

Inverse Reinforcement Learning: A Control Lyapunov Approach

Samuel Tesfazgi, Armin Lederer and Sandra Hirche

Abstract—Inferring the intent of an intelligent agent from demonstrations and subsequently predicting its behavior, is a critical task in many collaborative settings. A common approach to solve this problem is the framework of inverse reinforcement learning (IRL), where the observed agent, e.g., a human demonstrator, is assumed to behave according to an intrinsic cost function that reflects its intent and informs its control actions. In this work, we reformulate the IRL inference problem to learning control Lyapunov functions (CLF) from demonstrations by exploiting the inverse optimality property, which states that every CLF is also a meaningful value function. Moreover, the derived CLF formulation directly guarantees stability of inferred control policies. We show the flexibility of our proposed method by learning from goal-directed movement demonstrations in a continuous environment.

I. INTRODUCTION

Autonomous systems are increasingly deployed in close proximity and conjunction with humans, whether it is in healthcare robotics, semi-autonomous driving or manufacturing facilities. In many of these applications it is essential that the automation is able to effectively collaborate with the human partner. To this end, a model of the human’s behavior is needed. This model can be used not only to predict the human partner, but also to teach machines complex behaviors, for which the design of a controller is cumbersome. This concept is called imitation learning or programming by demonstrations. Approaches based on this paradigm are advantageous, since preferences can be encoded through the demonstrations and the emulated behavior is easily interpretable [1].

However, imitation learning is in general more difficult than just replicating the demonstrated behavior exactly, since the inferred control policy should ideally generalize well to unknown environments. Therefore, Abbeel and Ng [2] propose an indirect imitation learning approach called apprenticeship learning. Here, the agent is modelled to behave optimally with regards to an intrinsic cost function, which can be used to retrieve the agent’s control policy. Hence, the imitation learning problem is reformulated to inferring the agent’s cost function from demonstrations. This inference problem can be solved using inverse reinforcement learning (IRL), which is sometimes also referred to as inverse optimal control (IOC).¹ In [3], Ziebert et al. expand the IRL framework by introducing the notion of suboptimality in observed demonstrations through probabilistic models. More recently, in [4], neural network parameterizations are used to learn high dimensional nonlinear cost functions. However,

All authors are with the Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany [samuel.tesfazgi, armin.lederer, hirche]@tum.de

¹We consider IRL and IOC to be synonymous and use IRL subsequently.

the deployment of expressive function approximators has the drawback that it introduces model complexity to an already ill-posed inference problem. In addition, most state-of-the-art IRL methods need to continuously solve the forward optimal control problem in order to evaluate the generated trajectories under the current parametrization of the cost function, which is computationally expensive in general. Furthermore, no guarantees regarding the convergence behavior of inferred control policies are provided.

An alternative way of approaching the imitation learning problem are dynamical movement primitives (DMP). Here, the provided demonstrations of a goal-directed task are used to learn a dynamical system capturing the observed behavior through its attractor landscape [5]. For instance, asymptotically stable dynamical system models for describing robot reaching tasks are learned from demonstrations in [6]. DMP methods generally allow to make strong statements regarding the convergence properties of inferred models, due to their grounding in dynamical systems theory. Nevertheless, they lag behind the representational richness of IRL approaches. Since in DMP the agent and the system it is acting on are described as one dynamical system, no explicit model of the agent is retrieved and only pure imitation learning is possible.

In this work we propose a novel approach that uses the property of inverse optimality, which asserts that every Lyapunov function is a value function for some meaningful cost, to reformulate the IRL inference problem to learning CLFs from demonstrations. By utilizing tools of stochastic dynamical system theory we are able to provide formal guarantees for the stability of inferred control policies. Additionally, we propose non-parametric regression methods to learn the CLFs, therefore, resulting in flexible and data-driven human behavior models. The remainder of the paper is structured as follows: First the problem statement is introduced in Section II, while the reformulation of the IRL problem is presented in Section III. Finally, the evaluation of the proposed method follows in Section IV.

II. PROBLEM FORMULATION

Consider a discrete-time, control-affine system driven by the nonlinear dynamics²

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t) + \mathbf{g}(\mathbf{x}_t)\mathbf{u}_t, \quad (1)$$

with continuous states $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$, where \mathcal{X} is a compact set, and known $\mathbf{f}: \mathcal{X} \rightarrow \mathcal{X}$, $\mathbf{g}: \mathcal{X} \rightarrow \mathbb{R}^{n \times m}$ and initial

²Notation: Lower/upper case bold symbols denote vectors/matrices, $\mathbb{R}_+/\mathbb{N}_+$ all real/integer positive numbers, \mathbf{I}_n the $n \times n$ identity matrix, $\|\cdot\|$ the Euclidean norm, and $\mathcal{C}(\mathcal{X})$ the set of continuous functions over a compact set \mathcal{X} . $\sqrt{\cdot}$ applied to a matrix means it is applied element-wise.

condition $\mathbf{x}_0 \in \mathcal{X}$. The agent acting on the system is assumed to perform continuous control actions $\mathbf{u} \in \mathcal{U}(\mathbf{x}) \subseteq \mathbb{R}^m$, such that $\mathcal{U}(\mathbf{x}) = \{\mathbf{u} \in \mathbb{R}^m : \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \in \mathcal{X}\}$. Since the agent performing the task is an expert, it is reasonable to assume that trajectories generated by the agent are bounded. Furthermore, (1) is not a particular restrictive system class, since this structure holds for many mechanical systems.

