On Convex Data-Driven Inverse Optimal Control for Nonlinear, Non-stationary and Stochastic Systems [1]

Emiland Garrabe [b], Hozefa Jesawada [a], Carmen Del Vecchio [a], Giovanni Russo [b]

[a] *Department of Engineering, University of Sannio, Benevento, Italy*

[b] *Department of Information and Electrical Engineering and Applied Mathematics, University of Salerno, Italy*

Abstract

This paper is concerned with a finite-horizon inverse control problem, which has the goal of inferring, from observations, the possibly non-convex and non-stationary cost driving the actions of an agent. In this context, we present a result that enables cost estimation by solving an optimization problem that is convex even when the agent cost is not and when the underlying dynamics is nonlinear, non-stationary and stochastic. To obtain this result, we also study a finite-horizon forward control problem that has randomized policies as decision variables. For this problem, we give an explicit expression for the optimal solution. Moreover, we turn our findings into algorithmic procedures and we show the effectiveness of our approach via both in-silico and experimental validations with real hardware. All the experiments confirm the effectiveness of our approach.

1 Introduction

Inverse optimal control/reinforcement learning (IOC/IRL) refer to the problem of inferring the cost driving the actions of an agent from input/output observations Ab Azar et al. (2020). Tackling IOC/IRL problems is crucial to many scientific domains from engineering, psychology, economics, management and computer science. However, a key challenge is that the underlying optimization can become ill-posed even for a linear deterministic dynamics with convex cost. Motivated by this, in a finite-horizon setting, we propose an algorithm enabling cost estimation by solving an optimization problem that is convex even when the agent cost is not and when the underlying dynamics is nonlinear, non-stationary and stochastic. We now briefly survey some works related to the results/framework of this paper. For a comprehensive review on IOC/IRL problems we refer readers to e.g., Ab Azar et al. (2020).

Related Works

As noted in Bryson (1996), IOC methods were originally developed to determine control *functions* that generate observed outputs. More recently, also driven by the advancements in computational power and the increase in the availability of data, there has been a renewed interest in IOC (and IRL) methods. For stationary Markov Decision Processes (MPDs) an approach based on maximum entropy is proposed in Ziebart et al. (2008), with the resulting algorithm based on a backward/forward pass scheme. Additionally, following this research stream, Mehr et al. (2023) considers linear multi-agent games, while Levine and Koltun (2012) obtains results based on the local approximation of the cost/reward. Furthermore, Levine et al. (2011) also proposes a complementary approach based on the use of Gaussian Processes to model stochastic dynamics. The resulting algorithm requires matrix inversions and relies on optimization problems that are not convex in general. For stationary MDPS, Finn et al. (2016) builds on maximum-entropy and uses deep neural networks to estimate the cost. The approach is benchmarked

Email addresses: egarrabe@unisa.it (Emiland Garrabe), jesawada@unisannio.it (Hozefa Jesawada), c.delvecchio@unisannio.it (Carmen Del Vecchio), giovarusso@unisa.it (Giovanni Russo).

[1] An early version of this paper with only a sketch of the proof for one of the results and without the hardware validation has been submitted for presentation at the 62nd IEEE Conference on Decision and Control. The submitted conference work is Garrabe et al. (2023). EG and HJ are joint first authors. Corresponding author: Giovanni Russo.

on manipulation tasks. Also for manipulation tasks, Kalakrishnan et al. (2013) considers deterministic nonlinear systems, utilizing path integrals to learn the cost. In the context of deterministic systems, Self et al. (2022) introduces a model-based IRL algorithm, while Lian et al. (2022) tackles the IRL problem for multiplayer non-cooperative games. Linearly solvable MDPs are exploited in Dvijotham and Todorov (2010) and within this framework one can obtain a convex optimization problem to estimate the cost. However, this approach assumes that the agent can specify directly the state transition. A risk-sensitive IRL algorithm is proposed in Ratliff and Mazumdar (2020) for cost estimates in stationary MDPs assuming that the expert policy belongs to the exponential distribution. In the context of IOC, Nakano (2023) considers stochastic dynamics and proposes an approach to estimate the parameters of a control regularizer for finite-horizon problems. Moreover, Rodrigues (2022) tackles an IOC problem for known nonlinear deterministic systems with quadratic cost function in the input. In Do (2019) infinite-horizon IOC problems are considered for inverse optimal stabilization and inverse optimal gain assignment for stochastic nonlinear systems driven by Lévy processes. In Deng and Krstić (1997), stabilization problems in an infinite-horizon setting are considered and it is shown that for every system with a stochastic control Lyapunov function, one can construct a controller which is optimal with respect to some cost. In Jouini and Rantzer (2022) the cost design problem is considered and combinations of input/states compatible with a given value function are studied. Finally, the approach we propose relies on a technical result to find the optimal solution for a related finite-horizon forward control problem. This problem has randomized policies as decision variables and we refer to Garrabe and Russo (2022) for a survey on this class of sequential decision-making problems across learning and control.

Contributions

We propose an algorithm to tackle the inverse problem. Our algorithm enables cost estimation by solving an optimization problem that is convex even when the agent cost is not convex, non-stationary and when the underlying dynamics is nonlinear, non-stationary and stochastic. To the best of our knowledge, this is the first algorithm that allows to estimate the cost via a convex program in this setting and that, at the same time: (i) does not require that the agent can specify its state transitions; (ii) does not assume that the underlying MDP is stationary; (iii) does not require solving forward problems in each optimization iteration and can be used for non-deterministic dynamics.

More in detail, our main technical contributions can be summarized as follows:

(1) in the finite-horizon setting, we introduce a result to recast the cost estimate problem into a convex optimization problem. We show that convexity is guaranteed even when the task cost is not convex and non-stationary. Moreover, the result, which leverages certain probabilistic descriptions that can be obtained directly from data, does not require assumptions on the underlying dynamics besides the standard Markov assumption;
(2) to obtain our result on the inverse problem, we also tackle a related forward problem. This leads to a finite-horizon optimal control problem with randomized policies as decision variables. We then introduce a theoretical result that gives the explicit expression for the optimal solution, finding that this is a probability function with an exponential twisted kernel. We exploit this structure to develop our arguments on the inverse problem;
(3) the results for the inverse and the forward control problems are turned into algorithmic procedures (the documented code implementing the algorithms is available at `https://tinyurl.com/46uccxsf`);
(4) the algorithms are validated via a hardware test-bed. Namely, we use our algorithms to reconstruct the cost of robots navigating in an environment with the goal of reaching a desired destination while avoiding obstacles. Additionally, the paper includes a running example involving the swing-up of a pendulum. This example, together with the supporting documentation and code at `https://tinyurl.com/46uccxsf`, illustrates the more practical aspects of our results and highlights some of the key algorithmic implementation details.

The paper is organized as follows. We introduce some mathematical preliminaries and the control problem in Section 2. In Section 3, we present our main results for the inverse/forward problems and turn the results into algorithmic procedures. These are used, in Section 4, to tackle an application with a real hardware set-up that involves routing unicycle robots in an environment with obstacles. Concluding remarks are given in Section 5.

2 Mathematical Preliminaries and Problem Formulation

Sets are in *calligraphic* and vectors in **bold**. We let $\mathbb{K}$ be either $\mathbb{R}$ or $\mathbb{Z}$. A random variable is denoted by $\mathbf{V}$ and its realization is $\mathbf{v}$. We denote the *probability mass function* (pmf, for discrete variables) or *probability density function* (pdf, for continuous variables) of $\mathbf{V}$ by $p(\mathbf{v})$ and we let $\mathcal{D}$ be the convex subset of pdfs/pmfs. In what follows, we simply say that $p(\mathbf{v})$ is a probability function (pf). Whenever we take the sums/integrals involving pfs we always

assume that they exist. The expectation of a function $\mathbf{h}(\cdot)$ of a discrete $\mathbf{V}$ is denoted $\mathbb{E}_p[\mathbf{h}(\mathbf{V})] := \sum_{\mathbf{v}} \mathbf{h}(\mathbf{v})p(\mathbf{v})$, 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 (for continuous variables the summation is replaced by the integral on the support of $p(\mathbf{v})$). The joint pf of $\mathbf{V}_1$ and $\mathbf{V}_2$ is denoted by $p(\mathbf{v}_1, \mathbf{v}_2)$ and the conditional pf 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 pf of the form $p(\mathbf{v}_0, \ldots, \mathbf{v}_N)$ is compactly written as $p_{0:N}$ (by definition $p_{k:k} := p_k(\mathbf{v}_k)$). Also, functionals are denoted by capital calligraphic characters with arguments within curly brackets. We make use of the Kullback-Leibler (KL Kullback and Leibler (1951)) divergence, a measure of the proximity of the pair of pmfs $p(\mathbf{v})$ and $q(\mathbf{v})$, defined for discrete variables as $\mathcal{D}_{\mathrm{KL}}(p \,||\, q) := \sum_{\mathbf{v}} p(\mathbf{v}) \ln(p(\mathbf{v})/q(\mathbf{v}))$. (for continuous variables the sum is replaced by the integral). We also recall here the chain rule for the KL-divergence:

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

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

Functionals are denoted by capital calligraphic letters with arguments in curly brackes.

2.1 Problems Set-up

We let $\mathbf{X}_k \in \mathcal{X} \subseteq \mathbb{K}^n$ be the system state at time-step k and $\mathbf{U}_k \in \mathcal{U} \subseteq \mathbb{K}^p$ be the control input at time-step k. The time indexing is chosen so that the system transitions to $\mathbf{x}_k$ when $\mathbf{u}_k$ is applied. That is, by making the standard Markov assumption, the possibly non-stationary, nonlinear stochastic dynamics for the system under control is described by the pf $p^{(x)}_{k|k-1} := p^{(x)}_k(\mathbf{x}_k \mid \mathbf{u}_k, \mathbf{x}_{k-1})$. In what follows, we simply term this pf as target pf.

