrical Engineering & Applied Mathematics, University of Salerno, Italy. Emails: {jesawada, c.delvecchio}@unisannio.it, {egarrabe, giovarusso}@unisa.it.

In [5] a maximum entropy-based approach is proposed for stationary Markov Decision Processes (MDPs), which is based on a backward/forward pass scheme (see also [6] for linear multi-agent games). In [7] a local approximation of the reward is used and in [8] Gaussian processes are exploited, leading to a method that requires matrix inversion operations and to optimization problems that are not convex in general. Instead, in [9] manipulation tasks are considered and path integrals are used to learn the cost, while in [10] learning is achieved via deep networks for stationary MDPs. A model-based IRL approach for deterministic systems is presented in [11] for online cost estimation and [12] tackles the IRL problem in the context of deterministic multiplayer non-cooperative games. The framework of linearly solvable MDPs is instead leveraged in [13] and, while it has the advantages of avoiding solving forward MDPs in each iteration of the optimization and of yielding a convex optimizatiopn problem, it also assumes that the agent can specify directly the state transition. We also recall [14], where a risk-sensitive IRL method is proposed for stationary MDPs assuming that the expert policy belongs to the exponential distribution. Also, in the context of IOC, [15] considers stochastic dynamics and proposes an approach to learn the parameter of a control regularizer. The IOC problem for known nonlinear deterministic systems with quadratic cost function in the input is also considered in [16]. Finally, as we shall see, in order to obtain our results on the inverse problem we also solve a forward problem that involves optimizing, over probability functions, costs that contain a Kullback-Leibler divergence term. We refer to e.g., [2] for a survey on this class of problems in the context of sequential decision-making across learning and control.

Contributions: we introduce a number of results to estimate the possibly non-convex and non-stationary cost of an agent by observing its interactions with the environment, which can be nonlinear, non-stationary and stochastic, and for which just a probabilistic description is known. This probabilistic description can be obtained directly from data. Specifically, by leveraging a probabilistic framework, we give a result that enables to estimate the cost by solving an optimization problem that is convex even when the dynamics is nonlinear, non-stationary and stochastic. In order to obtain our result on the inverse problem, which leverages maximum likelihood arguments, we also tackle a forward problem. This leads to formulation of a finite-horizon optimal control problem with randomized policies as decision variables. For this problem, we find the optimal solution and show that this is a probability mass function with an exponential twisted kernel (this is a class of policies that is often assumed in works

on IRL). Also, we turn our result on cost estimation in an algorithm and its effectiveness is illustrated via simulations on the problem of estimating the cost of an agent that is stabilizing the unstable equilibrium of a pendulum.

While our results are inspired by works on IRL/IOC, this paper offers a number of key technical novelties. First, we do not require that the agent can specify its state transitions and we do not assume that the expert policy is stationary. Despite this, our approach leads to an optimization problem to estimate the cost that we prove to be convex. Moreover, our approach does not require running and solving forward problems in each iteration of the optimization and it does not require the underlying dynamics to be deterministic.

II. MATHEMATICAL PRELIMINARIES AND PROBLEM FORMULATION

Sets are in *calligraphic* and vectors in **bold**. A random variable is denoted by $\mathbf{V}$ and its realization is $\mathbf{v}$. We denote the *probability mass function*, or simply *pmf*, of $\mathbf{V}$ by $p(\mathbf{v})$ and we let $\mathcal{D}$ be the convex subset of pmfs. Whenever we take the sums involving pmfs we always assume that the sum exists. The expectation of a function $\mathbf{h}(\cdot)$ of $\mathbf{V}$ is $\mathbb{E}_p[\mathbf{h}(\mathbf{V})] := \sum_{\mathbf{v}} \mathbf{h}(\mathbf{v})p(\mathbf{v})$, where the sum is over the support of $p(\mathbf{v})$; whenever it is clear from the context, we omit the subscript in the sum. The joint pmf of $\mathbf{V}_1$ and $\mathbf{V}_2$ is denoted by $p(\mathbf{v}_1, \mathbf{v}_2)$ and the conditional pmf of $\mathbf{V}_1$ with respect to $\mathbf{V}_2$ is $p(\mathbf{v}_1 \mid \mathbf{v}_2)$. Countable sets are denoted by $\{w_k\}_{k_1:k_n}$, where w_k is the generic set element, k_1 (k_n) is the index of the first (last) element and $k_1 : k_n$ is the set of consecutive integers between (including) k_1 and k_n. A pmf of the form $p(\mathbf{v}_0, \ldots, \mathbf{v}_N)$ is compactly written as $p_{0:N}$ (by definition $p_{k:k} := p_k(\mathbf{v}_k)$). We use the shorthand notation $p_{k|k-1}$ to denote $p_k(\mathbf{v}_k \mid \mathbf{v}_{k-1})$. Also, functionals are denoted by capital calligraphic characters with arguments within curly brackets. We make use of the Kullback-Leibler (KL [17]) divergence, a measure of the proximity of the pair of pmfs $p(\mathbf{v})$ and $q(\mathbf{v})$, defined as $\mathcal{D}_{\mathrm{KL}}(p \mid\mid q) := \sum_{\mathbf{v}} p(\mathbf{v}) \ln(p(\mathbf{v})/q(\mathbf{v}))$. We also recall here the chain rule for the KL-divergence:

Lemma 1. *Let $\mathbf{V}$ and $\mathbf{Z}$ be two (possibly, vector) random variables and let $f(\mathbf{v}, \mathbf{z})$ and $g(\mathbf{v}, \mathbf{z})$ be two joint pmfs. Then, the following identity holds:*

$$\mathcal{D}_{KL}(f(\mathbf{v}, \mathbf{z}) \mid\mid g(\mathbf{v}, \mathbf{z})) = \mathcal{D}_{KL}(f(\mathbf{v}) \mid\mid g(\mathbf{v})) + \mathbb{E}_{f(\mathbf{v})}[\mathcal{D}_{KL}(f(\mathbf{z} \mid \mathbf{v}) \mid\mid g(\mathbf{z} \mid \mathbf{v}))]. \quad (1)$$

A. Set-up of the Control Problem

We let $\mathbf{X}_k \in \mathcal{X} \subseteq \mathbb{Z}^n$ be the system state at time step k and $\mathbf{U}_k \in \mathcal{U} \subseteq \mathbb{Z}^p$ be the control input at time step k. Throughout the paper, the time indexing is chosen so that the control input $\mathbf{u}_k$ is determined based on information available up to $k-1$ and when input $\mathbf{u}_k$ is applied, the system transitions from state $\mathbf{x}_{k-1}$ to state $\mathbf{x}_k$. We let: (i) $\boldsymbol{\Delta}_k := (\mathbf{x}_{k-1}, \mathbf{u}_k)$ be the input-state data pair collected from the system when it is in state $\mathbf{x}_{k-1}$ and $\mathbf{u}_k$ is applied; (ii) $\boldsymbol{\Delta}_{0:N} := (\{\boldsymbol{\Delta}_k\}_{1:N}, \mathbf{x}_N)$ be the dataset over the time horizon $\mathcal{T} := 0 : N$. We also denote by $p_{0:N} := p(\boldsymbol{\Delta}_{0:N})$

the joint pmf of the dataset. We use the wording *dataset* to denote a sequence of input-state data. Sometimes, in applications one has available a collection of datasets, which we term as *database* in what follows.

Remark 1. *As noted in [18], $p_{0:N}$ is a black box type model that can be obtained directly from the data and does not require assumptions on the underlying dynamics.*

We now make the standard assumption that the Markov property holds. Then, $p_{0:N}$ can be conveniently partitioned:

$$p_{0:N} = p_0(\mathbf{x}_0) \prod_{k=1}^{N} p_{k|k-1} = p_0(\mathbf{x}_0) \prod_{k=1}^{N} p_{k|k-1}^{(x)} p_{k|k-1}^{(u)}, \quad (2)$$

where we used the shorthand notation $p_{k|k-1}^{(x)} := p_k^{(x)}(\mathbf{x}_k \mid \mathbf{u}_k, \mathbf{x}_{k-1})$, $p_{k|k-1}^{(u)} := p_k^{(u)}(\mathbf{u}_k \mid \mathbf{x}_{k-1})$ and $p_{k|k-1} := p_{k|k-1}^{(x)} p_{k|k-1}^{(u)} = p(\mathbf{x}_k, \mathbf{u}_k \mid \mathbf{x}_{k-1})$. Also, it is useful to define the joint pmf $\bar{p}_{k-1:k} := p_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_k)$. We say that (2) is the probabilistic description of the system.

Remark 2. *In (2), the pmf $p_k^{(x)}(\mathbf{x}_k \mid \mathbf{u}_k, \mathbf{x}_{k-1})$ describes in probabilistic terms the evolution of system, while $p_k^{(u)}(\mathbf{u}_k \mid \mathbf{x}_{k-1})$ is the randomized policy from which, at time step k, the control input is sampled.*

We let $c_k : \mathcal{X} \to \mathbb{R}$ be the cost, at time-step k, associated to a given state, $\mathbf{x}_k$. Then, the expected cost incurred when the system is in state $\mathbf{x}_{k-1}$ and input $\mathbf{u}_k$ is applied is given by $\mathbb{E}_{p_{k|k-1}^{(x)}}[c_k(\mathbf{X}_k)]$. To address the inverse problem for cost estimation (see Section II-B for the problem statement) we first tackle the following forward problem:

Problem 1. *Given a joint pmf*

$$q_{0:N} := q_0(\mathbf{x}_0) \prod_{k=1}^{N} q_k^{(x)}(\mathbf{x}_k \mid \mathbf{u}_k, \mathbf{x}_{k-1}) q_k^{(u)}(\mathbf{u}_k \mid \mathbf{x}_{k-1}). \quad (3)$$

Find the sequence of pmfs, $\left\{p_{k|k-1}^{(u)}{}^*\right\}_{1:N}$, such that:

$$\left\{p_{k|k-1}^{(u)}{}^*\right\}_{1:N} \in \underset{\left\{p_{k|k-1}^{(u)}\right\}_{1:N}}{\arg\min} \left\{ \mathcal{D}_{KL}(p_{0:N} \mid\mid q_{0:N}) \right.$$
$$\left. + \sum_{k=1}^{N} \mathbb{E}_{\bar{p}_{k-1:k}}\left[\mathbb{E}_{p_{k|k-1}^{(x)}}[c_k(\mathbf{X}_k)]\right] \right\}$$
$$s.t.\ p_{k|k-1}^{(u)} \in \mathcal{D} \quad \forall k \in \mathcal{T}. \quad (4)$$