Following the optimality principle, the agent chooses its control actions \mathbf{u}_t according to a cost function with the goal of minimizing the accumulated stage costs $l: \mathcal{X} \times \mathcal{U} \rightarrow \mathbb{R}$ over time. Thereby, the agent employs the optimal policy

$$\begin{aligned} \boldsymbol{\pi}^*(\mathbf{x}) &= \operatorname{argmin}_{\boldsymbol{\pi}: \mathcal{X} \rightarrow \mathcal{U}} \sum_{t=0}^{\infty} \gamma^t l(\mathbf{x}_t, \boldsymbol{\pi}(\mathbf{x}_t)), & \mathbf{x}_0 = \mathbf{x} & \quad (2a) \\ &\text{such that } \mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t) + \mathbf{g}(\mathbf{x}_t)\boldsymbol{\pi}(\mathbf{x}_t), & & \quad (2b) \end{aligned}$$

where $\gamma \in \mathbb{R}_+$, $\gamma < 1$ is a discount factor.

While optimal control describes the problem of retrieving the optimal policy $\boldsymbol{\pi}^*$ given the known stage cost l , IRL considers the inverse problem. Here, an expert agent demonstrates state-action trajectories $\{(\mathbf{x}_t, \mathbf{u}_t)\}_{t=0}^T$, with $T \in \mathbb{N}_+$. Subsequently, IRL infers the unknown cost function under which the associated optimal policy best explains the observations. However, since the behavior of human agents does not exactly reflect the deterministic optimal policy in practice, e.g., due to motor noise, the unaccounted variations in observed control actions are typically modelled as random perturbations [3]. We consider perturbed policies of the form

$$\mathbf{u}_t = \tilde{\boldsymbol{\pi}}(\mathbf{x}_t) := \boldsymbol{\pi}^*(\mathbf{x}_t) + \sqrt{\boldsymbol{\Sigma}(\mathbf{x}_t)}\boldsymbol{\omega}_t, \quad (3)$$

where the perturbations are described by the independent and identically distributed random variable $\boldsymbol{\omega}$ and state-dependent covariance $\boldsymbol{\Sigma}(\mathbf{x}) \in \mathbb{R}^{m \times m}$. The perturbations $\boldsymbol{\omega}$ are generated according to a zero-mean truncated Normal distribution, where the truncation is chosen such that $\tilde{\boldsymbol{\pi}}(\mathbf{x}) = \mathbf{u} \in \mathcal{U}(\mathbf{x})$ remains true. Thereby, the closed-loop dynamics

$$\mathbf{x}_{t+1} = \tilde{\mathbf{f}}(\mathbf{x}_t) + \mathbf{g}(\mathbf{x}_t)\sqrt{\boldsymbol{\Sigma}(\mathbf{x}_t)}\boldsymbol{\omega}_t, \quad (4)$$

with $\tilde{\mathbf{f}}(\mathbf{x}_t) := \mathbf{f}(\mathbf{x}_t) + \mathbf{g}(\mathbf{x}_t)\boldsymbol{\pi}^*(\mathbf{x}_t)$, become stochastic.

The following assumptions are made for the closed-loop dynamics and the perturbed policy:

Assumption 1: The agent performs a goal-directed task, where the goal is defined by a unique target state $\mathbf{x}^* \in \mathcal{X}$.

Assumption 2: The variance in (3) vanishes at the target; $\lim_{\mathbf{x}_t \rightarrow \mathbf{x}^*} \boldsymbol{\Sigma}(\mathbf{x}_t) = \mathbf{0}$, such that $\tilde{\boldsymbol{\pi}}(\mathbf{x}^*) = \boldsymbol{\pi}^*(\mathbf{x}^*)$.

Intuitively, Assumption 2 states that the agent should act more deterministic close to the target \mathbf{x}^* , which is necessary for task completion according to Assumption 1. Since the target is unique, \mathbf{x}^* coincides with the minimization of the agent's cost function. Therefore, any agent that successfully performs the task has to act on the system such that it asymptotically converges to the desired final state \mathbf{x}^* . This has to hold true regardless of the stochasticity due to random perturbations in (4). In order to formalize this property, we introduce the following concept of stability.

Definition 1 ([7]): A system (4) has an asymptotically stable equilibrium \mathbf{x}^* on the set \mathcal{X} in probability if

1) for all $\varepsilon > 0$, $d > 0$, there exist $\delta > 0$, $t_0 \geq 0$ such that

$\|\mathbf{x}_0 - \mathbf{x}^*\| < \delta$ implies $P\{\|\mathbf{x}_t - \mathbf{x}^*\| < d\} \geq 1 - \varepsilon$, $\forall t \geq t_0$.

2) $P\{\lim_{t \rightarrow \infty} \|\mathbf{x}_t - \mathbf{x}^*\| = 0\} = 1$ for all $\mathbf{x}_0 \in \mathcal{X}$.

Intuitively, Definition 1 is the probabilistic analogue of the classical stability definition in the sense of Lyapunov [8]. Therefore, we subsequently refer to this property as asymptotic stability. Consequently, using Assumption 1 and 2, we can express the asymptotic minimization of the objective despite of stochasticity as follows.

Assumption 3: Policy (3) renders system (1) asymptotically stable at the target state \mathbf{x}^* .

