52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

Inverse optimal control for deterministic continuous-time nonlinear systems

Miles Johnson[1], Navid Aghasadeghi[2], and Timothy Bretl[1]

Abstract— **Inverse optimal control is the problem of computing a cost function with respect to which observed state and input trajectories are optimal. We present a new method of inverse optimal control based on minimizing the extent to which observed trajectories violate first-order necessary conditions for optimality. We consider continuous-time deterministic optimal control systems with a cost function that is a linear combination of known basis functions. We compare our approach with three prior methods of inverse optimal control. We demonstrate the performance of these methods by performing simulation experiments using a collection of nominal system models. We compare the robustness of these methods by analysing how they perform under perturbations to the system. To this purpose, we consider two scenarios: one in which we exactly know the set of basis functions in the cost function, and another in which the true cost function contains an unknown perturbation. Results from simulation experiments show that our new method is more computationally efficient than prior methods, performs similarly to prior approaches under large perturbations to the system, and better learns the true cost function under small perturbations.**

I. Introduction

In the problem of optimal control we are asked to find input and state trajectories that minimize a given cost function. In the problem of inverse optimal control we are asked to find a cost function with respect to which observed input and state trajectories are optimal. Methods of inverse optimal control are beginning to find widespread application in robotics. In this paper, we consider this problem under deterministic continuous-time nonlinear systems and cost functions modeled by a linear combination of known basis functions. Three existing methods which solve this problem are the following:

- The max-margin inverse reinforcement learning method of Abbeel and Ng [1]. This method is motivated by the problem of efficiently automating vehicle navigation tasks which currently require human expert operation. This method works by trying to learn a cost function that, when minimized, yields a trajectory with similar features as the expert. This method recently contributed to a framework which enables autonomous helicopter aerobatic flight based on observations of human expert pilots.
- The maximum-margin planning method of Ratliff, et al. [2]. This method shares the motivation of Abbeel and Ng, and works by minimizing a regularized risk function using an incremental subgradient method. This method contributed to a framework which mimics human driving of an autonomous mobile robot in complex off-road terrain.
- The method of Mombaur, et al. which we will call bi-level inverse optimal control [3]. This work is motivated by the problem of generating humanoid robot behavior which is similar to natural human motion. This method works by minimizing the sum squared error between predicted and observed trajectories. This method is applied to develop a model of human goal-oriented locomotion in the plane (i.e. paths taken during goal-oriented walking tasks) using observations from motion capture, and implement the model on a humanoid robot.

Despite differences in how learning is performed, these methods share common structure. One goal of this paper is to explain this common structure and compare these methods on a set of example problems. One common component to these algorithms is particularly important. Each method contains an inner loop which computes a *predicted trajectory* by minimizing a candidate cost function. In other words, each method solves a forward optimal control problem repeatedly in an inner loop. This can often lead to a computational bottleneck. The other goal of this paper is to develop an approach which does not solve a forward optimal control problem repeatedly in an inner loop. Our method, inspired by ideas from inverse optimization in [4], makes the assumption that observations may arise from a system which is only *approximately optimal*. We define how optimal a trajectory is based on how closely it satisfies necessary conditions for optimal control. This assumption allows us to define residual functions based on these necessary conditions which, when minimized over the unknown parameters, yields a solution which makes the observations most optimal. As we will show, this new approach reduces to solving a matrix Riccati differential equation followed by one least-squares minimization.

It is unclear at this point how all of these methods compare in terms of prediction accuracy, computational complexity and robustness to system perturbations. In this paper, we explore the performance of these methods using three example systems: (1) linear quadratic regulation, (2) quadratic regulation of a kinematic unicycle, and (3) calibration of an elastic rod. We compare the robustness of these methods by analysing how they perform under perturbations to the system. To this purpose, we consider two scenarios: one in which we exactly know the set of basis functions in the cost function, and another in which the true cost function contains an unknown perturbation. Results from simulation

[1]Miles Johnson and Timothy Bretl are with the Department of Aerospace Engineering at the University of Illinois at Urbana-Champaign.

[2]Navid Aghasadeghi is with the Department of Electrical Engineering at the University of Illinois at Urbana-Champaign.

experiments show that our new method is more computationally efficient than prior methods, performs similarly to prior approaches under large perturbations to the system, and better learns the true cost function under small perturbations.