Throughout the paper we make the following standard

Assumption 1. *The optimal cost of Problem 1 is bounded.*

For our derivations, it is also useful to introduce $q_{k|k-1}^{(x)} := q_k^{(x)}(\mathbf{x}_k \mid \mathbf{u}_k, \mathbf{x}_{k-1})$, $q_{k|k-1}^{(u)} := q_k^{(u)}(\mathbf{u}_k \mid \mathbf{x}_{k-1})$ and $q_{k|k-1} := q_{k|k-1}^{(x)} q_{k|k-1}^{(u)} = q(\mathbf{x}_k, \mathbf{u}_k \mid \mathbf{x}_{k-1})$.

As we shall see, the solution of Problem 1 is a sequence of randomized policies. At each k, the control input applied to the system, i.e. $\mathbf{u}_k^*$, is sampled from $p_{k|k-1}^{(u)}{}^*$. In the

cost functional of Problem 1, minimizing the second term minimizes the expected agent cost, while minimizing the first term amounts at minimizing the discrepancy between $p_{0:N}$ and $q_{0:N}$. Hence, the first term in the cost functional can be thought of as a regularizer, biasing the behavior of the closed loop system towards the pmf $q_{0:N}$. Typically, $q_{0:N}$ acts as passive dynamics [19], [20] or expresses desired behavior from demonstration databases [21] See also [2] for a survey on sequential decision-making problems that involve minimizing this class of cost functionals.

B. The Inverse Control Problem

The inverse control problem we consider consists in estimating both the cost-to-go for the agent, say $\bar{c}_k(\cdot)$, and the agent cost $c_k(\cdot)$ given a set of observed states/inputs sampled from $p_{k|k-1}^{(x)}$ and from the agent policy. In what follows, we denote by $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{u}}_k$ the observed state and control input at time-step k. We also make the following:

Assumption 2. *There exist some* $\mathbf{w}_k := [w_{k,1}, \ldots, w_{k,F}]^T$ *such that* $\bar{c}_k(\mathbf{x}_k) = -\mathbf{w}_k^T \mathbf{h}(\mathbf{x}_k)$, *where* $\mathbf{h}(\mathbf{x}_k) := [h_1(\mathbf{x}_k), \ldots, h_F(\mathbf{x}_k)]^T$ *and* $h_i : \mathcal{X} \to \mathbb{R}$ *are known functions,* $i = 1, \ldots, F$.

In what follows, we say that $\mathbf{h}(\mathbf{x}_k)$ is the features vector. The assumption, which is rather common in the literature see e.g., [5], [6], [9], [11], [13], formalizes the fact that the cost-to-go can be expressed as a linear combination of given, possibly nonlinear, *features* [22]. With our results in Section III-B we propose a maximum likelihood estimator for the cost (see e.g., [23] for a maximum likelihood framework for linear systems in the context of data-driven control).

III. Main Results

A. Computing the Optimal Policy for Problem 1

With the next result we give the solution to Problem 1.

Theorem 1. *Consider Problem 1 and let Assumption 1 hold. Then:*

(i) *the problem has the unique solution* $\{p_{k|k-1}^{(u)}{}^*\}_{1:N}$, *with*

$$p_{k|k-1}^{(u)}{}^* = \frac{\bar{p}_{k|k-1}^{(u)} \exp\left(-\mathbb{E}_{p_{k|k-1}^{(x)}}\left[\bar{c}_k(\mathbf{X}_k)\right]\right)}{\sum_{\mathbf{u}_k} \bar{p}_{k|k-1}^{(u)} \exp\left(-\mathbb{E}_{p_{k|k-1}^{(x)}}\left[\bar{c}_k(\mathbf{X}_k)\right]\right)},$$
(5)

where

$$\bar{p}_{k|k-1}^{(u)} := q_{k|k-1}^{(u)} \exp\left(-\mathcal{D}_{KL}\left(p_{k|k-1}^{(x)} \| q_{k|k-1}^{(x)}\right)\right),$$

and where $\bar{c}_k : \mathcal{X} \to \mathbb{R}$ *is obtained via the backward*

recursion

$$\bar{c}_k(\mathbf{x}_k) = c_k(\mathbf{x}_k) - \hat{c}_k(\mathbf{x}_k),$$

$$\hat{c}_k(\mathbf{x}_k) = \ln\left(\mathbb{E}_{q_{k+1|k}^{(u)}}\left[\exp\left(-\mathcal{D}_{KL}\left(p_{k+1|k}^{(x)} \| q_{k+1|k}^{(x)}\right)\right.\right.\right.$$
$$\left.\left.\left. -\mathbb{E}_{p_{k+1|k}^{(x)}}\left[\bar{c}_{k+1}(\mathbf{X}_{k+1})\right]\right)\right]\right),$$

$$\mathcal{D}_{KL}\left(p_{N+1|N}^{(x)} \| q_{N+1|N}^{(x)}\right)$$
$$+ \mathbb{E}_{p_{N+1|N}^{(x)}}\left[\bar{c}_{N+1}(\mathbf{X}_{N+1})\right] = 0;$$
(6)

(ii) *the corresponding minimum is given by:*

$$-\sum_{k=1}^{N} \mathbb{E}_{\bar{p}_{k-1}}\left[\hat{c}_{k-1}(\mathbf{X}_{k-1})\right],$$
(7)

where $\bar{p}_{k-1} := p_{k-1}(\mathbf{x}_{k-1})$.