Based on this assumption, we consider the problem of determining the value function

$$V^*(\mathbf{x}) = \sum_{t=0}^{\infty} \gamma^t l(\mathbf{x}_t, \boldsymbol{\pi}^*(\mathbf{x}_t)), \quad \mathbf{x}_0 = \mathbf{x}, \quad (5)$$

which describes the minimum cost-to-go when starting in a state \mathbf{x} and following the optimal policy $\boldsymbol{\pi}^*$ thereafter. Due to the Bellman equality, the optimal policy $\boldsymbol{\pi}^*$ can be defined equivalently as

$$\boldsymbol{\pi}^*(\cdot) = \operatorname{argmin}_{\mathbf{u}} [l(\cdot, \mathbf{u}) + \gamma V^*(\mathbf{f}(\cdot) + \mathbf{g}(\cdot)\mathbf{u})]. \quad (6)$$

Using this identity, the problem of estimating the stabilizing value function can now be formulated as a constrained functional optimization problem. Here, the posterior of V is maximized by evaluating the alignment of state-visitation probabilities, under the associated optimal policy $\boldsymbol{\pi}^*$, with observed trajectories $\{\tau_1, \dots, \tau_N\}$, where $\tau_n = \{\mathbf{x}_t^n\}_{t=0}^T$ and $N \in \mathbb{N}_+$. In addition, Assumption 3 must hold true, which leads to the following formulation:

$$V^* = \operatorname{argmax}_{V \in \mathcal{C}(\mathcal{X})} P\{V | \tau_1, \dots, \tau_N\} \quad (7a)$$

such that $\tilde{\boldsymbol{\pi}}$ asymptotically stabilizes (1). (7b)

From (7b) follows an additional constraint to the solution V^* . This structural constraint holds over the whole solution space and not only at data points, therefore, providing information at unobserved states. Furthermore, any policy derived from V^* is guaranteed to be stabilizing by design.

III. STABILITY-CERTIFIED INVERSE REINFORCEMENT LEARNING

In order to infer the value function V^* from training data τ_1, \dots, τ_N , while considering the stability constraint on the corresponding perturbed optimal policy $\tilde{\boldsymbol{\pi}}$, we exploit the inverse optimal relationship between value functions and CLFs. This allows us to transform the constraint (7b) into a Lyapunov-type constraint on the optimal value function in Section III-A. By considering the optimal value function as a control Lyapunov function, we can approximate the posterior maximization (7a) in a closed-form in Section III-B. In Section III-C, the problem is finally cast as a constrained kernel regression problem in order to efficiently solve the problem using machine learning techniques.

A. Lyapunov-Constrained Value Function Approximation

A practical method to ascertain the convergence property as introduced in Definition 1, without solving the underlying dynamical system equations, is by means of Lyapunov

stability theory [7]. Since the state space of the considered dynamics is bounded to a compact set, the following relaxation regarding positive definiteness constraints of Lyapunov function candidates can be concluded here.

Lemma 1: Consider a stochastic system of the form (4), which generates trajectories with states \mathbf{x}_t , such that $\mathbf{x}_0 \in \mathcal{X}$ implies $\mathbf{x}_t \in \mathcal{X}$ almost surely for all $t \in \mathbb{N}_+$ and compact sets \mathcal{X} . If there exists a continuous $W: \mathcal{X} \rightarrow \mathbb{R}_{+,0}$, such that

$$\mathbb{E}[W(\mathbf{x}_{t+1}) | \mathbf{x}_t] - W(\mathbf{x}_t) < 0, \quad \forall \mathbf{x}_t \in \mathcal{X} \setminus \{\mathbf{x}^*\} \quad (8a)$$

$$\mathbb{E}[W(\mathbf{x}_{t+1}) | \mathbf{x}_t] - W(\mathbf{x}_t) = 0, \quad \mathbf{x}_t = \mathbf{x}^* \quad (8b)$$

then, the system with equilibrium point \mathbf{x}^* is asymptotically stable in the sense of Definition 1.

Proof: We prove this lemma by showing that $\hat{W} = W - \min_{\mathbf{x} \in \mathcal{X}} W(\mathbf{x})$ is a Lyapunov function, which implies asymptotic stability in probability [7]. The decrease of \hat{W} along system trajectories is ensured by (8a), such that it remains to show that \hat{W} is positive definite. This is identical to proving that \mathbf{x}^* is the only minimizer of W on \mathcal{X} . In order to show this let $\mathcal{M} = \{\mathbf{x} \in \mathcal{X} : \min_{\mathbf{x}' \in \mathcal{X}} W(\mathbf{x}') = W(\mathbf{x})\}$ be the set of minimizers of W . Assume that there exists a $\hat{\mathbf{x}} \neq \mathbf{x}^*$, $\hat{\mathbf{x}} \in \mathcal{M}$. This implies that

$$\mathbb{E}[W(\hat{\mathbf{f}}(\hat{\mathbf{x}}) + \mathbf{g}(\hat{\mathbf{x}})\sqrt{\Sigma(\hat{\mathbf{x}})\boldsymbol{\omega}})] - W(\hat{\mathbf{x}}) \geq 0 \quad (9)$$

by definition of the set \mathcal{M} . However, this is a contradiction of (8a), such that \mathbf{x}^* is the unique minimizer of W . Therefore, \hat{W} is positive definite, which concludes the proof. ■

A major strength of Lyapunov stability theory is the existence of converse theorems, i.e., under the assumption of stability, a Lyapunov function is guaranteed to exist. Since the Lyapunov-like function W considered in Lemma 1 is merely a shifted Lyapunov function, this property extends. We exploit this together with the fact that every Lyapunov function is an optimal value function for some meaningful cost [9]. This so called inverse optimality property is employed to formulate the original problem (7) as a Lyapunov-constrained optimization problem, which is guaranteed to be feasible. This is shown in the following result.