The rest of the paper proceeds as follows. In Section II we discuss related work and note the variety of problems to which inverse optimal control and related methods are applied. In Section III we describe the class of systems we consider, and the associated inverse optimal control problem. In Section IV we describe the existing methods of inverse optimal control with which we compare our new method [1]–[3]. In Section V we develop our new method based on necessary conditions for optimal control. In Section VI we describe simulation experiments,and Section VII presents results and discussion.

II. Related Work

Inverse optimal control is often used as a solution approach to the broad problem of learning from demonstration, which is often referred to as imitation learning or apprenticeship learning. The problem of learning from demonstration is to derive a control policy (a mapping from states to actions) from examples, or demonstrations, provided by a teacher. Demonstrations are typically considered to be sequences of state-action pairs recorded during the teacher's demonstration.

There are generally two methods of approach within learning from demonstration. One approach is to learn a map from states to actions using classification or regression. Argall, et al. provide a survey of the work in this area [5]. The second general approach is to learn a cost function with respect to which observed input and state trajectories are (approximately) optimal, i.e. inverse optimal control [1]–[4], [6]–[21]. These methods have primarily focused on finite-dimensional optimization and stochastic optimal control problems.

In the context of finite parameter optimization, Keshavarz, et al. [4] develop an inverse optimization method which learns the value function of a discrete-time stochastic control system given observations. These ideas were extended to learn a cost function for a deterministic discrete-time system in Puydupin-Jamin, et al. [6], and a hybrid dynamical system in [22]. Similarly, Terekhov, et al. [7], [8] and Park, et al. [9] develop an inverse optimization method for deterministic finite-dimensional optimization problems with additive cost functions and linear constraints. Other recent work formulates an optimization problem which simultaneously learns a cost function and optimal trajectories [23], [24].

A variety of methods have been developed in the context of stochastic optimal control problems and, in particular, Markov decision processes [1], [11], [13], [15]–[17], [19], [20]. Ng and Russell [11] developed a method for stationary Markov decision processes based on linear programming. Abbeel and Ng [1] extends that work by finding a cost function with respect to which the expert's cost is less than that of predicted trajectories by a margin. A further extension simultaneously learns the system dynamics along specific trajectories of interest [13]. Ramachandran, et al. [16] takes a Bayesian approach and assumes that actions are distributed proportional to the future expected reward. The method developed in Ziebart, et al. [17] works by computing a probability distribution over all possible paths which matches features along the observed trajectory. Dvjijotham and Todorov [19] develop a method of inverse optimal control for linearly-solvable stochastic optimal control problems. Their method takes advantage of the fact that, for the class of system model they consider, the Hamilton-Jacobi-Bellman equation gives an explicit formula for the cost function once the value function is known. Aghasadeghi and Bretl [20] develop a method of inverse optimal control that uses path integrals to create a probability distribution over all possible paths. The problem is then one of maximizing the likelihood of observations.

Learning from demonstration methods are applied in three different areas. First, learning from demonstration has been applied as a method of data-driven automation [1], [2], [11]–[15], [17], [19], [25]–[27]. Tasks of interest include bipedal walking, navigation of aircraft, operation of agricultural and construction vehicles. Second, learning from demonstration methods have been applied to cognitive and neural modeling [3], [7]–[10], [18], [28]–[31]. Third, learning from demonstration methods have been applied to system identification of deformable objects [21], i.e. learning elastic stiffness parameters of objects such as surgical suture, rope, and hair.

III. Inverse Optimal Control: Problem Statement

In the rest of this paper, we consider the following class of optimal control problems

$$\begin{aligned} \underset{x,u}{\text{minimize}} \quad & \int_{t_0}^{t_f} c^T \phi[t, x(t), u(t)] dt \\ \text{subject to} \quad & \dot{x}(t) = f[t, x(t), u(t)] \\ & x(0) = x_{start} \\ & x(t_f) = x_{goal} \end{aligned} \tag{1}$$