Sketch of the proof. The full proof is omitted here for brevity and will be presented elsewhere. We give here a sketch of the proof, which is by induction.

Step 1. Consider the cost functional in (4). By means of Lemma 1, Problem 1 can be recast as the sum of the following two sub-problems:

$$\min_{\{p_{k|k-1}^{(u)}\}_{1:N-1}} \left\{ \mathcal{D}_{KL}\left(p_{0:N-1} \| q_{0:N-1}\right) \right.$$
$$\left. + \sum_{k=1}^{N-1} \mathbb{E}_{\bar{p}_{k-1:k}}\left[\mathbb{E}_{p_{k|k-1}^{(x)}}\left[c_k(\mathbf{X}_k)\right]\right] \right\}$$
$$s.t.\ p_{k|k-1}^{(u)} \in \mathcal{D}\ \ \forall k \in 1 : N-1,$$
(8a)

and

$$\min_{p_{N|N-1}^{(u)}} \left\{ \mathbb{E}_{\bar{p}_{N-1}}\left[\mathcal{D}_{KL}\left(p_{N|N-1} \| q_{N|N-1}\right) \right.\right.$$
$$\left.\left. + \mathbb{E}_{p_{N|N-1}}\left[c_N(\mathbf{X}_N)\right]\right] \right\}$$
$$s.t.\ p_{N|N-1}^{(u)} \in \mathcal{D}.$$
(8b)

Hence, the minimum of (8b) is $\mathbb{E}_{\bar{p}_{N-1}}\left[\mathcal{C}_N\left\{p_{N|N-1}^{(u)}{}^*\right\}\right]$, with $\mathcal{C}\left\{p_{N|N-1}^{(u)}{}^*\right\}$ being the optimal cost obtained by solving

$$\min_{p_{N|N-1}^{(u)}} \mathcal{D}_{KL}\left(p_{N|N-1} \| q_{N|N-1}\right) + \mathbb{E}_{p_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right]$$
$$s.t.\ p_{N|N-1}^{(u)} \in \mathcal{D},$$
(9)

where we set $\bar{c}_N(\mathbf{x}_N) := c_N(\mathbf{x}_N) + \hat{c}_N(\mathbf{x}_N)$, $\hat{c}_N(\mathbf{x}_N) = 0$. This corresponds to the recursion in (6) at $k = N$.
Step 2. The next step is to show that the problem in (9) can be conveniently written as

$$\min_{p_{N|N-1}^{(u)}} \left\{ \mathbb{E}_{p_{N|N-1}^{(u)}}\left[\mathcal{D}_{KL}\left(p_{N|N-1}^{(x)} \| q_{N|N-1}^{(x)}\right) \right.\right.$$
$$\left.\left. + \mathbb{E}_{p_{N|N-1}^{(x)}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right] + \alpha(\mathbf{x}_{N-1})\right\}$$
$$s.t.\ p_{N|N-1}^{(u)} \in \mathcal{D},$$
(10)

where $\alpha(\mathbf{x}_{N-1}) := \mathcal{D}_{\mathsf{KL}}\left(p_{N|N-1}^{(u)} \| q_{N|N-1}^{(u)}\right)$. By studying the second variation of the cost functional, it can be shown that this is strictly convex in the decision variable $p_{N|N-1}^{(u)}$. Hence, since the subset $\mathcal{D}$ is convex, the problem in (10) is a convex optimization problem.

Step 3. We find the solution to the problem in (9) by using the equivalent formulation given in (10). Since the problem in (10) is convex with a strictly convex cost functional, the unique optimal solution can be found by imposing the stationarity conditions on the Lagrangian, which is given by:

$$
\mathcal{L}(p_{N|N-1}^{(u)}, \lambda_N) = \mathbb{E}_{p_{N|N-1}^{(u)}}\left[\mathcal{D}_{\mathsf{KL}}\left(p_{N|N-1}^{(x)} \| q_{N|N-1}^{(x)}\right) + \right.
$$
$$
\left. \mathbb{E}_{p_{N|N-1}^{(x)}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right] + \mathcal{D}_{\mathsf{KL}}\left(p_{N|N-1}^{(u)} \| q_{N|N-1}^{(u)}\right)
$$
$$
+ \lambda_N \left(\sum_{\mathbf{u}_k} p_{N|N-1}^{(u)} - 1\right),
$$
$$(11)$$

where λ_N is the Lagrange multiplier corresponding to the constraint $p_{N|N-1}^{(u)} \in \mathcal{D}$. Now, by imposing the first order stationarity conditions on $\mathcal{L}(p_{N|N-1}^{(u)}, \lambda_N)$, it can be shown that the unique optimal solution is given by:

$$
p_{N|N-1}^{(u)}{}^* = \frac{\bar{p}_{N|N-1}^{(u)} \exp\left(-\mathbb{E}_{p_{N|N-1}^{(x)}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right)}{\sum_{\mathbf{u}_N} \bar{p}_{N|N-1}^{(u)} \exp\left(-\mathbb{E}_{p_{N|N-1}^{(x)}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right)}.
$$
$$(12)$$