Lemma 2: The Lyapunov-constrained functional optimization problem

$$V^* = \operatorname{argmax}_{V \in \mathcal{C}(\mathcal{X})} P\{V | \tau_1, \dots, \tau_N\}, \quad (10a)$$

$$\text{s.t.} \quad \mathbb{E}[\Delta V(\mathbf{x})] < 0, \quad \forall \mathbf{x} \in \mathcal{X} \setminus \{\mathbf{x}^*\} \quad (10b)$$

$$\mathbb{E}[\Delta V(\mathbf{x}^*)] = 0, \quad (10c)$$

with $\mathbb{E}[\Delta V(\mathbf{x}_t)] = \mathbb{E}[V(\mathbf{x}_{t+1}) | \mathbf{x}_t] - V(\mathbf{x}_t)$ is feasible.

Proof: Since the perturbed optimal policy $\tilde{\boldsymbol{\pi}}$ asymptotically stabilizes system (1) in the sense of Definition 1, the converse Lyapunov theorem guarantees the existence of a Lyapunov function satisfying constraints (10b)-(10c) [10]. Moreover, every Lyapunov function also resembles an optimal value function for some meaningful cost l [9], which makes the Lyapunov function a valid solution of (10). ■

Note that the realization of the stability constraint (7b) through the constraints (10b) and (10c) does not pose a restriction to the solution space, since every Lyapunov-like function is an optimal value function for a continuum of

stage costs with different optimal control laws. Hence, the considered stability constraints still allow to infer different agent preferences modelled by the intrinsic costs l .

B. Closed-Form Likelihood Expression

While the focus of the previous section lies on deriving a feasible expression for the stability constraint, we deal with the problem of maximizing the posterior (7a) in this section. In order to solve this problem, we follow a Bayesian approach, which directly leads to the proportional relationship

$$P\{V | \tau_1, \dots, \tau_N\} \propto P\{\tau_1, \dots, \tau_N | V\}P\{V\}, \quad (11)$$

where $P\{V\}$ denotes the prior probability distribution over value functions V , which is a design choice. Since the trajectories τ_1, \dots, τ_N are generated independently using the perturbed optimal policy $\tilde{\boldsymbol{\pi}}$, they are conditionally independent given the optimal policy $\boldsymbol{\pi}^*$. Due to (6), this implies

$$P\{\tau_1, \dots, \tau_N | V\} = P\{\tau_1 | V\} \dots P\{\tau_N | V\}. \quad (12)$$

Similarly, observed states \mathbf{x}_t^n along a trajectory τ_n are conditionally Markovian given the optimal policy $\boldsymbol{\pi}^*$. Thus, it follows by the same argument as before that

$$P\{\tau_n | V\} = P\{\mathbf{x}_0^n | \tau_n\} \prod_{t=1}^T P\{\mathbf{x}_t^n | V, \mathbf{x}_{t-1}^n\}, \quad (13)$$

where $P\{\mathbf{x}_0 | \tau_n\} = P\{\mathbf{x}_0\}$ is the prior initial state distribution. Since this prior is independent of V , we have

$$P\{\tau_n | V\} \propto \prod_{t=1}^T P\{\mathbf{x}_t^n | V, \mathbf{x}_{t-1}^n\}. \quad (14)$$

Because the agent behaves optimally, it follows that for each of these probabilities, the next state \mathbf{x}_t^n generated by the policy $\tilde{\boldsymbol{\pi}}$ applied to the state \mathbf{x}_{t-1}^n can be determined using (4). Since the considered perturbations $\boldsymbol{\omega}$ to the optimal policy (3) are truncated normally distributed, a closed form expression for the probabilities can be obtained. For improved readability, we assume $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, which is not particular restrictive in practice, since the deterministic part of (3) dominates the perturbations in general. Accordingly, resulting in the following expression for the probabilities

$$P\{\mathbf{x}_t^n | V, \mathbf{x}_{t-1}^n\} = \quad (15)$$

$$\mathcal{N}(\mathbf{x}_t^n | \mathbf{f}(\mathbf{x}_{t-1}^n) + \mathbf{g}(\mathbf{x}_{t-1}^n)\boldsymbol{\pi}^*(\mathbf{x}_{t-1}^n), \mathbf{g}^\top(\mathbf{x}_{t-1}^n)\boldsymbol{\Sigma}(\mathbf{x}_{t-1}^n)\mathbf{g}(\mathbf{x}_{t-1}^n)),$$

where $\boldsymbol{\pi}^*$ is defined in (6). By combining all these equalities, we obtain the log-likelihood

$$\log(P\{V | \tau_1, \dots, \tau_N\}) \propto \log(P\{V\}) - \sum_{n=1}^N \sum_{t=1}^T (\mathbf{e}_t^n)^\top \boldsymbol{\Gamma}_t^n \mathbf{e}_t^n, \quad (16)$$

where

$$\mathbf{e}_t^n = \mathbf{x}_t^n - \mathbf{f}(\mathbf{x}_{t-1}^n) - \mathbf{g}(\mathbf{x}_{t-1}^n)\boldsymbol{\pi}^*(\mathbf{x}_{t-1}^n) \quad (17)$$

$$\boldsymbol{\Gamma}_t^n = (\mathbf{g}^\top(\mathbf{x}_{t-1}^n)\boldsymbol{\Sigma}(\mathbf{x}_{t-1}^n)\mathbf{g}(\mathbf{x}_{t-1}^n))^{-1}. \quad (18)$$

This function measures how well the perturbed optimal policy $\tilde{\boldsymbol{\pi}}$ approximates the observed demonstrations τ_1, \dots, τ_N of the agent. Thereby, the optimal value function V^* is indirectly inferred, as it induces the unperturbed optimal policy $\boldsymbol{\pi}^*$ via (6), thus, influencing the loss in (16).