Remark 1 *In the running example we estimate the target pf from data. We let $\mathbf{\Delta}_k := (\mathbf{x}_{k-1}, \mathbf{u}_k)$ be the input-state data pair collected from the system when this is in state $\mathbf{x}_{k-1}$ and $\mathbf{u}_k$ is applied. Also, $\mathbf{\Delta}_{0:N} := (\{\mathbf{\Delta}_k\}_{1:N}, \mathbf{x}_N)$ be the dataset over the time horizon $\mathcal{T} := 0 : N$. 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.*

In order to formalize the control problems we introduce the pf:

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

where $p^{(u)}_{k|k-1} := p^{(u)}_k(\mathbf{u}_k \mid \mathbf{x}_{k-1})$ is a randomized policy and initial conditions are embedded via the prior $p_0(\mathbf{x}_0)$. Also, we use the shorthand notation $p_{k|k-1} := p^{(x)}_{k|k-1} p^{(u)}_{k|k-1} = p(\mathbf{x}_k, \mathbf{u}_k \mid \mathbf{x}_{k-1})$.

Remark 2 *The pf $p_{0:N}$ describes in probabilistic terms the evolution of closed-loop system when, at each k, a given policy, say $p^{(u)}_k(\mathbf{u}_k \mid \mathbf{x}_{k-1})$, is used. With the forward control problem formalized in Section 2.2 we aim to design the policy. This policy is then exploited to tackle the inverse problem in Section 2.3, where we seek to estimate the cost.*

2.2 The Forward Control Problem

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 $\mathbb{E}_{p^{(x)}_{k|k-1}}[c_k(\mathbf{X}_k)]$. The forward control problem considered in this paper is formalized with the following:

Problem 1 *Given a joint pf $q_{0:N} := q_0(\mathbf{x}_0) \prod_{k=1}^{N} q^{(x)}_k(\mathbf{x}_k \mid \mathbf{u}_k, \mathbf{x}_{k-1}) q^{(u)}_k(\mathbf{u}_k \mid \mathbf{x}_{k-1})$. Find the sequence of pfs,*

$\left\{ {p_{k|k-1}^{(u)}}^{\star} \right\}_{1:N}$, *such that:*

$$\left\{ {p_{k|k-1}^{(u)}}^{\star} \right\}_{1:N} \in \underset{\left\{p_{k|k-1}^{(u)}\right\}_{1:N}}{arg\ min} \left\{ \mathcal{D}_{KL}\left(p_{0:N} \mid\mid q_{0:N}\right) + \sum_{k=1}^{N} \mathbb{E}_{\bar{p}_{k-1:k}} \left[\mathbb{E}_{p_{k|k-1}^{(x)}} \left[c_k(\mathbf{X}_k) \right] \right] \right\} \\ s.t.\ p_{k|k-1}^{(u)} \in \mathcal{D} \quad \forall k \in \mathcal{T}. \tag{2}$$

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

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

Remark 3 *In the cost of Problem 1, 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 reference pf* $q_{0:N}$*.*

Typically, $q_{0:N}$ can be a passive dynamics, see e.g. Todorov (2007); Cammardella et al. (2019), or can be used to capture some desired behavior extracted from e.g., demonstrations as in Gagliardi and Russo (2022). Also, we note the so-called fully probabilistic control problem aimed at making $p_{0:N}$ as similar as possible (in the KL-divergence sense) to the reference pf $q_{0:N}$, see e.g. Kárný (1996); Kárný and Guy (2006) is a special case of Problem 1 when there is no cost. Further, when $p_k^{(x)}(\mathbf{x}_k, \mathbf{u}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) = \pi(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ and $q_k^{(x)}(\mathbf{x}_k, \mathbf{u}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) = \omega(\mathbf{x}_k \mid \mathbf{x}_{k-1})$. Then, Problem 1 becomes the KL-control problem Todorov (2009); Theodorou et al. (2009); Kappen et al. (2012).

For our derivations, it is also useful to introduce $q_{k|k-1}^{(x)} := q_k^{(x)}\left(\mathbf{x}_k \mid \mathbf{u}_k, \mathbf{x}_{k-1}\right)$, $q_{k|k-1}^{(u)} := q_k^{(u)}\left(\mathbf{u}_k \mid \mathbf{x}_{k-1}\right)$ and $q_{k|k-1} := q_{k|k-1}^{(x)} q_{k|k-1}^{(u)} = q\left(\mathbf{x}_k, \mathbf{u}_k \mid \mathbf{x}_{k-1}\right)$. Finally, in what follows, when we want to stress dependency of the cost of Problem 1 on the decision variables, we denote this by $\mathcal{J}\left\{p_{k|k-1}^{(u)}\right\}_{1:N}$.

Assumption 1 *There exists some* $\left\{\tilde{p}_{k|k-1}^{(u)}\right\}_{1:N}$*, with* $\tilde{p}_{k|k-1}^{(u)} := \tilde{p}^{(u)}(\mathbf{u}_k \mid \mathbf{x}_{k-1})$*, that is feasible for Problem 1 and such that* $\mathcal{J}\left\{\tilde{p}_{k|k-1}^{(u)}\right\}_{1:N}$ *is bounded.*

2.3 The Inverse Control Problem

The inverse control problem we consider consists in estimating 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 *The cost-to-go is expressed as a linear combination of features. That is,* $\bar{c}_k\left(\mathbf{x}_k\right) = -{\mathbf{w}_k}^T \mathbf{h}\left(\mathbf{x}_k\right)$*, where* $\mathbf{h}(\mathbf{x}_k) := [h_1(\mathbf{x}_k), \ldots, h_f(\mathbf{x}_k)]^T$ *is the features vector and* $h_i : \mathcal{X} \to \mathbb{R}$ *are known functions,* $i = 1, \ldots, f$*, and* $\mathbf{w}_k := [w_{k,1}, \ldots, w_{k,f}]^T$ *is a vector of weights.*

The assumption is rather common in the literature see e.g., Ziebart et al. (2008); Mehr et al. (2023); Kalakrishnan et al. (2013); Self et al. (2022); Dvijotham and Todorov (2010); Goodfellow et al. (2016). With our results in Section 3.2 we propose a maximum likelihood estimator for $\bar{c}_k(\cdot)$ and $c_k(\cdot)$. See e.g., Yin et al. (2023) for a maximum likelihood framework for linear systems in the context of data-driven control.

Running Example: Pendulum Control

We consider the control of a pendulum and in this first part of the running example we introduce the setting used for our experiments. The forward control problem consists in stabilizing a pendulum on its unstable equilibrium. The cost used to formulate this forward control problem (tackled in the next part of the example) is:

$$c(\mathbf{x}_k) = (\theta_k - \theta_d)^2 + 0.01(\omega_k - \omega_d)^2, \tag{3}$$

with $\theta_d = 0$ and $\omega_d = 0$ ($\theta_d = 0$ corresponds to the unstable equilibrium of the pendulum). With the inverse control problem, tackled in the last part of this example, we estimate the cost from observed states and control inputs.

The pendulum dynamics, which is only used to generate data, is given by:

$$\begin{aligned} \theta_k &= \theta_{k-1} + \omega_{k-1} dt + W_\theta \\ \omega_k &= \omega_{k-1} + \left(\frac{g}{l} \sin(\theta_{k-1}) + \frac{u_k}{ml^2} \right) dt + W_\omega, \end{aligned} \tag{4}$$

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 rod length, m is the mass of the pendulum, g is the gravity and $dt = 0.1$s is the discretization step. Also, W_θ and W_ω are sampled from Gaussians, i.e. $W_\theta \sim \mathcal{N}(0, 0.05)$ and $W_\omega \sim \mathcal{N}(0, 0.1)$. We let $\mathbf{X}_k := [\theta_k, \omega_k]^T$ and $u_k \in \mathcal{U}$ with $\mathcal{U} := [-2.5, 2.5]$. The parameters were chosen as in Garrabe and Russo (2022) so that the target pendulum, i.e. the pendulum that we want to control, had parameters $m = 1$kg, $l = 0.6$m, while the reference pendulum (from which the pfs $q^{(x)}_{k|k-1}$ and $q^{(u)}_{k|k-1}$ are extracted) had $m = 0.5$kg, $l = 0.5$m.

To illustrate the application of our results in both the continuous and the discrete settings, we built both pdfs and pmfs for the target and reference pendulum. These were built directly from data obtained by simulating (4). The process we followed is outlined below (see `https://tinyurl.com/46uccxsf` for the details).

Target pendulum. For the discrete setting, we estimated the empirical pmf, $p^{(x)}_{k|k-1}$, for the target pendulum. To do so, we set $\mathcal{X} := [-\pi, \pi] \times [-5, 5]$ and: (i) built a database of 10000 simulations of 100 time-steps each (at each step of the simulations the control input was sampled from the uniform distribution); (ii) discretized the set $\mathcal{X}$ in 50×50 bins; (iii) used the histogram filter to estimate the empirical pmf from the data. The algorithm for the histogram filter can be found in e.g., Garrabe and Russo (2022) that also provides the related documented code. Instead, for the continuous setting we estimated the pdf $p^{(x)}_{k|k-1}$ via Gaussian Processes (Rasmussen et al., 2006, Chapter 2). To this aim, we: (i) used a database of 30 simulations and 100 time-steps to build a prior (again, at each k the control input was sampled from the uniform distribution); (ii) picked the covariance function as a squared exponential kernel. By doing so, at each k, $p^{(x)}_{k|k-1}$ was a Gaussian with mean and variance inferred from the data.

Reference pendulum. As for the target pendulum, we estimated both a continuous and a discrete $q^{(x)}_{k|k-1}$ and this was done by following the process described for the target pendulum. Instead, $q^{(u)}_{k|k-1}$ was obtained as in Garrabe and Russo (2022) by adding Gaussian noise to a Model Predictive Control (MPC) policy (able to stabilize the unstable equilibrium of the reference pendulum) and subsequently discretizing this pf.

3 Main Results

We now present the main technical results to tackle the inverse and the forward control problems of Section 2. We first present a result tackling the forward control problem by giving the optimal solution of Problem 1. Then, this result is used to tackle the inverse control problem. To streamline the presentation, the results are stated for discrete variables. The proofs for continuous variables follow similar technical derivations and are omitted here for brevity.