This is the optimal solution given in (5) for $k = N$, with $\bar{c}_N(\mathbf{x}_N)$ generated via the backward recursion in (6). Hence, the minimum for the sub-problem in (8b) is

$$
\mathbb{E}_{\bar{p}_{N-1}}\left[\mathcal{C}_N\left\{p_{N|N-1}^{(u)}{}^*\right\}\right] = -\mathbb{E}_{\bar{p}_{N-1}}\left[\hat{c}_{N-1}(\mathbf{X}_{N-1})\right], \quad (13)
$$

where

$$
\hat{c}_{N-1}(\mathbf{x}_{N-1})
$$
$$
:= \ln\left(\mathbb{E}_{q_{N|N-1}^{(u)}}\left[\exp\left(-\mathcal{D}_{\mathsf{KL}}\left(p_{N|N-1}^{(x)} \| q_{N|N-1}^{(x)}\right)\right.\right.\right.
$$
$$
\left.\left.\left. - \mathbb{E}_{p_{N|N-1}^{(x)}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right)\right]\right).
$$

This is the optimal cost for $k = N$ given in (7). Next, we make use of the minimum found for the sub-problem (8b) to solve the sub-problem corresponding to $k \in 1 : N - 1$.

Step 4. It can be shown that the problem in (8a) can be again split as the sum of two sub-problems: one sub-problem for the time-steps up to $N - 2$ and a sub-problem for $k = N - 1$. Moreover, the latter sub-problem is again independent on the former and has the same structure as (8b). Then, following the arguments used in Step 3, we have that the unique optimal

solution for the sub-problem at $k = N - 1$ is

$$
p_{N-1|N-2}^{(u)}{}^*
$$
$$
= \frac{\bar{p}_{N-1|N-2}^{(u)} \exp\left(-\mathbb{E}_{p_{N-1|N-2}^{(x)}}\left[\bar{c}_{N-1}(\mathbf{X}_{N-1})\right]\right)}{\sum_{\mathbf{u}_{N-1}} \bar{p}_{N-1|N-2}^{(u)} \exp\left(-\mathbb{E}_{p_{N-1|N-2}^{(x)}}\left[\bar{c}_{N-1}(\mathbf{X}_{N-1})\right]\right)}.
$$
$$(14)$$

Hence, (14) is the optimal solution given in (5) for $k = N-1$, with $\bar{c}_{N-1}(\mathbf{x}_{N-1})$ obtained via the backward recursion in (6). We can now draw the desired conclusions.

Step 5. By iterating Step 4 we find that, at each of the remaining time steps in the window $1 : N - 2$, Problem 1 can always be split in sub-problems, where the sub-problem corresponding to the last time instant in the window can be solved independently on the others. Hence, the optimal solution for the sub-problem is

$$
p_{k|k-1}^{(u)}{}^* = \frac{\bar{p}_{k|k-1}^{(u)} \exp\left(-\mathbb{E}_{p_{k|k-1}^{(x)}}\left[\bar{c}_k(\mathbf{X}_k)\right]\right)}{\sum_{\mathbf{u}_k} \bar{p}_{k|k-1}^{(u)} \exp\left(-\mathbb{E}_{p_{k|k-1}^{(x)}}\left[\bar{c}_k(\mathbf{X}_k)\right]\right)}. \quad (15)
$$

This is the optimal solution given in (5) at time k, with $\bar{c}_k(\mathbf{x}_k)$ obtained from the backward recursion in (6). Part (i) of the result is then proved. Moreover, the corresponding optimal cost at time k is $-\mathbb{E}_{\bar{p}_{k-1}}\left[\hat{c}_{k-1}(\mathbf{X}_{k-1})\right]$. Hence, the optimal cost Problem 1 is $-\sum_{k=1}^{N} \mathbb{E}_{\bar{p}_{k-1}}\left[\hat{c}_{k-1}(\mathbf{X}_{k-1})\right]$ and this proves part (ii) of the result. $\square$

Remark 3. *The optimal solution given in Theorem 1 has an exponential twisted kernel. This class of policies is also known in the literature as soft-max/Boltzmann policies and are often assumed in IRL/IOC works, see e.g., [5], [14], [24].*

B. Estimating the Cost

Next, we show that the cost $\bar{c}_k(\cdot)$ can be estimated by observing a sequence of states sampled from $p_{k|k-1}^{(x)}$ when control inputs sampled from $p_{k|k-1}^{(u)}{}^*$ are applied. This can be useful in settings where one has access to observations of e.g., an *expert*. By estimating $\bar{c}_k(\cdot)$ one can also bypass the computation of the cost-to-go via the backward recursion in Theorem 1. With the next result, we propose an estimator for the cost-to-go. The estimator does not require any linearity/stationary assumption and the underlying dynamics can be stochastic and obtained directly from data.