However, in order to maximize (16) the optimal policy $\boldsymbol{\pi}^*$ is needed. A closed-form expression for $\boldsymbol{\pi}^*$ is generally difficult to obtain and is only available for some problems, e.g., linear systems with quadratic costs. Although (16) can be practically employed in (7) by approximately solving the optimal control problem to determine the optimal policy $\boldsymbol{\pi}^*$, this approach is computationally demanding in general [4], particularly considering that the stability constraint needs to be enforced. Therefore, we make use of the fact that we already know that V is a Lyapunov-like function due to the constraints (10b) and (10c). Additionally, we know from the inverse optimality property and Assumption 3 that there exists a CLF, which is equivalent to V^* [9]. This approach allows us to construct closed-form control laws $\hat{\boldsymbol{\pi}}$ for many, more general system classes, e.g., control-affine systems [11], [12]. Hence, instead of employing the relationship between the optimal value function V^* and the optimal policy $\boldsymbol{\pi}^*$, we propose to approximate the inference problem by exploiting the closed-form control law $\hat{\boldsymbol{\pi}}$ coming along with the CLF. Analogue to before, the Lyapunov-like function V is fitted by evaluating the incurred loss under the closed-form control law $\hat{\boldsymbol{\pi}}$. This immediately leads to the approximation

$$\log(P\{V | \tau_1, \dots, \tau_N\}) \approx \log(P\{V\}) - \sum_{n=1}^N \sum_{t=1}^T (\hat{\boldsymbol{e}}_t^n)^\top \boldsymbol{\Gamma}_t^n \hat{\boldsymbol{e}}_t^n, \quad (19)$$

where

$$\hat{\boldsymbol{e}}_t^n = \mathbf{x}_t^n - \mathbf{f}(\mathbf{x}_{t-1}^n) - \mathbf{g}(\mathbf{x}_{t-1}^n) \hat{\boldsymbol{\pi}}(\mathbf{x}_{t-1}^n). \quad (20)$$

This log-likelihood still allows to infer V indirectly by fitting a policy, merely substituting the optimal policy $\boldsymbol{\pi}^*$ by the closed-form policy $\hat{\boldsymbol{\pi}}$. Thereby, optimization with the approximate log-likelihood (19) yields a function V , which reflects the agent's preferences, since the associated policy $\hat{\boldsymbol{\pi}}$ best possibly represents the observed demonstrations τ_1, \dots, τ_N . Even though the obtained V is not an optimal value function in general, this formulation allows to encode arbitrary agent behaviors in principle, only limited by the flexibility of the closed-form control policy. This flexibility is often sufficient to represent arbitrary training data. In the following theorem this is exemplarily shown for a time-discrete system with control-affine structure (1) and policy

$$\hat{\boldsymbol{\pi}}(\mathbf{x}) = -\beta [\nabla V(\mathbf{f}(\mathbf{x})) \mathbf{g}(\mathbf{x})]^\top \quad (21)$$

where $\beta > 0$ and (21) is taken from [12].

Theorem 1: Given a control-affine system and a data set with N trajectories τ_n generated by an agent with stabilizing deterministic policy $\hat{\boldsymbol{\pi}}$, i.e., $\boldsymbol{\Sigma} = \mathbf{0}$. If $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}')$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, $\mathbf{x} \neq \mathbf{x}'$, there always exists a function V such that the closed-loop control law (21) generates the training data.

Proof: It is straightforward to see that the control Lyapunov policy (21) generates given trajectories τ_n if

$$\mathbf{g}(\mathbf{x}_k^n) \boldsymbol{\pi}(\mathbf{x}_k^n) = -\beta \mathbf{g}(\mathbf{x}_k^n) \mathbf{g}^\top(\mathbf{x}_k^n) (\nabla V(\mathbf{f}(\mathbf{x}_k^n)))^\top \quad (22)$$

holds for all $t = 0, \dots, T-1$, $n = 1, \dots, N$. This immediately gives explicit conditions for the gradient of V , e.g., if $\mathbf{g}(\mathbf{x}_k^n)$ has full rank, we have

$$\beta (\nabla V(\mathbf{f}(\mathbf{x}_k^n)))^\top = -(\mathbf{g}(\mathbf{x}_k^n) \mathbf{g}^\top(\mathbf{x}_k^n))^{-1} \mathbf{g}(\mathbf{x}_k^n) \boldsymbol{\pi}(\mathbf{x}_k^n). \quad (23)$$

Therefore, this condition requires the existence of a function with specified gradient values, which can be achieved with a continuous function when its arguments are different. Since this is satisfied by assumption, i.e., $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}')$ for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, $\mathbf{x} \neq \mathbf{x}'$, a function V generating trajectories τ^n using the policy (21) is guaranteed to exist. ■

Even though this finding demonstrates that the approximation is flexible enough to represent the training data, a possible optimality gap between $\boldsymbol{\pi}^*$ and $\hat{\boldsymbol{\pi}}$ remains. However, for uninformative priors in (19) and assuming the noiseless case $\hat{\boldsymbol{\pi}} = \boldsymbol{\pi}^*$, it is possible to make the loss in (20) arbitrarily small, such that the closed-form policy $\hat{\boldsymbol{\pi}}$ and optimal policy $\boldsymbol{\pi}^*$ agree on the training data. Since policies $\hat{\boldsymbol{\pi}}$ as defined in (21) follow the gradient descent paradigm similarly as optimal policies, the gradient of a convex optimal value function exhibits the same direction as that of the learned CLF on the data. Hence, the level sets of the optimal value function V^* and the CLF V are similar in this case.