3.1 Tackling the Forward Control Problem

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 $\{{p^{(u)}_{k|k-1}}^\star\}_{1:N}$ *, with*

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

where $\bar{p}^{(u)}_{k|k-1} := q^{(u)}_{k|k-1} \exp\left(-\mathcal{D}_{KL}\left(p^{(x)}_{k|k-1} \,||\, q^{(x)}_{k|k-1}\right)\right)$ *and where* $\bar{c}_k : \mathcal{X} \to \mathbb{R}$ *is obtained via the backward recursion*

$$\begin{aligned}
&\bar{c}_k(\mathbf{x}_k) = c_k(\mathbf{x}_k) - \hat{c}_k(\mathbf{x}_k),\\
&\hat{c}_k(\mathbf{x}_k) = \ln\left(\mathbb{E}_{q^{(u)}_{k+1|k}}\left[\exp\left(-\mathcal{D}_{KL}\left(p^{(x)}_{k+1|k} \,||\, q^{(x)}_{k+1|k}\right) - \mathbb{E}_{p^{(x)}_{k+1|k}}\left[\bar{c}_{k+1}(\mathbf{X}_{k+1})\right]\right)\right]\right),\\
&\mathcal{D}_{KL}\left(p^{(x)}_{N+1|N} \,||\, q^{(x)}_{N+1|N}\right) + \mathbb{E}_{p^{(x)}_{N+1|N}}\left[\bar{c}_{N+1}(\mathbf{X}_{N+1})\right] = 0;
\end{aligned} \tag{6}$$

(ii) the corresponding minimum is given by:

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

PROOF. The proof, which is by induction, is organized in steps as in Gagliardi and Russo (2022). In **Step** 1, we decompose Problem 1 into two sub-problems, where the sub-problem corresponding to $k = N$ can be solved independently from the other. Then, in **Step** 2, we show how the sub-problem at $k = N$ is convex with its cost functional being strictly convex and (**Step** 3) we find an explicit solution for the problem. Once we solve this problem, we make use of the minimum we found and show that (**Step** 4) we can split again the problem so that the sub-problem at $k = N-1$ can be solved independently on the other. Moreover, the problem at $k = N-1$ has the same structure as the problem at $k = N$. In **Step** 5, we draw the desired conclusions, noting the consistent structure of sub-problems at each time instant, for $k = 1, \ldots, N-2$.

Step 1**.** By means of Lemma 1 the cost in (2) can be written as $\mathcal{D}_{\mathrm{KL}}\left(p_{0:N-1} \,||\, q_{0:N-1}\right) + \sum_{k=1}^{N-1} \mathbb{E}_{\bar{p}_{k-1:k}}\left[\mathbb{E}_{p^{(x)}_{k|k-1}}\left[c_k(\mathbf{X}_k)\right]\right] + \mathbb{E}_{\bar{p}_{N-1}}\left[\mathcal{D}_{\mathrm{KL}}\left(p_{N|N-1} \,||\, q_{N|N-1}\right) + \mathbb{E}_{p_{N|N-1}}\left[c_N(\mathbf{X}_N)\right]\right]$, which has been obtained by noticing that $\mathcal{D}_{\mathrm{KL}}\left(p_{N|N-1} \,||\, q_{N|N-1}\right)$ only depends on the state at $k = N-1$. Hence, Problem 1 can be recast as the sum of the following two sub-problems:

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

and

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

Note that the sub-problem in (8b) is independent on the sub-problem in (8a) and hence Problem 1 can solved by first solving (8b) and then by taking into account its solution to solve (8a). To this aim, we let $\mathcal{C}_N\left\{p^{(u)}_{N|N-1}\right\} := \mathcal{D}_{\mathrm{KL}}\left(p_{N|N-1} \,||\, q_{N|N-1}\right) + \mathbb{E}_{p_{N|N-1}}\left[c_N(\mathbf{X}_N)\right]$ and note that, in the sub-problem (8b), the decision variable $p^{(u)}_{N|N-1}$ is independent on the pf over which the expectation of $\mathcal{C}_N\left\{p^{(u)}_{N|N-1}\right\}$ is taken, i.e. $\bar{p}_{N-1}$. Hence, the minimum of (8b) is $\mathbb{E}_{\bar{p}_{N-1}}\left[\mathcal{C}_N\left\{p^{(u)}_{N|N-1}{}^{\star}\right\}\right]$, with $\mathcal{C}_N\left\{p^{(u)}_{N|N-1}{}^{\star}\right\}$ being the optimal cost obtained by solving

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

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 constraint in (9) is linear in the decision variable and hence convexity of the problem can be shown by proving that its cost functional is convex. To this aim, we note that the following chain of identities hold for the

cost $\mathcal{C}_N\left\{p^{(u)}_{N|N-1}\right\}$:

$$\begin{aligned}
&\mathcal{C}_N\left\{p^{(u)}_{N|N-1}\right\} := \mathcal{D}_{\mathrm{KL}}\left(p_{N|N-1} \,||\, q_{N|N-1}\right) + \mathbb{E}_{p_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right] \\
&= \sum_{\mathbf{u}_N} p^{(u)}_{N|N-1} \sum_{\mathbf{x}_N} p^{(x)}_{N|N-1} \ln \frac{p^{(x)}_{N|N-1}}{q^{(x)}_{N|N-1}} + \sum_{\mathbf{x}_N} p^{(x)}_{N|N-1} \sum_{\mathbf{u}_N} p^{(u)}_{N|N-1} \ln \frac{p^{(u)}_{N|N-1}}{q^{(u)}_{N|N-1}} + \sum_{\mathbf{x}_N, \mathbf{u}_N} p^{(x)}_{N|N-1} p^{(u)}_{N|N-1} \bar{c}_N(\mathbf{x}_N) \\
&= \mathbb{E}_{p^{(u)}_{N|N-1}}\left[\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N|N-1} \,||\, q^{(x)}_{N|N-1}\right) + \mathbb{E}_{p^{(x)}_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right] + \mathcal{D}_{\mathrm{KL}}\left(p^{(u)}_{N|N-1} \,||\, q^{(u)}_{N|N-1}\right),
\end{aligned}$$

where we used: (i) the definitions of KL-divergence, $p_{N|N-1}$ and $q_{N|N-1}$; (ii) Fubini's theorem; (iii) the fact that $\mathcal{D}_{\mathrm{KL}}\left(p^{(u)}_{N|N-1} \,||\, q^{(u)}_{N|N-1}\right)$ does not depend on $\mathbf{x}_N$ and $\sum_{\mathbf{x}_N} p^{(x)}_{N|N-1} = 1$. Hence, the problem in (9) can be conveniently written as

$$\begin{aligned}
&\min_{p^{(u)}_{N|N-1}} \mathbb{E}_{p^{(u)}_{N|N-1}}\left[\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N|N-1} \,||\, q^{(x)}_{N|N-1}\right) + \mathbb{E}_{p^{(x)}_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right] + \mathcal{D}_{\mathrm{KL}}\left(p^{(u)}_{N|N-1} \,||\, q^{(u)}_{N|N-1}\right) \\
&\quad s.t.\ p^{(u)}_{N|N-1} \in \mathcal{D}.
\end{aligned} \tag{10}$$

We now show that the cost in (10) is strictly convex in the decision variable and we show this by studying its second variation with respect to $p^{(u)}_{N|N-1}$. In fact, the cost can be explicitly written as (recall that the sum is taken over the support of $p^{(u)}_{N|N-1}$)

$$\sum_{\mathbf{u}_N} p^{(u)}_{N|N-1}\left(\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N|N-1} \,||\, q^{(x)}_{N|N-1}\right) + \mathbb{E}_{p^{(x)}_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right] + \ln p^{(u)}_{N|N-1} - \ln q^{(u)}_{N|N-1}\right), \tag{11}$$

and hence its convexity can be studied by studying the second variation (with respect to $p^{(u)}_{N|N-1}$) of the quantity inside the sum in (11). This is equal to $1/p^{(u)}_{N|N-1}$ which is therefore strictly positive in the support of $p^{(u)}_{N|N-1}$. This shows that the problem in (9) is convex, with its cost functional being strictly convex.

Step 3. We now find the solution to the problem in (9) by using the equivalent formulation given in (10). The Lagrangian of the problem in (10) is

$$\begin{aligned}
\mathcal{L}(p^{(u)}_{N|N-1}, \lambda_N) = &\mathbb{E}_{p^{(u)}_{N|N-1}}\left[\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N|N-1} \,||\, q^{(x)}_{N|N-1}\right) + \mathbb{E}_{p^{(x)}_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right] \\
&+ \mathcal{D}_{\mathrm{KL}}\left(p^{(u)}_{N|N-1} \,||\, q^{(u)}_{N|N-1}\right) + \lambda_N\left(\sum_{\mathbf{u}_N} p^{(u)}_{N|N-1} - 1\right),
\end{aligned} \tag{12}$$

where λ_N is the Lagrange multiplier corresponding to the constraint $p^{(u)}_{N|N-1} \in \mathcal{D}$. We find the optimal solution by imposing the first order stationarity conditions on $\mathcal{L}(p^{(u)}_{N|N-1}, \lambda_N)$. We start with imposing the stationarity condition with respect to $p^{(u)}_{N|N-1}$. This yields $\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N|N-1} \,||\, q^{(x)}_{N|N-1}\right) + \mathbb{E}_{p^{(x)}_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right] + \ln \frac{p^{(u)}_{N|N-1}}{q^{(u)}_{N|N-1}} + 1 + \lambda_N = 0$, and hence any candidate solution to the problem in (10) (and hence (9)) is of the form:

$$\tilde{p}^{(u)}_{N|N-1} = \frac{q^{(u)}_{N|N-1} \exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N|N-1} \,||\, q^{(x)}_{N|N-1}\right) - \mathbb{E}_{p^{(x)}_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right)}{\exp\left(1 + \lambda_N\right)}. \tag{13}$$

Now, by imposing the stationarity condition for $\mathcal{L}(p^{(u)}_{N|N-1}, \lambda_N)$ with respect to λ_N, we get $\sum_{\mathbf{u}_N} p^{(u)}_{N|N-1} - 1 = 0$.