Theorem 2. *Let Assumption 2 hold and let* $\hat{\Delta} = \left\{(\hat{\mathbf{x}}_0, \hat{\mathbf{u}}_1), \ldots, (\hat{\mathbf{x}}_{M-1}, \hat{\mathbf{u}}_M)\right\}$ *be a sequence of data, with* $\hat{\mathbf{x}}_k \sim p_{k|k-1}^{(x)}$, $\hat{\mathbf{u}}_k \sim p_{k|k-1}^{(u)}{}^*$ *and where* $p_{k|k-1}^{(u)}{}^*$ *is from Theorem 1. Then, the maximum likelihood estimate for* $\bar{c}_k(\mathbf{x}_k)$, *say* $\bar{c}_k^*(\mathbf{x}_k)$, *is given by* $\bar{c}_k^*(\mathbf{x}_k) = -\mathbf{w}_k^{*T}\mathbf{h}(\mathbf{x}_k)$, *where* $\mathbf{w}_k^*$ *is obtained by solving the convex optimization*

problem

$$\mathbf{w}^* := \left[\mathbf{w}_1^{*T}, \ldots, \mathbf{w}_M^{*T}\right]^T \in$$

$$\underset{\mathbf{w}}{arg\ min}\Bigg\{ \sum_{k=1}^{M} \Bigg(-\mathbb{E}_{p(\mathbf{x}_k|\hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)}\left[\mathbf{w}_k^T\mathbf{h}(\mathbf{x}_k)\right]$$

$$+ \ln\Bigg(\sum_{\mathbf{u}_k} \bar{q}_{k|k-1}^{(u)}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)$$

$$\exp\left(\mathbb{E}_{p(\mathbf{x}_k|\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)}\left[\mathbf{w}_k^T\mathbf{h}(\mathbf{x}_k)\right]\right)\Bigg)\Bigg)\Bigg\}, \tag{16}$$

and where

$$\bar{q}_{k|k-1}^{(u)}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_k) := (q(\mathbf{u}_k \mid \hat{\mathbf{x}}_{k-1})$$
$$\exp\left(-\mathcal{D}_{KL}\left(p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1},\mathbf{u}_k) \mid\mid q(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1},\mathbf{u}_k))\right)\right), \tag{17}$$

Sketch of the proof. The result is based on maximum likelihood, leveraging the structure of the policy of Theorem 1. Convexity follows from the fact that the feasibility domain is convex and the cost function is a linear combination of the log-sum-exp function and of a linear function (in the decision variables). The proof, omitted here for brevity, will be presented elsewhere. □

Remark 4. *The problem in* (16) *is an unconstrained convex optimization problem with a twice differentiable cost. Constraints on the $\mathbf{w}_k$'s can be added to capture application-specific requirements, such as* dwell-time *constraints.*

Next, we propose an estimator when the cost, which we simply denote by $c(\cdot)$, is stationary. The result (the proof of which is omitted here for brevity) implies that the cost can be estimated from a *greedy* policy obtained via Theorem 1. Note that, in this case, the decision variable in the resulting optimization is $\mathbf{w}_s \in \mathbf{R}^F$ rather than $\mathbf{w} \in \mathbf{R}^{F \times M}$.

Corollary 1. *Let Assumption 2 hold and consider $p_{k|k-1}^{(u)\ *}$ obtained at each k from Theorem 1 with $N = 1$. Further, let the cost be stationary. Then, the maximum likelihood estimate for the cost is $c^*(\mathbf{x}_k) = -\mathbf{w}_s^{*T}\mathbf{h}(\mathbf{x}_k)$, where $\mathbf{w}_s^*$ is given by:*

$$\mathbf{w}_s^* \in \underset{\mathbf{w}_s}{arg\ min}\Bigg\{ \sum_{k=1}^{M} \left(-\mathbb{E}_{p(\mathbf{x}_k|\hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)}\left[\mathbf{w}_s^T\mathbf{h}(\mathbf{x}_k)\right]\right)$$

$$+ \sum_{k=1}^{M} \ln\Bigg(\sum_{\mathbf{u}_k} \bar{q}_{k|k-1}^{(u)}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)$$

$$\exp\left(\mathbb{E}_{p(\mathbf{x}_k|\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)}\left[\mathbf{w}_s^T\mathbf{h}(\mathbf{x}_k)\right]\right)\Bigg)\Bigg\}. \tag{18}$$

with $\mathbf{w}_s \in \mathbf{R}^f$ and $\bar{q}_{k|k-1}^{(u)}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)$ defined in Theorem 2.

Corollary 1 implies that, rather conveniently, the cost can be learned from a greedy policy rather than from the optimal policy. The result can be also turned into an algorithmic procedure with its main steps given in Algorithm 1.