C. Formulation as Constrained Kernel Regression Problem

While (10) in combination with the approximate log-likelihood (19) is theoretically appealing for inferring V , it requires solving a functional optimization problem, which is not tractable in practice. Therefore, expressive function approximators are needed. Due to the lack of structural knowledge regarding the value function V of a human demonstrator, kernel-methods [13] are suitable, since this class of machine learning techniques does not take a predefined structure, but instead constructs one solely using provided training data. To achieve this, they rely on a kernel function

$$k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}, \quad (\mathbf{x}, \mathbf{x}') \mapsto k(\mathbf{x}, \mathbf{x}'), \quad (24)$$

which defines a structure to the data by acting as a similarity measure between data points. Every kernel defines a so called reproducing kernel Hilbert space (RKHS)

$$\mathcal{H}_k = \left\{ h: h(\cdot) = \sum_{i=1}^N \alpha_i k(\cdot, \mathbf{z}_i), \|h\|_k = \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \alpha_i \alpha_j k(\mathbf{z}_i, \mathbf{z}_j) < \infty \right\}, \quad (25)$$

where $\alpha_i \in \mathbb{R}$ and $\mathbf{z}_i \in \mathcal{X}$. The RKHS comprises all the functions that can be approximated through the kernel.

Since we cannot optimize over arbitrary functions, we restrict ourselves to a parameterization of continuous functions. Linear combinations of universal kernels are particularly well-suited, since their spanned function space \mathcal{H}_k is known to be dense in the continuous functions, thus, allowing to approximate continuous functions arbitrarily well [14]. Moreover, due to the analogy between certain kernel regression problems and Gaussian process regression [15], the prior $P(V)$ can be intuitively defined, such that $\log(P(V)) \propto -\|V\|_k$. This results in the constrained kernel regression problem

$$\min_{V \in \mathcal{H}_k} \sum_{n=1}^N \sum_{t=1}^T (\hat{\boldsymbol{e}}_t^n)^\top \boldsymbol{\Gamma}_t^n \hat{\boldsymbol{e}}_t^n + \lambda \|V\|_k^2, \quad (26a)$$

$$\text{s.t.} \quad \mathbb{E}[\Delta V(\mathbf{x})] < 0 \quad \text{for all } \mathbf{x} \in \mathcal{X} \setminus \{\mathbf{x}^*\}, \quad (26b)$$

$$\mathbb{E}[\Delta V(\mathbf{x}^*)] = 0, \quad (26c)$$

where $\lambda \in \mathbb{R}_+$ is a small constant.

The stability constraint (26b) remains a restriction preventing the direct implementation of this optimization problem, as it must hold for an uncountable, infinite set of states \mathbf{x} . This condition can only be resolved exactly in simple problems, e.g., quadratic Lyapunov functions and linear systems. Therefore, we relax the stability condition and allow an increase of the Lyapunov function along trajectories in a small neighborhood $\mathcal{B}_\xi = \{\mathbf{x} \in \mathcal{X} : \|\mathbf{x} - \mathbf{x}^*\| \leq \xi\}$ of the equilibrium \mathbf{x}^* , such that convergence of the closed-loop system trajectories to a neighborhood of the equilibrium is still guaranteed [16]. This allows us to approximate (26) as a practically tractable optimization problem using a discretization of the stability conditions, such that strong theoretical guarantees are retained as shown in the following theorem.

Theorem 2: Let $\mathcal{X}_\xi = \{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_{N_\xi}\}$, $N_\xi \in \mathbb{N}_+$ be a discretization over \mathcal{X} with grid constant $\xi \in \mathbb{R}_+$, i.e., $\xi = \max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{x}' \in \mathcal{X}_\xi} \|\mathbf{x} - \mathbf{x}'\|$. Moreover, consider a continuous kernel k , continuous system dynamics \mathbf{f} and \mathbf{g} and let $L_{\Delta V}$ denote the Lipschitz constant of $\mathbb{E}[\Delta V(\mathbf{x})]$. Then, any solution $V \in \mathcal{H}_k$ of the optimization problem

$$\min_{V \in \mathcal{H}_k} \sum_{n=1}^N \sum_{t=1}^T (\hat{\mathbf{e}}_t^n)^\top \Gamma_t^n \hat{\mathbf{e}}_t^n + \lambda \|V\|_k^2, \quad (27a)$$

$$\text{s.t.} \quad \mathbb{E}[\Delta V(\hat{\mathbf{x}}_i)] < -L_{\Delta V} \xi, \quad \forall \hat{\mathbf{x}}_i \in \mathcal{X}_\xi \setminus \{\hat{\mathbf{x}}^*\} \quad (27b)$$

$$\mathbb{E}[\Delta V(\hat{\mathbf{x}}^*)] \leq 0 \quad (27c)$$

$$\hat{\mathbf{x}}^* = \underset{\hat{\mathbf{x}}_i \in \mathcal{X}_\xi}{\operatorname{argmin}} V(\hat{\mathbf{x}}_i), \quad (27d)$$

admits a representation of the form

$$V(\cdot) = \sum_{n=1}^N \alpha_n k(\mathbf{x}_n, \cdot) + \sum_{i=1}^{N_\xi} \alpha_{N+i} k(\hat{\mathbf{x}}_i, \cdot), \quad (28)$$

and satisfies condition (10b) for system (4) on all $\mathcal{X} \setminus \mathcal{B}_\xi$.