where $x(t) \in \mathcal{X} \subset \mathbb{R}^n$ is the state, $u(t) \in \mathcal{U} \subset \mathbb{R}^m$ is the input, $\phi : \mathbb{R} \times \mathcal{X} \times \mathcal{U} \to \mathbb{R}^k_+$ are known basis functions, and $c \in \mathbb{R}^k$ is an unknown parameter vector to be learned. We assume, without loss of generality, that $\|c\| \leq 1$. We assume that the system equations

$$\dot{x}(t) = f[t, x(t), u(t)] \tag{2}$$

are well posed, that is, for every initial condition x_{start} and every admissible control $u(t)$, the system $\dot{x}(t) = f[x(t), u(t)]$ has a unique solution x on $t \in [0, t_f]$. This is satisfied, for example, when f is continuous in t and u and differentiable ($\mathcal{C}^1$) in x, f_x is continuous in t and u, and u is piecewise continuous as a function of t [32], [33]. The objective basis function ϕ is assumed to be smooth in x and u. This problem also assumes there are no input and state constraints. These constraints are often important in practice, and will be the subject of future work.

The problem of *inverse optimal control* is to infer the unknown parameters with respect to which a given trajectory, the *observation*, is a local extremal solution to problem (1). This observed trajectory is denoted by

$$(x^*, u^*) = \{x^*(t), u^*(t) : t \in [0, t_f]\}. \tag{3}$$

For convenience, we will often drop the asterisk and refer to an optimal trajectory as (x, t). We also consider observing multiple trajectories, each local minima of problem (1) for different boundary conditions. We will refer to a set D of M observations as follows

$$D = \left\{ \left(x^{*(i)}, u^{*(i)} \right) \right\} \quad \text{for } i = 1, \ldots, M \tag{4}$$

where each trajectory has boundary conditions

$$\left(x^{(i)}_{start}, x^{(i)}_{goal} \right) \quad \text{for } i = 1, \ldots, M. \tag{5}$$

An important quantity in the methods discussed in this paper is the accumulated value of the unweighted basis functions along a trajectory, which we call the *feature vector* of a trajectory $\mu(x, u)$, and define by

$$\mu(x, u) = \int_{t_0}^{t_f} \phi[t, x(t), u(t)] dt. \tag{6}$$

In practice, one would generally have sampled observations of the behavior of the system, but for the analysis in this paper, we assume we have perfect observations of the continuous-time system trajectories.

In practice, the solution of (1) is typically obtained using a numerical optimal control solver such as direct multiple shooting or collocation. In this paper, we use the pseudospectral optimal control package GPOPS [34] to numerically solve the forward problem in the prior methods which require it.

IV. Three Prior Methods

In this section we formally describe the three prior methods of inverse optimal control with which we compare the new method developed in Section V. In their original form, the method of Abbeel and Ng, and the method of Ratliff, et al. were developed in the context of Markov decision processes. The general structure and theoretical guarantees of the methods apply with slight modification to the deterministic continuous-time class of problems we consider in this paper, specified in Equation (1).

A. Method of Mombaur, et al.

The method of Mombaur, et al. [3] works by searching for the cost function parameter c which minimizes the sum-squared error between predicted and observed trajectories. This method has two main components. In the upper-level, a derivative-free optimization technique is used to search for the cost function parameter c. In the lower-level, a numerical optimal control method is used to solve the forward optimal control problem (1) for a candidate value of c. We will now discuss the two levels in detail.

The objective of the upper-level derivative-free optimization is given by the following

$$\underset{c}{\text{minimize}} \int_{t_0}^{t_f} \| [x^c(t); u^c(t)] - [x^*(t); u^*(t)] \|^2 dt \tag{7}$$

where $[x^*(t); u^*(t)]$ is the vector concatenation of the state and input of the observed trajectory at time t, and $[x^c(t); u^c(t)]$ is the solution to the forward problem (1), given the parameter vector c. Mombaur, et al. [3] discuss high performance derivative-free algorithms to minimize this upper-level objective. For our baseline analysis in this paper, however, we use the Matlab `fminsearch` implementation of the Nelder-Mead simplex algorithm. Iterations of the Nelder-Mead algorithm constitute the upper-level of this method. Upon selecting a new candidate value of c, the lower-level proceeds by solving (1) for the candidate value, generating a predicted trajectory (x^c, u^c). Given this trajectory, the upper-level objective can be evaluated, and the search for a new candidate c continues. This method is easily extended for the case where multiple trajectories are observed by considering the sum of predicted errors with respect to each observed trajectory in the upper-level objective.