Algorithm 1 Pseudo-code from Corollary 1

Inputs: observed data $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_M$ and $\hat{\mathbf{x}}_0, \ldots, \hat{\mathbf{x}}_M$,
$\quad\quad$ f-dimensional features vector $\mathbf{h}(\mathbf{x}_k)$,
$\quad\quad$ $p_{k|k-1}^{(x)}$, $q_{k|k-1}^{(x)}$, $q_{k|k-1}^{(u)}$
Output: $\bar{c}^*(\mathbf{x}_k)$
for $k = 1$ to M **do**
$\quad$ Compute $\bar{q}_{k|k-1}^{(u)}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)$ using (17)
end for
Compute $\mathbf{w}_s^*$ by solving the problem in (18)

IV. APPLICATION EXAMPLE

We illustrate the effectiveness of our results by considering the problem of stabilizing a pendulum on its unstable equilibrium point. Specifically, given a suitable cost, we first used Theorem 1 to compute the optimal policy and then we leveraged Corollary 1 to estimate the cost used in the policy. The pendulum dynamics (only used to generate data) is:

$$\theta_k = \theta_{k-1} + \omega_{k-1}dt + W_\theta$$
$$\omega_k = \omega_{k-1} + \left(\frac{g}{l}\sin(\theta_{k-1}) + \frac{u_k}{ml^2}\right)dt + W_\omega, \tag{19}$$

where θ_k is the angular position, ω_k is the angular velocity and u_k is the torque applied on the hinged end. The parameter l is the length of the rod, m is the mass of the pendulum, g is the gravity and dt is the discretization step. Also, W_θ and W_ω capture Gaussian noise on the state variables. In our experiments we set $W_\theta \sim \mathcal{N}(0, 0.05)$ and $W_\omega \sim \mathcal{N}(0, 0.1)$. As in [2] we let $\mathbf{X}_k := [\theta_k, \omega_k]^T$. Also, $\mathbf{X}_k \in \mathcal{X}$ and $u_k \in \mathcal{U}$, where the set $\mathcal{X} := [-\pi, \pi] \times [-5, 5]$ and $\mathcal{U} := [-2.5, 2.5]$ are discretised in 50×50 and 20 bins respectively.

The *target* pendulum we wanted to control had parameters $m = 1$kg, $l = 0.6$m, and $dt = 0.1$s. We also considered a different (i.e., *source*) pendulum with parameters $m = 0.5$kg, $l = 0.5$m, and $dt = 0.1$s. We obtained the pmfs $p_{k|k-1}^{(x)}$ and $q_{k|k-1}^{(x)}$ from a database collected following the process from [2], leveraging the source code that was provided therein. We obtained $q_{k|k-1}^{(u)}$ by controlling the source pendulum via Model Predictive Control (MPC) with a receding horizon window width of 20 steps. The action space was $\mathcal{U}$ and the cost function at each k was $\sum_{t \in k:k+H-1}(\theta_k^2 + 0.1\omega_k^2) + \theta_{k+H}^2 + 0.5\omega_{k+H}^2$. Then, as in [2], we added Gaussian noise to the MPC control inputs so that $q_{k|k-1}^{(u)}$ was $\mathcal{N}(\tilde{u}_k, 0.2)$, with $\tilde{u}_k$ being the control input computed via MPC.

Given this set-up, we first computed $p_{k|k-1}^{(u)\ *}$ for the target pendulum using Theorem 1 with $N = 1$ and using as cost:

$$c(\mathbf{x}_k) = (\theta_k - \theta_d)^2 + 0.01(\omega_k - \omega_d)^2, \tag{20}$$

with $\theta_d = 0$ and $\omega_d = 0$ ($\theta_d = 0$ corresponds to the unstable equilibrium). Then, the control input to the target pendulum was obtained by sampling from $p_{k|k-1}^{(u)\ *}$. In Figure 1 the behavior is shown for the angular position of the controlled pendulum and the corresponding control input. The figure clearly illustrates that the unstable equilibrium is stabilized. Next, we illustrated the effectiveness of Algorithm 1 in
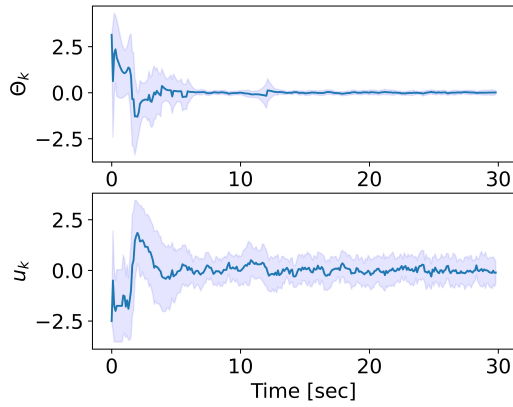
Fig. 1. Target pendulum angular position and corresponding control input. Figure obtained from 20 simulations. Bold lines represents the mean and the shaded region is confidence interval corresponding to the standard deviation.
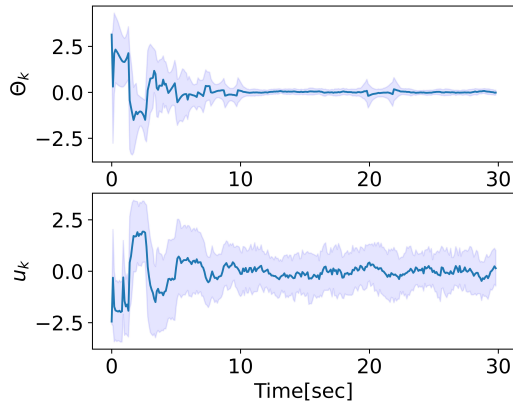

Fig. 2. Angular position and control input for the target pendulum when this is controlled by a policy using the estimated cost. Figure obtained by running 20 simulations. Bold lines and shaded area defined as in Figure 1.