Proof: The constrained optimization problem (27) can be transformed into an unconstrained one by using Lagrange multipliers. Since the regularizer $\|V\|_k^2$ is strictly increasing on $[0, \infty[$, the obtained unconstrained regularized risk functional conforms to the requirements of the generalized representer theorem, which guarantees that a solution of the form (28) is admitted [17]. This proves the first part of the theorem. The second part follows from continuity of the kernel k , such that all functions of the form (28) are continuous. Together with the continuity of \mathbf{f} and \mathbf{g} follows that ΔV is continuous, which immediately implies the existence of a Lipschitz constant $L_{\Delta V}$ on \mathcal{X} . Therefore, tightening the stability constraint (10b) by $L_{\Delta V} \xi$ on the grid $\mathcal{X}_\xi \setminus \{\hat{\mathbf{x}}^*\}$ ensures that it is satisfied for all $\mathbf{x} \in \mathcal{X} \setminus \mathcal{B}_\xi$, which concludes the proof. ■ In order to transform the stability conditions into tractable constraints, we adopt a discretization approach in Theorem 2, which is a commonly used method in the context of numerical analysis of non-parametric Lyapunov functions [18], [19]. It can be seen clearly that this approach requires trading-off computational complexity and flexibility of the feasible solutions, since small values of ξ result in a slight constraint tightening, but in turn cause a high computational complexity. This can be exploited by employing multi-

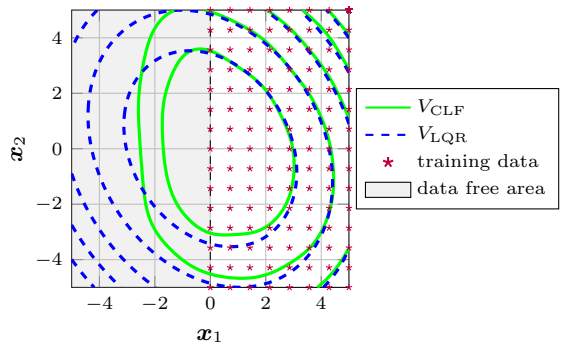


Fig. 1. Contours of the ground truth value function illustrated in blue, dashed lines and the learned CLF given in green. The purple stars depict the training data, while the greyed out part shows the data free area.

resolution grids, in which the grid constant is adapted to the training data and the dynamics, such that high flexibility is provided where required, while unnecessary computational effort is avoided. In order to further reduce the conservatism of the constraint tightening, multi-resolution grids can be combined with local Lipschitz constants, which can be efficiently computed for kernel methods using GPUs [20].

IV. EVALUATION

For the evaluation of the proposed approach, we examine two scenarios. First, the principle capacity of the CLF to approximate an optimal value function is shown, by considering a linear-quadratic problem for which ground truth information is known. Secondly, the flexibility of the method is demonstrated by learning from human demonstrations in a goal-directed movement task.

In the first problem, we consider a system governed by the linear dynamics $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t$ and define a quadratic running cost $c(\mathbf{x}_t, \mathbf{u}_t) = \mathbf{x}_t^\top \mathbf{Q}\mathbf{x}_t + \mathbf{u}_t^\top \mathbf{R}\mathbf{u}_t$, where

$$\mathbf{A} = \begin{bmatrix} 1 & 0.1 \\ 0 & 0.9 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix}.$$

Using the LQR it is straightforward to compute the optimal value function, which takes a quadratic form here. To learn the CLF we observe the one step trajectories generated by the deterministic optimal feedback control, i.e., $\boldsymbol{\Sigma} = \mathbf{0}$, at $N = 120$ points. The state space is bounded in $\mathcal{X} \in [-5, 5]^2$ and an 11×11 grid spanning equidistantly over the state space is used to ensure the stability constraint. The regularization factor is set to $\lambda = 1/\bar{N}$, where $\bar{N} = NT + N_\xi$ equals the total number of observed data points. The optimization is initialized with $\boldsymbol{\alpha} = \mathbf{0}$ and the control law is given by (21) with $\beta = 0.2$. In Fig. 1 the contour plots for the ground truth value function and the learned CLF are depicted. It can clearly be seen that both functions agree in the shape of their level curves in areas of the state space, where training data is provided. Thus, the learned Lyapunov-like function constitutes an appropriate approximation for the optimal value function. However, the approximation does not take a quadratic form in the data free area, since the kernel-based approach does not assume a predefined structure and requires data to build one. Therefore, our proposed method remains flexible in approximating the

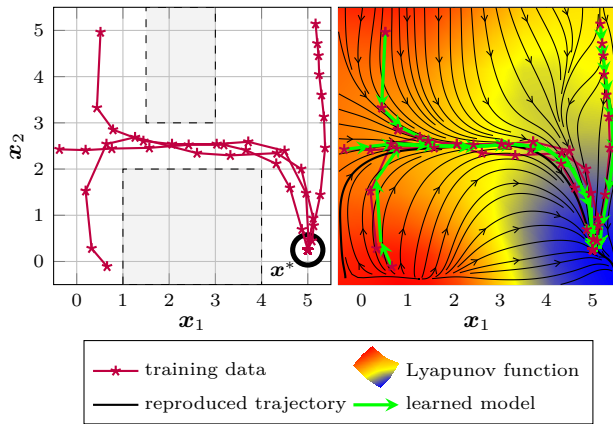


Fig. 2. (Left) Depiction of the goal-directed movement task environment. The goal position \mathbf{x}^* is indicated by the bold circle on the bottom right. The grey, transparent rectangles visualize areas associated with high costs. With purple lines the human demonstrations are depicted. (Right) Visualization of the kernel regression result for the goal-directed movement task. The learned Lyapunov function is visualized in form of a projected surface plot and as streamlines (black) along the gradient direction. The simulated trajectories under the inferred control law $\hat{\pi}$ are given in green.

value function and does not limit the expressiveness of resulting control policies. We demonstrate this flexibility in the following learning from human demonstrations task.