B. Method of Abbeel and Ng

The method of Abbeel and Ng [1] was originally developed for infinite-horizon Markov decision processes with discounted reward. The goal of this method is to find a control policy which yield a feature vector close to that of the observed trajectory. The method is initialized by selecting a random cost function parameter vector $c^{(0)}$ and solving the forward problem (1) to obtain an initial predicted trajectory $(x^{(0)}, u^{(0)})$ and associated feature vector $\mu^{(0)}$. On the i-th iteration, solve the following quadratic program:

$$\begin{aligned} \underset{c^{(i)}, b^{(i)}}{\text{minimize}} \quad & \|c^{(i)}\|^2 \\ \text{subject to} \quad & (c^{(i)})^T \mu^* \leq (c^{(i)})^T \mu^{(j)} - b^{(i)} \\ & \qquad \text{for} \quad j = 0, \ldots, i-1 \\ & b^{(i)} > 0 \end{aligned} \tag{8}$$

where $b^{(i)}$ is the margin on the i-th iteration, and μ^* and $\mu^{(j)}$ are the feature vectors of the optimal trajectory and j-th trajectory, respectively (recall the definition given in (6)). If $b^{(i)} < \epsilon$, then terminate. Otherwise, given the result from the quadratic program, $c^{(i)}$, solve the forward optimal control problem, Equation (1), with $c = c^{(i)}$ to obtain the predicted trajectory $(x^{(i)}, u^{(i)})$ and associated feature vector $\mu^{(i)}$. Set $i = i + 1$ and repeat.

As shown in [1], this method terminates after a finite number of iterations (the theorems stated and proved in [1] carry over with minor modification to the deterministic continuous-time case, which we omit for brevity). Upon termination, this algorithm returns a set of policies Π, and there exists at least one policy in Π that yields a feature vector differing from the expert's by no more than ϵ.

C. Method of Ratliff, et al.

The maximum margin planning method of Ratliff, et al. [2] is an inverse optimal control method that learns a cost function for which the expert policy has lower expected cost than every alternative policy by a margin that scales with the *loss* of that policy. As in the method of Abbeel and Ng, we begin with the following quadratic program

$$\begin{aligned} \underset{c,b}{\text{minimize}} \quad & \|c\|^2 \\ \text{subject to} \quad & c^T\mu^* \leq c^T\mu(x,u) - b \\ & \text{for all} \quad (x,u) \in S \end{aligned}$$

where S is some set of trajectories. However, instead of considering a finite collection of trajectories, consider all possible trajectories (x,u) s.t. $\dot{x}(t) = f(t,x(t),u(t))$. Then the constraints in the quadratic program are satisfied if

$$c^T\mu^* \leq \min_{\substack{(x,u) \\ \dot{x}=f(x,u)}} \left(c^T\mu(x,u) - b\right)$$

Also, instead of trying to find the maximum fixed margin b, consider a margin which depends on the trajectory, let $b = L(x,u)$, where $L(x,u)$ denotes a loss function. This loss function typically specifies the closeness of a trajectory (x,u) to the observed trajectory (x^*,u^*), i.e. the loss function is zero near the observed trajectory and increases gradually to 1 away from the observed trajectory. Finally, slack variables ζ are introduced to allow constraint violations. The problem is now given by

$$\begin{aligned} \underset{c,\zeta}{\text{minimize}} \quad & \zeta + \frac{\lambda}{2}\|c\|^2 \\ \text{subject to} \quad & c^T\mu^* \leq \min_{\substack{(x,u) \\ \dot{x}=f(x,u)}} \left(c^T\mu(x,u) - L(x,u)\right) + \zeta \end{aligned} \tag{9}$$

where $\lambda \geq 0$ is a constant that trades off between the penalizing constraint violations and a desire for small weight vectors. Since the slack variables are tight, they can be pulled into the objective function to obtain:

$$J(c) = \lambda\|c\|^2 + c^T\mu^{*(i)} - \min_{\substack{(x,u) \\ \dot{x}=f(x,u)}} \left\{c^T\mu(x,u) - L(x,u)\right\} \tag{10}$$

This convex program can be solved using subgradient descent. As shown in [2], a subgradient $g(c)$ of $J(c)$ is given by $g(c) = \mu^* - \widehat{\mu} + \lambda c$, where $\widehat{\mu}$ represents the solution to $\arg\min_\mu \left(c^T\mu + L(\mu)\right)$, i.e. the solution to the forward optimal control problem (1) with cost function augmented by the loss function. The unknown parameter c is then updated iteratively using subgradient descent $c^{(i+1)} = c^{(i)} - \alpha_i g(c^{(i)})$.