Such a condition must hold for the candidate solution and hence we get:

$$\sum_{\mathbf{u}_N} q^{(u)}_{N|N-1} \exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N|N-1} \,||\, q^{(x)}_{N|N-1}\right) - \mathbb{E}_{p^{(x)}_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right) = \exp\left(1+\lambda_N\right). \tag{14}$$

Since the problem in (10) is convex with a strictly convex cost functional, (13) and (14) imply Ben-Tal et al. (1988) that the unique optimal solution for the problem is

$$p^{(u)}_{N|N-1}{}^{\star} = \frac{q^{(u)}_{N|N-1} \exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N|N-1} \,||\, q^{(x)}_{N|N-1}\right) - \mathbb{E}_{p^{(x)}_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right)}{\sum_{\mathbf{u}_N} q^{(u)}_{N|N-1} \exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N|N-1} \,||\, q^{(x)}_{N|N-1}\right) - \mathbb{E}_{p^{(x)}_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right)}. \tag{15}$$

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). Moreover, from (15) and (11), it follows that the minimum of the problem in (10) is:

$$\mathcal{C}_N\left\{p^{(u)}_{N|N-1}{}^{\star}\right\} = -\ln\left(\sum_{\mathbf{u}_N} q^{(u)}_{N|N-1} \exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N|N-1} \,||\, q^{(x)}_{N|N-1}\right) - \mathbb{E}_{p^{(x)}_{N|N-1}}\left[\bar{c}_N(\mathbf{X}_N)\right]\right)\right).$$

Hence, the minimum for the sub-problem in (8b) is

$$\mathbb{E}_{\bar{p}_{N-1}}\left[\mathcal{C}_N\left\{p^{(u)}_{N|N-1}{}^{\star}\right\}\right] = -\mathbb{E}_{\bar{p}_{N-1}}\left[\hat{c}_{N-1}(\mathbf{X}_{N-1})\right], \tag{16}$$

where

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

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**.** Since the problem in (2) has been split as the sum of the sub-problems in (8a) - (8b) and since the solution of (8b) gives the minimum (16), we have that solving Problem 1 is equivalent to solve

$$\begin{aligned} \min_{\{p^{(u)}_{k|k-1}\}_{1:N-1}} \; & \mathcal{D}_{\mathrm{KL}}\left(p_{0:N-1} \,||\, q_{0:N-1}\right) + \sum_{k=1}^{N-1} \mathbb{E}_{\bar{p}_{k-1:k}}\left[\mathbb{E}_{p^{(x)}_{k|k-1}}\left[c_k(\mathbf{X}_k)\right]\right] - \mathbb{E}_{\bar{p}_{N-1}}\left[\hat{c}_{N-1}(\mathbf{X}_{N-1})\right] \\ & s.t.\; p^{(u)}_{k|k-1} \in \mathcal{D} \;\; \forall k \in 1 : N-1. \end{aligned} \tag{17}$$

Now, for the cost of the above problem we have:

$$\begin{aligned} & \mathcal{D}_{\mathrm{KL}}\left(p_{0:N-1} \,||\, q_{0:N-1}\right) + \sum_{k=1}^{N-1} \mathbb{E}_{\bar{p}_{k-1:k}}\left[\mathbb{E}_{p^{(x)}_{k|k-1}}\left[c_k(\mathbf{X}_k)\right]\right] - \mathbb{E}_{\bar{p}_{N-1}}\left[\hat{c}_{N-1}(\mathbf{X}_{N-1})\right] \\ & = \mathcal{D}_{\mathrm{KL}}\left(p_{0:N-2} \,||\, q_{0:N-2}\right) + \sum_{k=1}^{N-2} \mathbb{E}_{\bar{p}_{k-1:k}}\left[\mathbb{E}_{p^{(x)}_{k|k-1}}\left[c_k(\mathbf{X}_k)\right]\right] + \mathbb{E}_{p_{0:N-2}}\left[\mathcal{D}_{\mathrm{KL}}\left(p_{N-1|N-2} \,||\, q_{N-1|N-2}\right)\right] \\ & + \mathbb{E}_{\bar{p}_{N-2:N-1}}\left[\mathbb{E}_{p^{(x)}_{N-1|N-2}}\left[c_{N-1}(\mathbf{X}_{N-1})\right]\right] - \mathbb{E}_{\bar{p}_{N-1}}\left[\hat{c}_{N-1}(\mathbf{X}_{N-1})\right]. \end{aligned} \tag{18}$$

Moreover, for the above expression note that:

(i) $\mathcal{D}_{\mathrm{KL}}\left(p_{N-1|N-2} \,||\, q_{N-1|N-2}\right)$ only depends on $\mathbf{X}_{N-2}$ and hence

$$\mathbb{E}_{p_{0:N-2}}\left[\mathcal{D}_{\mathrm{KL}}\left(p_{N-1|N-2} \,||\, q_{N-1|N-2}\right)\right] = \mathbb{E}_{\bar{p}_{N-2}}\left[\mathcal{D}_{\mathrm{KL}}\left(p_{N-1|N-2} \,||\, q_{N-1|N-2}\right)\right];$$

(ii) $\mathbb{E}_{\bar{p}_{N-2:N-1}}\left[\mathbb{E}_{p^{(x)}_{N-1|N-2}}\left[c_{N-1}(\mathbf{X}_{N-1})\right]\right] = \mathbb{E}_{\bar{p}_{N-2}}\left[\mathbb{E}_{p_{N-1|N-2}}\left[c_{N-1}(\mathbf{X}_{N-1})\right]\right]$

(iii) moreover, $\hat{c}_{N-1}(\cdot)$ only depends on $\mathbf{X}_{N-1}$ and hence $-\mathbb{E}_{\bar{p}_{N-1}}\left[\hat{c}_{N-1}(\mathbf{X}_{N-1})\right] = -\mathbb{E}_{\bar{p}_{N-2}}\left[\mathbb{E}_{p_{N-1|N-2}}\left[\hat{c}_{N-1}(\mathbf{X}_{N-1})\right]\right]$.

This implies that the last three terms in the last equality in (18) can be conveniently written as

$$\mathbb{E}_{\bar{p}_{N-2}}\left[\mathcal{D}_{\mathrm{KL}}\left(p_{N-1|N-2} \,||\, q_{N-1|N-2}\right) + \mathbb{E}_{p_{N-1|N-2}}\left[\bar{c}_{N-1}(\mathbf{X}_{N-1})\right]\right],$$

where $\bar{c}_{N-1}(\mathbf{x}_{N-1}) := c_{N-1}(\mathbf{x}_{N-1}) - \hat{c}_{N-1}(\mathbf{x}_{N-1})$. This corresponds to the recursion (6) at time $k = N-1$. Now, the problem in (17) can be again split, this time as the sum of the following two sub-problems:

$$\begin{aligned} \min_{\{p^{(u)}_{k|k-1}\}_{1:N-2}} \; & \mathcal{D}_{\mathrm{KL}}\left(p_{0:N-2} \,||\, q_{0:N-2}\right) + \sum_{k=1}^{N-2} \mathbb{E}_{\bar{p}_{k-1:k}}\left[\mathbb{E}_{p^{(x)}_{k|k-1}}\left[c_k(\mathbf{X}_k)\right]\right] \\ & s.t.\; p^{(u)}_{k|k-1} \in \mathcal{D} \quad \forall k \in 1:N-2, \end{aligned} \tag{19a}$$

and

$$\begin{aligned} \min_{p^{(u)}_{N-1|N-2}} \; & \mathbb{E}_{\bar{p}_{N-2}}\left[\mathcal{D}_{\mathrm{KL}}\left(p_{N-1|N-2} \,||\, q_{N-1|N-2}\right) + \mathbb{E}_{p_{N-1|N-2}}\left[\bar{c}_{N-1}(\mathbf{X}_{N-1})\right]\right] \\ & s.t.\; p^{(u)}_{N-1|N-2} \in \mathcal{D}. \end{aligned} \tag{19b}$$

Again, the sub-problem in (19b) is independent on the sub-problem in (19a) and its decision variable, i.e. $p^{(u)}_{N-1|N-2}$, is independent on $\bar{p}_{N-2}$. Hence the minimum is $\mathbb{E}_{\bar{p}_{N-2}}\left[\mathcal{C}_{N-1}\left\{{p^{(u)}_{N-1|N-2}}^{\star}\right\}\right]$, with

$$\mathcal{C}_{N-1}\left\{p^{(u)}_{N-1|N-2}\right\} := \mathcal{D}_{\mathrm{KL}}\left(p_{N-1|N-2}| \mid q_{N-1|N-2}\right) + \mathbb{E}_{p_{N-1|N-2}}\left[\bar{c}_{N-1}(\mathbf{X}_{N-1})\right],$$

and ${p^{(u)}_{N-1|N-2}}^{\star}$ being the solution of

$$\begin{aligned} \min_{p^{(u)}_{N-1|N-2}} \; & \mathcal{D}_{\mathrm{KL}}\left(p_{N-1|N-2} \,||\, q_{N-1|N-2}\right) + \mathbb{E}_{p_{N-1|N-2}}\left[\bar{c}_{N-1}(\mathbf{X}_{N-1})\right] \\ & s.t.\; p^{(u)}_{N-1|N-2} \in \mathcal{D}. \end{aligned} \tag{20}$$

This has the same structure as (9) and hence the unique optimal solution for the sub-problem in (20) is

$${p^{(u)}_{N-1|N-2}}^{\star} = \frac{q^{(u)}_{N-1|N-2} \exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N-1|N-2} \,||\, q^{(x)}_{N-1|N-2}\right) - \mathbb{E}_{p^{(x)}_{N-1|N-2}}\left[\bar{c}_{N-1}(\mathbf{X}_{N-1})\right]\right)}{\sum_{\mathbf{u}_{N-1}} \bar{p}^{(u)}_{N-1|N-2} \exp\left(-\mathbb{E}_{p^{(x)}_{N-1|N-2}}\left[\bar{c}_{N-1}(\mathbf{X}_{N-1})\right]\right)}, \tag{21}$$

That is, (21) 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). Moreover, the corresponding cost for (19b) is $\mathbb{E}_{\bar{p}_{N-2}}\left[\mathcal{C}_{N-1}\left\{{p^{(u)}_{N-1|N-2}}^{\star}\right\}\right] = -\mathbb{E}_{\bar{p}_{N-2}}\left[\hat{c}_{N-2}(\mathbf{X}_{N-2})\right]$, where $\hat{c}_{N-2}(\mathbf{x}_{n-2}) = \ln\left(\mathbb{E}_{q^{(u)}_{N-1|N-2}}\left[\exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N-1|N-2} \,||\, q^{(x)}_{N-1|N-2}\right) - \mathbb{E}_{p^{(x)}_{N-1|N-2}}\left[\bar{c}_{N-1}(\mathbf{X}_{N-1})\right]\right)\right]\right)$. This is the optimal cost for $k = N-1$ given in (7). We can now draw the desired conclusions.