The task is performed in a two-dimensional, continuous space, where the demonstrator attempts to reach a target position, whilst avoiding areas associated with high costs. The environment and human generated demonstrations are depicted in Fig. 2 on the left. Generally, this setup can be thought of as an abstraction of typical programming by demonstration tasks, i.e., human reaching motions. The considered system is driven by the nonlinear dynamics

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_1(1 + s(x_1)) \\ x_2(1 + s(x_2)\cos^2(\frac{\pi}{5}x_2)) \end{bmatrix} \quad (29)$$

where $s(x) = (2/(1 + \exp(x - 2.5)) - 1)\sin^2(\pi x/5)$ and $\mathbf{g}(\mathbf{x}) = (2 - \cos^2(\pi x_2/5)\sin^2(\pi x_1/5))\mathbf{I}_2$, $\mathbf{x} = [x_1 \ x_2]^T$.

The state space is bounded in $\mathcal{X} \in [-0.5, 5.5]^2$ and a 10×10 grid spanning equidistantly over the state space is used. The number of observed trajectories is $N = 4$ and we sample $T = 11$ observations from each trajectory. For the computation of the expectation in (27) we draw and average over five points generated by a Gaussian distribution with standard deviation $\sigma = 0.05$, centered around the deterministic V . The regularization factor λ , the initialization of $\boldsymbol{\alpha}$ and the control factor β stay the same as before.

From the results in Fig. 2 it can clearly be seen that the proposed method is capable of simultaneously learning a Lyapunov function and an accompanying control law that drives any point in the bounded state space to the desired equilibrium point, i.e., the system is asymptotically stable. When comparing the reproduced trajectories generated by the control law (21) with the training data, it becomes apparent that the inferred control policy reflects the depicted preferences in the human demonstrations. This demonstrates that the learned Lyapunov function is flexible enough to encode the observed behavior in a similar fashion as a

value function would. Furthermore, this is accomplished in a data efficient manner, since our method only requires 44 human-generated data points for the presented results.

V. CONCLUSION

This paper presents a novel perspective on the IRL inference problem, by using tools from dynamical system theory. Based on the concept of inverse optimality, the inference task is reformulated to learning CLFs from demonstrations. Through kernel-based non-parametric regression the CLFs are learned efficiently and with minimum structural assumptions. Finally, derived policies are certifiably stable and are shown to encode the preferences of the human demonstrator.

REFERENCES

- [1] A. Hussein, M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, no. 2, 2017.
- [2] P. Abbeel and A. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the International Conference on Machine Learning*, New York, NY, USA, 2004, p. 1.
- [3] B. Ziebart, A. Maas, J. Bagnell, and A. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2008, pp. 1433–1438.
- [4] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *Proceedings of the International Conference on Machine Learning*, 2016, pp. 49–58.
- [5] A. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Comput.*, vol. 25, no. 2, pp. 328–373, 2013.
- [6] S. Khansari-Zadeh and A. Billard, "Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions," *Rob. Auton. Syst.*, vol. 62, no. 6, pp. 752–765, 2014.
- [7] Y. Li, W. Zhang, and X. Liu, "Stability of Nonlinear Stochastic Discrete-Time Systems," *J. Appl. Math.*, vol. 2013, no. 2, 2013.
- [8] H. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.
- [9] R. Freeman and P. Kokotovic, "Inverse optimality in robust stabilization," *SIAM J Control Optim.*, vol. 34, no. 4, pp. 1365–1391, 1996.
- [10] A. Teel, J. Hespanha, and A. Subbaraman, "A Converse Lyapunov Theorem and Robustness for Asymptotic Stability in Probability," *IEEE Trans. Automat. Contr.*, vol. 59, no. 9, pp. 2426–2441, 2014.
- [11] C. Byrnes, W. Lin, and B. Ghosh, "Stabilization of discrete-time nonlinear systems by smooth state feedback," *Syst. Control. Lett.*, vol. 21, no. 3, pp. 255–263, 1993.
- [12] A. Bacciotti and A. Biglio, "Some remarks about stability of nonlinear discrete-time control systems," *Nonlinear Differ. Equ. Appl.*, vol. 8, no. 4, pp. 425–438, 2001.
- [13] T. Hofmann, B. Schölkopf, and A. Smola, "Kernel Methods in Machine Learning," *Ann. Stat.*, vol. 36, no. 3, pp. 1171–1220, 2008.
- [14] C. Micchelli, Y. Xu, and H. Zhang, "Universal Kernels," *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, 2006.
- [15] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press, 2006.
- [16] J. Umlauf, A. Lederer, and S. Hirche, "Learning Stable Gaussian Process State Space Models," in *Proceedings of the American Control Conference*, 2017, pp. 1499–1504.
- [17] B. Schölkopf, R. Herbrich, and A. Smola, "A Generalized Representer Theorem," in *Proceedings of the 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*, 2001, pp. 416–426.
- [18] F. Berkenkamp, R. Moriconi, A. Schoellig, and A. Krause, "Safe Learning of Regions of Attraction for Uncertain, Nonlinear Systems with Gaussian Processes," in *Proceedings of the IEEE Conference on Decision and Control*, 2016, pp. 4661–4666.
- [19] A. Lederer and S. Hirche, "Local Asymptotic Stability Analysis and Region of Attraction Estimation with Gaussian Processes," in *Proceedings of the IEEE Conference on Decision and Control*, 2019.
- [20] A. Lederer, M. Kessler, and S. Hirche, "GP3 : A Sampling-based Analysis Framework for Gaussian Processes," in *Proceedings of the 21st IFAC World Congress*, 2020.