reconstructing the cost given in (20) that was used for policy computation. To this aim, we used a dataset of 300 datapoints collected from a single simulation where the target pendulum was controlled by the policy computed above. We defined the features as $\mathbf{h}(\mathbf{x}_k) = [| \theta_k - \theta_d |, | \omega_k - \omega_d |]^T$. We then obtained from Algorithm 1 the weights $\mathbf{w}_s^* = [-3.3, -2.03]^T$ and hence the estimated cost was $c^*(\mathbf{x}_k) = 3.3 | \theta_k - \theta_d | + 2.03 | \omega_k - \omega_d |$. Note that the weight was higher for θ_k than for ω_k, consistently with the cost in (20). Finally, with this estimated cost, we used Theorem 1 with $N = 1$ to obtain a new policy, which we used on the target pendulum. Simulations (in Figure 2) illustrate that this policy with the estimated cost effectively stabilizes the pendulum.

V. CONCLUSIONS

We considered the problem of estimating the possibly non-convex cost of an agent by observing its interactions with a nonlinear, non-stationary, and stochastic environment. Using probabilistic descriptions from data and/or first-principles, we formulated a convex optimization problem to estimate the cost. To solve the inverse problem, we also formulated a convex finite-horizon optimal control problem and found its optimal solution. The results were turned into an algorithm and illustrated through simulations, with future work focusing on environment learning and constrained control tasks.

REFERENCES

[1] N. Ab Azar, A. Shahmansoorian, and M. Davoudi, "From inverse optimal control to inverse reinforcement learning: A historical review," *Annual Reviews in Control*, vol. 50, pp. 119–138, 2020.

[2] Émiland Garrabé and G. Russo, "Probabilistic design of optimal sequential decision-making algorithms in learning and control," *Annual Reviews in Control*, vol. 54, pp. 81–102, 2022.

[3] E. Crisostomi, B. Ghaddar, F. Hausler, J. Naoum-Sawaya, G. Russo, and R. Shorten, Eds., *Analytics for the Sharing Economy: Mathematics, Engineering and Business Perspectives*. Springer, 2020.

[4] A. E. Bryson, "Optimal control-1950 to 1985," *IEEE Control Systems Magazine*, vol. 16, pp. 26–33, 1996.

[5] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *23rd International conference on Artificial intelligence*, 2008, pp. 1433–1438.

[6] N. Mehr, M. Wang, M. Bhatt, and M. Schwager, "Maximum-entropy multi-agent dynamic games: Forward and inverse solutions," *IEEE Transactions on Robotics*, pp. 1–15, 2023.

[7] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *29th International Conference on Machine Learning*, 2012, p. 475–482.

[8] S. Levine, Z. Popovic, and V. Koltun, "Nonlinear inverse reinforcement learning with Gaussian processes," in *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[9] M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal, "Learning objective functions for manipulation," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1331–1336.

[10] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *33rd International Conference on Machine Learning*, vol. 48, 2016, p. 49–58.

[11] R. Self, M. Abudia, S. N. Mahmud, and R. Kamalapurkar, "Model-based inverse reinforcement learning for deterministic systems," *Automatica*, vol. 140, p. 110242, 2022.

[12] B. Lian, W. Xue, F. L. Lewis, and T. Chai, "Inverse reinforcement learning for multi-player noncooperative apprentice games," *Automatica*, vol. 145, p. 110524, 2022.

[13] K. Dvijotham and E. Todorov, "Inverse optimal control with linearly-solvable MDPs," in *27th International Conference on Machine Learning*, 2010, p. 335–342.

[14] L. J. Ratliff and E. Mazumdar, "Inverse risk-sensitive reinforcement learning," *IEEE Transactions on Automatic Control*, vol. 65, pp. 1256–1263, 2020.

[15] Y. Nakano, "Inverse stochastic optimal controls," *Automatica*, vol. 149, p. 110831, 2023.

[16] L. Rodrigues, "Inverse optimal control with discount factor for continuous and discrete-time control-affine systems and reinforcement learning," in *2022 IEEE 61st Conference on Decision and Control*, 2022, pp. 5783–5788.

[17] S. Kullback and R. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, pp. 79–87, 1951.

[18] V. Peterka, "Bayesian approach to system identification," in *Trends and Progress in System identification*. Elsevier, 1981, pp. 239–304.

[19] E. Todorov, "Linearly-solvable Markov decision problems," in *Advances in Neural Information Processing Systems*, vol. 19, 2007, pp. 1369–1376.

[20] N. Cammardella, A. Bušić, Y. Ji, and S. Meyn, "Kullback-Leibler-Quadratic optimal control of flexible power demand," in *2019 IEEE 58th Conference on Decision and Control*, 2019, pp. 4195–4201.

[21] D. Gagliardi and G. Russo, "On a probabilistic approach to synthesize control policies from example datasets," *Automatica*, vol. 137, p. 110121, 2022.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[23] M. Yin, A. Iannelli, and R. S. Smith, "Maximum likelihood estimation in data-driven modeling and control," *IEEE Transactions on Automatic Control*, vol. 68, pp. 317–328, 2023.

[24] P. Guan, M. Raginsky, and R. M. Willett, "Online Markov Decision Processes with Kullback–Leibler control cost," *IEEE Transactions on Automatic Control*, vol. 59, pp. 1423–1438, 2014.