Step 5. By iterating Step 4, 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 is given by:

$$\begin{aligned} \min_{p^{(u)}_{k|k-1}} \; & \mathbb{E}_{\bar{p}_{k-1}}\left[\mathcal{D}_{\mathrm{KL}}\left(p_{k|k-1} \,||\, q_{k|k-1}\right) + \mathbb{E}_{p_{k|k-1}}\left[\bar{c}_k(\mathbf{X}_k)\right]\right] \\ & s.t.\; p^{(u)}_{k|k-1} \in \mathcal{D}, \end{aligned} \tag{22}$$

where $\bar{c}_k(\mathbf{x}_k) := c_k(\mathbf{x}_k) - \hat{c}_k(\mathbf{x}_k)$ and

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

This yields the recursion in (6) a time k. Hence, the optimal solution for the sub-problem is

$$p^{(u)}_{k|k-1}{}^{\star} = \frac{q^{(u)}_{k|k-1}\exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{k|k-1} \,||\, q^{(x)}_{k|k-1}\right) - \mathbb{E}_{p^{(x)}_{k|k-1}}\left[\bar{c}_k(\mathbf{X}_k)\right]\right)}{\sum_{\mathbf{u}_k} q^{(u)}_{k|k-1}\exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{k|k-1} \,||\, q^{(x)}_{k|k-1}\right) - \mathbb{E}_{p^{(x)}_{k|k-1}}\left[\bar{c}_k(\mathbf{X}_k)\right]\right)}. \tag{23}$$

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 $-\mathbb{E}_{\bar{p}_{k-1}}\left[\hat{c}_{k-1}(\mathbf{X}_{k-1})\right]$. Thus, 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. □

Remark 4 *The optimal solution in Theorem 1 has an exponential twisted kernel. This structure is exploited to prove our result on the inverse problem. This class of policies, also known as soft-max/Boltzmann policies, are often assumed in IRL/IOC works, see e.g., Ratliff and Mazumdar (2020); Ziebart et al. (2008); Guan et al. (2014).*

3.1.1 Turning Theorem 1 into an Algorithm

Theorem 1 can be turned into an algorithmic procedure with its main steps given in Algorithm 1. The algorithm, which takes as input $\mathcal{T}$, $q_{0:N}$, $p^{(x)}_{k|k-1}$ and $c_k(\cdot)$ and outputs $\left\{p^{(u)}_{k|k-1}{}^{\star}\right\}_{1:N}$, computes $p^{(u)}_{k|k-1}{}^{\star}$ following (5) and (6). Rather conveniently, when $c_k(\cdot)$ is available offline, the first two lines in the for loop of Algorithm 1 can also be computed offline so that at run time only the last line in the loop needs to be executed. The documented version of the code for Algorithm 1 provided at `https://tinyurl.com/46uccxsf` leverages this feature.

Algorithm 1 Pseudo-code from Theorem 1 (computing the optimal policy)

Inputs: $\mathcal{T}$, $q_{0:N}$, $p^{(x)}_{k|k-1}$ and $c_k(\cdot)$

Output: $\left\{p^{(u)}_{k|k-1}{}^{\star}\right\}_{1:N}$

Set: $\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{N+1|N} \,||\, q^{(x)}_{N+1|N}\right) + \mathbb{E}_{p^{(x)}_{N+1|N}}\left[\bar{c}_{N+1}(\mathbf{X}_{N+1})\right] \leftarrow 0$;

for $k = N$ to 1 **do**

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

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

$$p^{(u)}_{k|k-1}{}^{\star} \leftarrow \frac{q^{(u)}_{k|k-1}\exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{k|k-1}||q^{(x)}_{k|k-1}\right) - \mathbb{E}_{p^{(x)}_{k|k-1}}\left[\bar{c}_k(\mathbf{X}_k)\right]\right)}{\sum_{\mathbf{u}_k} q^{(u)}_{k|k-1}\exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{k|k-1}||q^{(x)}_{k|k-1}\right) - \mathbb{E}_{p^{(x)}_{k|k-1}}\left[\bar{c}_k(\mathbf{X}_k)\right]\right)}$$

end for

Remark 5 *For continuous variables, the statement of Theorem 1 remains unchanged with the only difference being in the fact that computing the KL-divergence and expectations requires, in general, integrations (and not summations). As a result, the steps shown in Algorithm 1 remain unchanged when the pfs are continuous (with the only difference being the fact that expectations and KL-divergence are evaluated for continuous random variables).*

Running Example (continue)

We continue the running example by using Theorem 1 (and hence Algorithm 1) to swing-up the target pendulum introduced in the previous part of the example. The cost associated to this control task is given in (3). For reasons that will be clear later, for both the continuous and discrete settings described above, at each k we sampled the

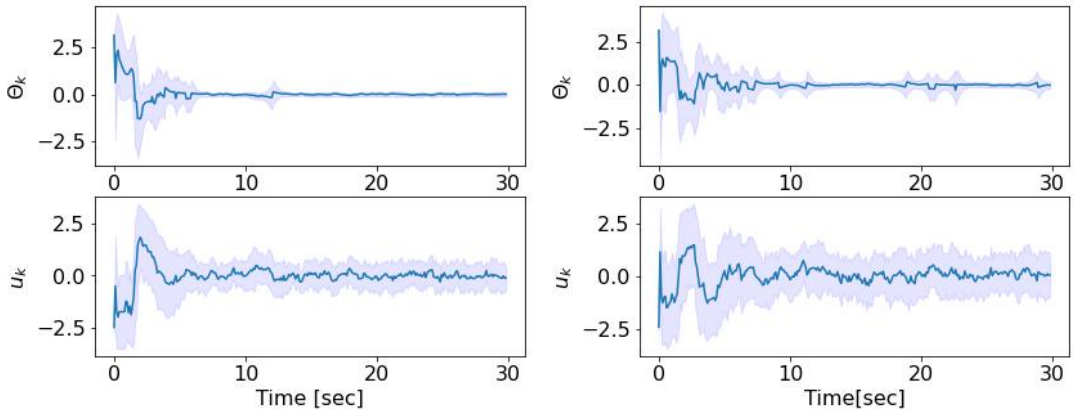

Fig. 1. Target pendulum angular position and corresponding control input. Results obtained when: (i) pfs are discrete and estimated via the histogram filter (left); (ii) pfs are estimated via Gaussian Processes (right). Panels obtained from 20 simulations. Bold lines represent the mean and the shaded region is confidence interval corresponding to the standard deviation.

control input from the policy computed (given $\mathbf{x}_{k-1}$) via Algorithm 1 with $N = 1$. As shown in Figure 1, in both the continuous and discrete settings, the policy ${p^{(u)}_{k|k-1}}^{\star}$ computed via Algorithm 1 stabilized the target pendulum. In the figure, the behavior is shown for the angular position of the controlled pendulum when the control input is sampled, at each k, from ${p^{(u)}_{k|k-1}}^{\star}$. See our github for the implementation details at `https://tinyurl.com/46uccxsf`.

3.2 Tackling the Inverse Control Problem

First, we give a result to estimate, via a convex problem, $\bar{c}_k(\cdot)$ by observing a sequence of states sampled from $p^{(x)}_{k|k-1}$ when the control inputs sampled from ${p^{(u)}_{k|k-1}}^{\star}$ are applied. Then, we build on this result to estimate $c_k(\cdot)$. Observed quantities are denoted with the hat symbol and we write $p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1}, \hat{\mathbf{u}}_k)$, $p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$ and $q(\mathbf{u}_k \mid \hat{\mathbf{x}}_{k-1})$ to denote $p(\mathbf{x}_k \mid \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1}, \mathbf{u}_k = \hat{\mathbf{u}}_k)$, $p(\mathbf{x}_k \mid \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$ and $q(\mathbf{u}_k \mid \mathbf{x}_{k-1} = \hat{\mathbf{x}}_{k-1})$, respectively.