Ratliff, et al. [2] show that for constant step size α, this method achieves linear convergence to some neighborhood of the minimum cost. In addition, they show that for diminishing step size $\alpha_j = 1/j$, the method will converge to a local minimum at a sub-linear rate.

V. A New Method Based on Necessary Conditions for Optimality

The three methods described in the previous section shared common structure. In particular, each method solves a forward optimal control problem repeatedly in an inner loop. They do this in order to compare the observed trajectory (or feature vectors) with predicted trajectories given a candidate cost function. In this section, we derive another approach inspired by recent work in inverse convex optimization by Keshavarz, et al. [4]. The key idea in our approach is that we assume that the observations are perfect measurements of the system evolution, and that the expert is only *approximately optimal*, where we define what it means to be approximately optimal below. Under this new set of assumptions, we can immediately say how optimal the agent is by looking at how well the demonstration trajectory satisfies the necessary conditions for optimal control. To do this, we use the necessary conditions to define a set of residual functions. The inverse optimal control problem is then solved by minimizing these residual functions over the unknown parameters. In the remainder of this section, we will describe these different stages in detail.

A. Inverse Optimal Control Formulation

Consider a trajectory (x,u) of the system given in Equation (2). The minimum principle gives us necessary conditions for (x,u) to be a local minimum of Eq. (1) [35], [36]. The Hamiltonian function for the problem we consider is

$$H(x,u,p) = c^T\phi(t,x,u) + p^T f(t,x,u) \tag{11}$$

For a given c, if (x^*,u^*) is optimal, the necessary conditions for optimal control state that there exists a costate trajectory $p^* : \mathbb{R} \to \mathbb{R}^n$ such that

$$0 = \dot{p}^*(t)^T + \nabla_x H(x^*(t),u^*(t),p^*(t)) \tag{12}$$
$$0 = \nabla_u H(x^*(t),u^*(t),p^*(t)) \tag{13}$$

We apply these necessary conditions to our problem (1) to obtain

$$\begin{aligned} 0 &= \dot{p}(t)^T + c^T\nabla_x\phi[t,x(t),u(t)] + p(t)^T\nabla_x f[t,x(t),u(t)] \\ 0 &= c^T\nabla_u\phi[t,x(t),u(t)] + p(t)^T\nabla_u f[t,x(t),u(t)] \end{aligned}$$

if $(x(t),u(t))$ is optimal, these conditions will be satisfied. If the trajectory is *approximately optimal*, these conditions are approximately satisfied. We formalize this by defining *residual functions* from these necessary conditions. Our method consists in minimizing the extent to which observed trajectories violate these necessary conditions, i.e. minimizing the extent to which the residual functions are not equal to zero (where we define what it means to be close to zero in Section V-B). Let

$$z(t) = \begin{bmatrix} c \\ p(t) \end{bmatrix} \in \mathbb{R}^{k+n} \qquad v(t) = \dot{p}(t) \in \mathbb{R}^n$$

The residual function $r[z(t), v(t)]$ is then defined as

$$r[z(t), v(t)] = \begin{bmatrix} \nabla_x \phi \Big|_{(x,u)}^T & \nabla_x f \Big|_{(x,u)}^T \\ \nabla_u \phi \Big|_{(x,u)}^T & \nabla_u f \Big|_{(x,u)}^T \end{bmatrix} z(t) + \begin{bmatrix} I \\ 0 \end{bmatrix} v(t)$$
$$= F(t)z(t) + G(t)v(t) \tag{14}$$

where we have just rearranged the necessary conditions. The notation $(\cdot)\big|_{(x,u)}$ is shorthand for evaluating the particular function along the trajectory given in the observation, i.e. $\nabla_x \phi \Big|_{(x,u)} \equiv \nabla_x \phi \left[t, x(t), u(t)\right]$.