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

$$\mathbf{w}^{\star} := \left[{\mathbf{w}_1^*}^T, \ldots, {\mathbf{w}_M^*}^T\right]^T \in \\ \arg\min_{\mathbf{w}} \left\{ \sum_{k=1}^{M} \left(-\mathbb{E}_{p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1}, \hat{\mathbf{u}}_k)} \left[\mathbf{w}_k^T \mathbf{h}(\mathbf{x}_k)\right] + \ln\left(\sum_{\mathbf{u}_k} \bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \exp\left(\mathbb{E}_{p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)} \left[\mathbf{w}_k^T \mathbf{h}(\mathbf{x}_k)\right]\right) \right) \right) \right\}, \tag{24}$$

and where

$$\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) := q(\mathbf{u}_k \mid \hat{\mathbf{x}}_{k-1}) \exp\left(-\mathcal{D}_{KL}\left(p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \mid\mid q(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)\right)\right). \tag{25}$$

PROOF. Assumption 2, together with the fact that the observed actions are sampled from the policy of Algorithm 1 implies that sequence of control inputs $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_M$ are determined by sampling, at each k, from the pmf

$${p^{(u)}_{k|k-1}}^{\star} = \frac{\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \exp\left(\sum_{\mathbf{x}_k} p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \mathbf{w}_k^T \mathbf{h}(\mathbf{x}_k)\right)}{\sum_{\mathbf{u}_k} \bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \exp\left(\sum_{\mathbf{x}_k} p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \mathbf{w}_k^T \mathbf{h}(\mathbf{x}_k)\right)}. \tag{26}$$

Then, consider the negative log-likelihood

$$
\begin{aligned}
L\left(\mathbf{w}\right) &= -\ln \prod_{k=1}^{M} \frac{\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)\exp\left(\sum_{\mathbf{x}_k} p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)\mathbf{w}_k^T\mathbf{h}(\mathbf{x}_k)\right)}{\sum_{\mathbf{u}_k}\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)\exp\left(\sum_{\mathbf{x}_k} p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1},\mathbf{u}_k)\mathbf{w}_k^T\mathbf{h}(\mathbf{x}_k)\right)} \\
&= \sum_{k=1}^{M}\left(-\ln\left(\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)\right) - \sum_{\mathbf{x}_k} p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)\mathbf{w}_k^T\mathbf{h}(\mathbf{x}_k)\right) \\
&+ \sum_{k=1}^{M}\ln\left(\sum_{\mathbf{u}_k}\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)\exp\left(\sum_{\mathbf{x}_k} p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1},\mathbf{u}_k)\mathbf{w}_k^T\mathbf{h}(\mathbf{x}_k)\right)\right),
\end{aligned} \tag{27}
$$

where $\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)$ is given in (25) and where the notation $L\left(\mathbf{w}\right)$ is used to stress that the log-likelihood is a function only of the weights $\mathbf{w}_k$'s. The estimator is obtained by minimizing the negative log-likelihood and this results in the unconstrained problem:

$$
\min_{\mathbf{w}} L(\mathbf{w}), \tag{28}
$$

where

$$
\begin{aligned}
L(\mathbf{w}) := &\sum_{k=1}^{M}\left(-\ln\left(\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)\right) - \sum_{\mathbf{x}_k} p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)\mathbf{w}_k^T\mathbf{h}(\mathbf{x}_k)\right) \\
&+ \sum_{k=1}^{M}\ln\left(\sum_{\mathbf{u}_k}\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)\exp\left(\sum_{\mathbf{x}_k} p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1},\mathbf{u}_k)\mathbf{w}_k^T\mathbf{h}(\mathbf{x}_k)\right)\right)
\end{aligned}
$$

Moreover, the first term in the cost function does not depend on the decision variable. Hence, such term can be removed from the formulation of the problem and this yields (24). Problem's convexity follows from the cost function being a convex combination of a linear function and the log-sum-exponential function (which is convex). □

Remark 6 *The problem in (24) is an unconstrained convex optimization problem with a twice differentiable cost. Constraints on the* $\mathbf{w}_k$*'s can be added to the problem to capture additional desired properties of the cost. For example, when the weights cannot change arbitrarily fast, a* dwell-time *constraint can be added to the formulation Morato et al. (2020). When the constraint set is convex, the problem can be solved with e.g. interior-point methods.*

Next, we propose an estimator when the cost, which we simply denote by $c(\cdot)$, is stationary. Rather conveniently, the result implies that the cost can be estimated from a *greedy* policy obtained via Theorem 1.

Corollary 1 *Let Assumption 2 hold and consider* ${p^{(u)}_{k|k-1}}^{\star}$ *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* $c^{\star}(\mathbf{x}_k) = -{\mathbf{w}_s^{\star}}^T\mathbf{h}(\mathbf{x}_k)$*, where* $\mathbf{w}_s^{\star}$ *is given by:*

$$
\mathbf{w}_s^{\star} \in \underset{\mathbf{w}_s}{arg\ min}\left\{\sum_{k=1}^{M}\left(-\mathbb{E}_{p(\mathbf{x}_k|\hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)}\left[\mathbf{w}_s^T\mathbf{h}(\mathbf{x}_k)\right]\right) + \sum_{k=1}^{M}\ln\left(\sum_{\mathbf{u}_k}\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)\exp\left(\mathbb{E}_{p(\mathbf{x}_k|\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)}\left[\mathbf{w}_s^T\mathbf{h}(\mathbf{x}_k)\right]\right)\right)\right\}. \tag{29}
$$

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

PROOF. Follows from Theorem 2 after noticing that, when $N = 1$ and the cost is stationary, Theorem 1 yields

$$
{p^{(u)}_{k|k-1}}^{\star} = \frac{q^{(u)}_{k|k-1}\exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{k|k-1} \mid\mid q^{(x)}_{k|k-1}\right) - \mathbb{E}_{p^{(x)}_{k|k-1}}\left[c(\mathbf{X}_k)\right]\right)}{\sum_{\mathbf{u}_k} q^{(u)}_{k|k-1}\exp\left(-\mathcal{D}_{\mathrm{KL}}\left(p^{(x)}_{k|k-1} \mid\mid q^{(x)}_{k|k-1}\right) - \mathbb{E}_{p^{(x)}_{k|k-1}}\left[c(\mathbf{X}_k)\right]\right)}.
$$

□

Following Corollary 1, when the cost is stationary, one needs to solve an optimization problem that has as decision variable the $\mathbf{w}_s \in \mathbf{R}^f$ rather than $\mathbf{w} \in \mathbf{R}^{Mf}$. Also, Corollary 1 only requires that data are collected via a *greedy* policy obtained from Theorem 1 with $N = 1$. From the policy computation viewpoint, this is convenient as it bypasses the need to solve the backward recursion in (6).

3.2.1 Turning Corollary 1 into an Algorithm

We turn Corollary 1 into an algorithmic procedure with its main steps given in Algorithm 2. An algorithm for Theorem 2 can also be obtained and it is omitted here for brevity.

Algorithm 2 Pseudo-code from Corollary 1 (estimating the cost)

Inputs: observed data $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_M$ and $\hat{\mathbf{x}}_0, \ldots, \hat{\mathbf{x}}_M$,
f-dimensional features vector $\mathbf{h}(\mathbf{x}_k)$,
$p^{(x)}_{k|k-1}$, $q^{(x)}_{k|k-1}$, $q^{(u)}_{k|k-1}$
Output: $c^\star(\mathbf{x}_k)$
for $k = 1$ to M **do**
  Compute $\bar{q}^{(u)}_{k|k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$ using (25)
end for
Compute $\mathbf{w}_s^\star$ by solving the problem in (29)
$c^\star(\mathbf{x}_k) \leftarrow -\mathbf{w}_s^{\star T}\mathbf{h}(\mathbf{x}_k)$

3.3 A Special Case

We now discuss a special case of our results for the forward and inverse problems when the reference pfs are uniform (with all pfs having compact supports which we do not highlight in what follows to avoid long detours in the notation). We discuss this special case separately for the forward and inverse problems.

Forward problem. The KL-divergence component in the cost of Problem 1 becomes an entropic regularizer and it can be shown that the optimal policy in (5) becomes

$$p^{(u)\ \star}_{k|k-1} = \frac{\exp\left(-\mathbb{E}_{p^{(x)}_{k|k-1}}\left[\ln\left(p^{(x)}_{k|k-1}\right) + \bar{c}_k(\mathbf{X}_k)\right]\right)}{\sum_{\mathbf{u}_k}\exp\left(-\mathbb{E}_{p^{(x)}_{k|k-1}}\left[\ln\left(p^{(x)}_{k|k-1}\right) + \bar{c}_k(\mathbf{X}_k)\right]\right)}, \tag{30}$$

where $\bar{c}_k(\mathbf{x}_k) = c_k(\mathbf{x}_k) - \hat{c}_k(\mathbf{x}_k)$ and with the backward recursion in (6) becoming

$$\begin{aligned} &\hat{c}_k(\mathbf{x}_k) = \ln\left(\sum_{\mathbf{u}_k}\exp\left(-\mathbb{E}_{p^{(x)}_{k+1|k}}\left[\ln\left(p^{(x)}_{k+1|k}\right) + \bar{c}_{k+1}(\mathbf{X}_{k+1})\right]\right)\right), \\ &\mathbb{E}_{p^{(x)}_{N+1|N}}\left[\ln\left(p^{(x)}_{N+1|N}\right) + \bar{c}_{N+1}(\mathbf{X}_{N+1})\right] = 0. \end{aligned} \tag{31}$$

Inverse problem. Since in the optimal policy is now the one in (30), the convex problem in (24) becomes

$$\begin{aligned} &\mathbf{w}^\star := \left[\mathbf{w}_1^{\star T}, \ldots, \mathbf{w}_M^{\star T}\right]^T \in \\ &\arg\min_{\mathbf{w}}\left\{\sum_{k=1}^{M}\left(-\mathbb{E}_{p(\mathbf{x}_k|\hat{\mathbf{x}}_{k-1},\hat{\mathbf{u}}_k)}\left[\mathbf{w}_k^T\mathbf{h}(\mathbf{x}_k)\right] + \ln\left(\sum_{\mathbf{u}_k}\exp\left(-\mathbb{E}_{p(\mathbf{x}_k|\hat{\mathbf{x}}_{k-1},\mathbf{u}_k)}\left[\ln(p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)) + \mathbf{w}_k^T\mathbf{h}(\mathbf{x}_k)\right]\right)\right)\right)\right\}. \end{aligned} \tag{32}$$

Remark 7 *In the special case useful for applications where $p^{(x)}_{k|k-1}$ is a Gaussian pdf, then in* (32) *one can use the explicit expression for the entropic term $-\mathbb{E}_{p^{(x)}_{k+1|k}}[\ln(p^{(x)}_{k+1|k})]$.*

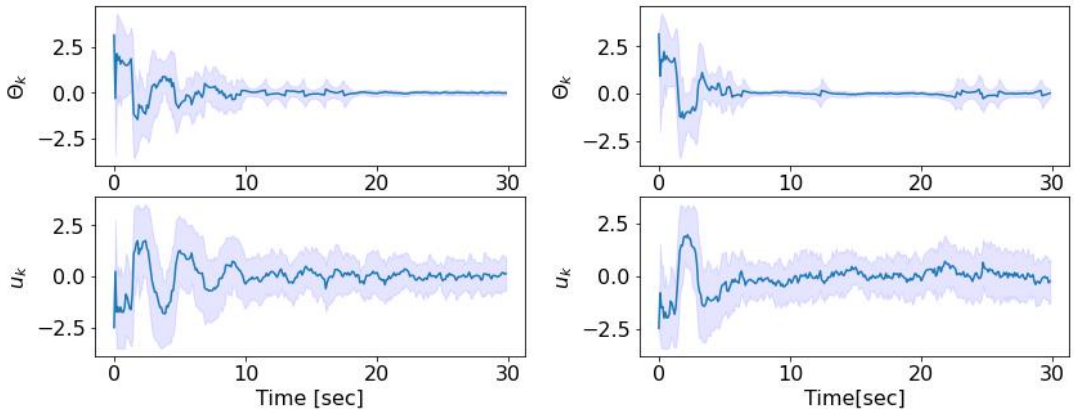

Fig. 2. Angular position and corresponding control input of the target Pendulum when the pf is estimated via the histogram filter (left panels) and Gaussian Processes (right panels). Figures obtained from 20 simulations, using the estimated cost. Bold lines represents the mean and the shaded region is confidence interval corresponding to the standard deviation.

Running Example (continue)

In this final part of the running example we show the application of Algorithm 2 when this is leveraged to reconstruct the cost used to tackle the forward control problem, i.e. the cost in (3) that was given as an input to Algorithm 1. Again, following the previous parts of the example, the continuous and discrete settings are discussed separately. See `https://tinyurl.com/46uccxsf` for the implementation details.

Discrete setting. We used approximately 300 data-points collected from a single simulation of the target pendulum controlled via the policy computed in the previous part of the running example (from this dataset, we removed the points where there were discontinuities due to angle wrapping). These were the observed data given as an input to Algorithm 2. Also, we defined the features as $\mathbf{h}(\mathbf{x}_k) = [|\theta_k - \theta_d|, |\omega_k - \omega_d|]^T$ and obtained from Algorithm 2 (using CVX Diamond and Boyd (2016) to solve the optimization problem) the weights $\mathbf{w}_s^\star = [-6.09, -4.48]^T$. Hence, in accordance with Corollary 1, the estimated cost was $c^\star(\mathbf{x}_k) = 6.09|\theta_k - \theta_d| + 4.48|\omega_k - \omega_d|$. To validate the result, we used this estimated cost as input for Algorithm 1 to solve the forward control problem. As shown in Figure 2 (left panel) the policy computed with the estimated cost effectively stabilizes the pendulum.

Continuous setting. Again, we used a dataset of 300 data-points collected from a single simulation where the target pendulum was controlled by the policy from Algorithm 1. This time we used as features vector $\mathbf{h}(\mathbf{x}_k) = \left[(\cos(\theta_k) - \cos(0.0))^2, (\cos(\theta_k) - \cos(\pi))^2\right]^T$. When this vector was given as an input to Algorithm 2 (using again CVX to solve the problem) the weights we obtained were $\mathbf{w}_s^* = [-13.66, 8.50]^T$ so that $c^\star(\mathbf{x}_k) = 13.66(\cos(\theta_k) - \cos(0))^2 - 8.50(\cos(\theta_k) - \cos(\pi))^2$. Note that the first term in the estimated cost conveniently drives the pendulum towards the unstable equilibrium while the second term is pushing it away from the stable equilibrium of π or $-\pi$, consistently with the cost in (3). Again, we then gave this estimated cost as an input to Algorithm 1 and verified that it was in fact able to swing-up the pendulum. The results are given in Figure 2 (right panel).

4 Application Example

We use our results to tackle an application involving routing unicycle robots in an environment with obstacles. We first use Algorithm 1 to compute a policy routing the robot to reach a desired destination while avoiding obstacles. Then, we leverage Algorithm 2 to estimate the navigation cost from observed robot trajectories. The results are validated both via simulations and via real hardware experiments, leveraging the *Robotarium* platform that offers both a hardware infrastructure and a high-fidelity simulator Wilson et al. (2020). The robots available in the Robotarium are unicycles of dimensions 11cm $\times$ 8.5cm $\times$ 7.5cm (width, length, height) and these can move within a rectangular, *work*, area of 3m $\times$ 2m. The platform is equipped with cameras with a top view to track the robots' positions. Further, the Robotarium allows users to consider for control design a single integrator dynamics rather than the unicycle dynamics (the platform provides support functions to map single integrator dynamics to the unicycle dynamics). Hence, the dynamics of the robot we want to control is $\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{u}_k dt$, where $\mathbf{x}_k = [p_{x,k}, p_{y,k}]^T$ is the position of the robot at time-step k, $\mathbf{u}_k = [v_{x,k}, v_{y,k}]^T$ is the input vector of velocities (in the platform, each of the components are constrained to have modulus less than 0.5m/s) and $dt = 0.033s$ is the Robotarium time-step.

Given this set-up, for the application of our results we set $\mathbf{x}_k \in \mathcal{X} := [-1.5, 1.5] \times [-1, 1]$ and $\mathbf{u}_k \in \mathcal{U} := [-0.5, 0.5] \times [-0.5, 0.5]$. Moreover, in our experiments we also emulated measurement noise for the robot position that we assumed to be Gaussian as in e.g., Kong et al. (2021); Yoo et al. (2016). Thus, $p^{(x)}_{k|k-1} \sim \mathcal{N}(\mathbf{x}_{k-1} + \mathbf{u}_k dt, \Sigma)$, where we set

$$\mathbf{\Sigma} = \begin{bmatrix} 0.001 & 0.0002 \\ 0.0002 & 0.001 \end{bmatrix}.$$

Also, we decided not to build a target pf for this application and hence we set $q^{(x)}_{k|k-1}$ and $q^{(u)}_{k|k-1}$ to be uniform distributions. Further, we discretized $\mathcal{U}$ in a 5×5 grid. For the forward control problem, we used the cost

$$c(\mathbf{x}_k) = 30(\mathbf{x}_k - \mathbf{x}_d)^2 + 20 \sum_{i=1}^{n} g_i(\mathbf{x}_k), \tag{33}$$

where: (i) $\mathbf{x}_d$ is the desired goal/destination for the robot so that the first term in the cost promotes reaching the goal; (ii) n is the number of obstacles; (ii) the terms in the sum promote obstacle avoidance. Specifically, the g_i's would increase as the robot gets closer to the obstacle. Specifically, to this aim we picked the g_i's as Gaussians, i.e.

$$g_i(\mathbf{x}_k) := \frac{1}{\sqrt{(2\pi)^2 \det(\mathbf{\Sigma}_o)}} \exp\left(-\frac{1}{2} (\mathbf{x}_k - \mathbf{o}_i)^\top \mathbf{\Sigma}_o^{-1} (\mathbf{x}_k - \mathbf{o}_i) \right), \tag{34}$$

with $\mathbf{o}_i$ being the coordinates of the barycenter of the i-th obstacle and $\mathbf{\Sigma}_o$ being the co-variance matrix. See our github for the specific values of these parameters. A plot of the cost function is given in Figure 3 (top-left panel).

Given this set-up, we used Algorithm 1 to control the robot. In our implementation, available at our github: (i) the integrals required to compute the expectations in the algorithm were estimated via Monte Carlo sampling; (ii) since $p^{(x)}_{k|k-1}$ is Gaussian, we used the analytic expression of the entropy in the algorithm rather than numerically estimating the related integrals (see Remark 7). We ran 4 experiments from different initial positions for the robot and, for each of the experiments we recorded the robot trajectories. In all the experiments, as shown in the top-right panel of Figure 3, the algorithm was able to properly route the robot to the destination while avoiding the obstacles. A video recording from one of the experiments is also given on our github (see `https://tinyurl.com/46uccxsf`).

Next, we leveraged Algorithm 2 to estimate the cost using the data collected from the above experiments. To do so, we used a 16-dimensional features vector (i.e., $f = 16$), with the first h_i being equal to $(\mathbf{x}_k - \mathbf{x}_d)^2$ and with the other h_i's being Gaussians of the form (34) but centered around 15 uniformly distributed points in the Robotarium work area. Again, we used CVX to solve the underlying optimization problem and, conveniently, this returned a vector of weights that would assign high values in magnitude to weights that corresponded to the obstacles and to the final destination. See the github for the specific expressions of the features and for the values we obtained for the weights. Finally, in order to investigate the effectiveness of the estimated cost, we gave this as an input to Algorithm 1 with the aim of verifying if the robot would still be able to achieve the destination while avoiding obstacles. Specifically, we carried on these experiments by letting the robots start in positions that were different from the ones contained in the dataset used to estimate the cost. The results in Figure 3 confirm successful routing of the robot and obstacle avoidance. See `https://tinyurl.com/46uccxsf` for a recording of an experiment with the estimated cost.

5 Conclusions

We considered the problem of inferring the possibly non-convex and non-stationary cost driving the actions of an agent from observations of its interactions with a nonlinear, non-stationary, and stochastic environment. By leveraging probabilistic descriptions that can be derived both directly from the data and from first-principles, we presented a result to tackle the inverse problem by solving a convex optimization problem. To obtain this result we also tackled a forward control problem with randomized policies as decision variables. For this forward problem, we found an explicit expression for the optimal solution, showing that this has to be a probability function with exponential twisted kernel. The results were also turned into algorithmic procedures with the code made openly

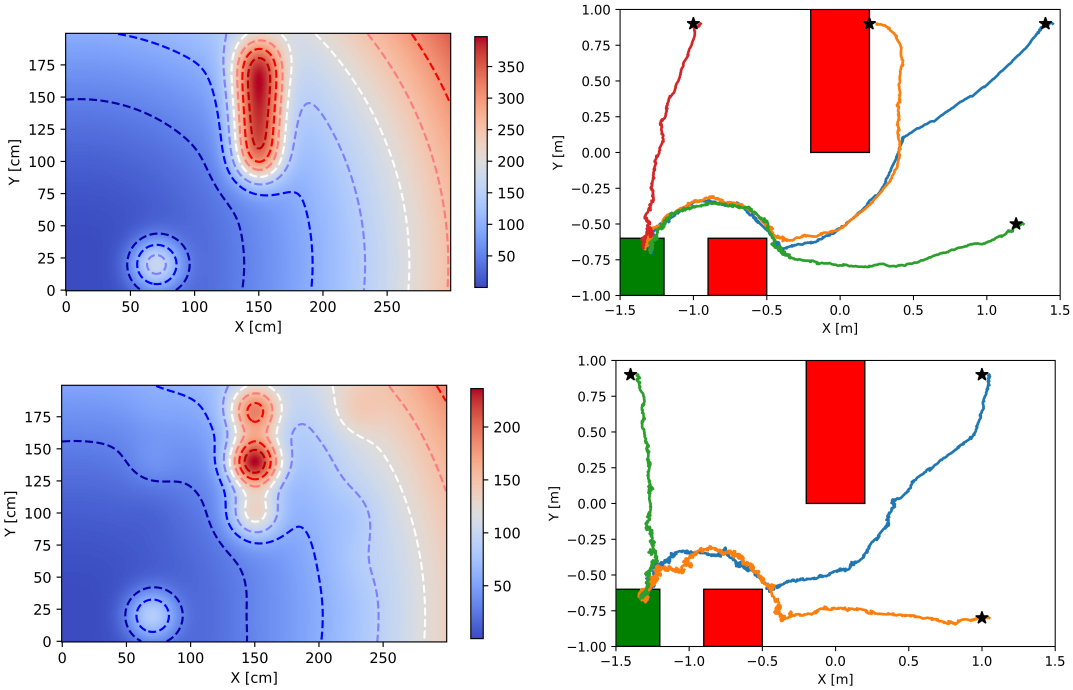

Fig. 3. Top-left: the cost used for the forward control problem. Note that the cost increases in correspondence of the obstacles (in red in the right panels) and its minimum is attained at the goal destination (in green in the right panels). Top-right: robot trajectories starting from different initial conditions depicted as ($\star$) when the policy from Algorithm 1 is used. The control input at each k is sampled from the algorithm's computed policy. Bottom panels: estimated cost plot (left) and robot trajectories under Algorithm 1 with the estimated cost and various initial conditions.

available. The effectiveness of our algorithms was experimentally evaluated via in-silico and hardware validations. As part of our future work, we aim to relax the assumption that the cost function is linearly parametrized in a predefined set of features. To this aim we intend to explore alternative parametrizations, such as adaptive features, to investigate whether they can lead to other convex optimization problems for cost learning. Additionally, we plan to use our results in designing control systems with human-in-the-loop and incentive schemes in sharing economy settings Crisostomi et al. (2020). Finally, we are also interested in exploring the use of our results for multi-agents systems performing repetitive tasks Nair et al. (2022); Bertsekas (2021) and in an online context Guan et al. (2014).

Acknowledgments

EG and GR would like to thank Dr. Guy and Prof. Kárný (Institute of Information Theory and Automation at the Czech Academy of Sciences) for the insightful discussions on a preliminary version of the results presented here.

References

Ab Azar, N., Shahmansoorian, A., Davoudi, M., 2020. From inverse optimal control to inverse reinforcement learning: A historical review. Annual Reviews in Control 50, 119–138.

Ben-Tal, A., Teboulle, M., Charnes, A., 1988. The role of duality in optimization problems involvng entropy functionals with applications to information theory. Journal of optimization theory and applications 58, 209–223.

Bertsekas, D., 2021. Multiagent reinforcement learning: Rollout and policy iteration. IEEE/CAA Journal of Automatica Sinica 8 (2), 249–272.

Bryson, A. E., 1996. Optimal control-1950 to 1985. IEEE Control Systems Magazine 16, 26–33.

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

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

Deng, H., Krstić, M., 1997. Stochastic nonlinear stabilization — ii: Inverse optimality. Systems & Control Letters 32 (3), 151–159.
URL https://www.sciencedirect.com/science/article/pii/S0167691197000674
Diamond, S., Boyd, S., 2016. Cvxpy: A python-embedded modeling language for convex optimization. The Journal of Machine Learning Research 17 (1), 2909–2913.
Do, K., 2019. Inverse optimal control of stochastic systems driven by Lévy processes. Automatica 107, 539–550.
URL https://www.sciencedirect.com/science/article/pii/S0005109819303097
Dvijotham, K., Todorov, E., 2010. Inverse optimal control with linearly-solvable MDPs. In: 27th International Conference on Machine Learning. p. 335–342.
Finn, C., Levine, S., Abbeel, P., 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In: 33rd International Conference on Machine Learning. Vol. 48. p. 49–58.
Gagliardi, D., Russo, G., 2022. On a probabilistic approach to synthesize control policies from example datasets. Automatica 137, 110121.
Garrabe, E., Jesawada, H., Del Vecchio, C., Russo, G., 2023. Inverse data-driven optimal control for nonlinear stochastic non-stationary systems. Submitted to the 62nd IEEE Conference on Decision and Control, CDC 2023.
Garrabe, E., Russo, G., 2022. Probabilistic design of optimal sequential decision-making algorithms in learning and control. Annual Reviews in Control 54, 81–102.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.
Guan, P., Raginsky, M., Willett, R. M., 2014. Online Markov Decision Processes with Kullback–Leibler control cost. IEEE Transactions on Automatic Control 59, 1423–1438.
Jouini, T., Rantzer, A., 2022. On cost design in applications of optimal control. IEEE Control Systems Letters 6, 452–457.
Kalakrishnan, M., Pastor, P., Righetti, L., Schaal, S., 2013. Learning objective functions for manipulation. In: 2013 IEEE International Conference on Robotics and Automation. pp. 1331–1336.
Kappen, H. J., Gómez, V., Opper, M., Feb 2012. Optimal control as a graphical model inference problem. Machine Learning 87 (2), 159–182.
Kárný, M., 1996. Towards fully probabilistic control design. Automatica 32 (12), 1719–1722.
Kárný, M., Guy, T. V., 2006. Fully probabilistic control design. Systems & Control Letters 55 (4), 259–265.
Kong, H., Shan, M., Sukkarieh, S., Chen, T., Zheng, W. X., 2021. Kalman filtering under unknown inputs and norm constraints. Automatica 133, 109871.
Kullback, S., Leibler, R., 1951. On information and sufficiency. Annals of Mathematical Statistics 22, 79–87.
Levine, S., Koltun, V., 2012. Continuous inverse optimal control with locally optimal examples. In: 29th International Conference on Machine Learning. p. 475–482.
Levine, S., Popovic, Z., Koltun, V., 2011. Nonlinear inverse reinforcement learning with Gaussian processes. In: Advances in Neural Information Processing Systems. Vol. 24.
Lian, B., Xue, W., Lewis, F. L., Chai, T., 2022. Inverse reinforcement learning for multi-player noncooperative apprentice games. Automatica 145, 110524.
Mehr, N., Wang, M., Bhatt, M., Schwager, M., 2023. Maximum-entropy multi-agent dynamic games: Forward and inverse solutions. IEEE Transactions on Robotics, 1–15.
Morato, M. M., Bastos, S. B., Cajueiro, D. O., Normey-Rico, J. E., 2020. An optimal predictive control strategy for covid-19 (sars-cov-2) social distancing policies in brazil. Annual Reviews in Control.
Nair, S. H., Tseng, E. H., Borrelli, F., 2022. Collision avoidance for dynamic obstacles with uncertain predictions using model predictive control. In: 2022 IEEE 61st Conference on Decision and Control (CDC). pp. 5267–5272.
Nakano, Y., 2023. Inverse stochastic optimal controls. Automatica 149, 110831.
Rasmussen, C. E., Williams, C. K., et al., 2006. Gaussian processes for machine learning. Vol. 1. Springer.
Ratliff, L. J., Mazumdar, E., 2020. Inverse risk-sensitive reinforcement learning. IEEE Transactions on Automatic Control 65, 1256–1263.
Rodrigues, L., 2022. 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. pp. 5783–5788.
Self, R., Abudia, M., Mahmud, S. N., Kamalapurkar, R., 2022. Model-based inverse reinforcement learning for deterministic systems. Automatica 140, 110242.
Theodorou, E., Buchli, J., Schaal, S., 2009. Path integral-based stochastic optimal control for rigid body dynamics. In: 2009 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning. pp. 219–225.
Todorov, E., 2007. Linearly-solvable Markov decision problems. In: Advances in Neural Information Processing Systems. Vol. 19. pp. 1369–1376.
Todorov, E., 2009. Efficient computation of optimal actions. Proceedings of the National Academy of Sciences 106 (28), 11478–11483.
Wilson, S., Glotfelter, P., Wang, L., Mayya, S., Notomista, G., Mote, M., Egerstedt, M., 2020. The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot

systems. IEEE Control Systems Magazine 40 (1), 26–44.
Yin, M., Iannelli, A., Smith, R. S., 2023. Maximum likelihood estimation in data-driven modeling and control. IEEE Transactions on Automatic Control 68, 317–328.
Yoo, J., Kim, H. J., Johansson, K. H., 2016. Mapless indoor localization by trajectory learning from a crowd. In: 2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN). pp. 1–7.
Ziebart, B. D., Maas, A., Bagnell, J. A., Dey, A. K., 2008. Maximum entropy inverse reinforcement learning. In: 23rd International conference on Artificial intelligence. pp. 1433–1438.