This formulation can also be extended to handle multiple observations. Consider M trajectories, which may have different boundary conditions, but have the same fixed final time t_f $\left\{\left(x^{(i)}, u^{(i)}\right)\right\} \quad i = 1, \ldots, M$ with each $(x^{(i)}, u^{(i)}) = \left\{x^{(i)}(t), u^{(i)}(t) : t \in [0, t_f]\right\}$. The development is the same and the unknown parameters $z(t)$ and $v(t)$ are simply extended to include the unknown costate trajectories of the M observations. The particular structure of the residual function is such that the amount of computation grows linearly with the number of observations, just as it does with the three prior methods of inverse optimal control. We now describe how our method minimizes these residual functions over the unknown parameters $c, p(t), \dot{p}(t)$ in order to best satisfy necessary conditions for optimality.

B. Residual Optimization

To solve for the unknown parameters $z(t)$ and $v(t)$ that cause the observed trajectories to best satisfy necessary conditions for optimal control, we solve the following problem

$$\begin{aligned} \underset{z(t), v(t)}{\text{minimize}} \quad & \int_{t_0}^{t_f} \|r[z(t), v(t)]\|^2 dt \\ \text{subject to} \quad & \dot{z}(t) = \begin{bmatrix} 0 \\ I \end{bmatrix} v(t) \\ & z(0) = z_0 \qquad \text{(unknown)} \end{aligned} \tag{15}$$

where

$$\|r[z(t), v(t)]\|^2 = z^T F^T F z + v^T G^T G v + z^T F^T G v \tag{16}$$

where the argument t has been dropped for brevity. If $z(0)$ were known, this would be a standard LQR problem (with cross terms)

$$\begin{aligned} \min_{z(t), v(t)} \quad & \int_{t_0}^{t_f} \left\{z^T Q z + v^T R v + z^T S v\right\} dt \\ \text{subject to} \quad & \dot{z}(t) = Az + Bv \\ & z(0) = z_0 \end{aligned} \tag{17}$$

where

$$A(t) = 0 \qquad B(t) = \begin{bmatrix} 0 \\ I \end{bmatrix}$$
$$Q(t) = F(t)^T F(t) \qquad R(t) = G(t)^T G(t)$$
$$S(t) = F(t)^T G(t).$$

Solving this LQR problem yields the linear control policy and quadratic value function

$$v(t) = K(t)z(t) \qquad V(z_0) = z_0^T P(0) z_0$$

where

$$K(t) = -(G(t)^T G(t))^{-1} \left(G(t)^T F(t) + B(t)^T P(t)\right)$$

and where $P(t)$ represents the solution to the LQR Riccati equation. We complete our solution for $z(t)$ by solving the following problem

$$\underset{z_0}{\text{minimize}} \quad z_0^T P(0) z_0.$$

Without normalization, this quadratic program is satisfied by the trivial solution $z_0 = 0$. Normalization is performed by using prior knowledge about the problem domain. For example, when the forward optimal control problem has a quadratic cost function, one can often assume that one of the weights is equal to 1.

VI. Simulation Experiments

A. Evaluating Methods under Noise-free Observations

To evaluate the performance of the three recent inverse optimal control methods described in Section IV and the new method introduced in Section V, we perform numerical simulations in which we observe optimal trajectories of three different systems and learn the objective function for each system. For each system, we collect the optimal trajectories, i.e. noise-free observations, by simulating the system acting under the optimal control policy for particular boundary conditions and fixed terminal time. We collect simulations for 50 random boundary conditions. These experiments form a baseline of comparison, whose results can be used to understand the fundamental behavior of each method.

B. Robustness under Cost Perturbation

To evaluate the robustness of the four methods, we perform the following perturbation to the inverse optimal control problem. Up to this point we have considered the true cost function to be perfectly modeled by the weighted combination of known basis functions, as shown in Equation (1). In our perturbation simulations, we assume that the given cost basis functions are only an approximation to the true cost function. In particular, we set the true cost function to be

$$J(u) = \int_{t_0}^{t_f} c^T \phi[t, x(t), u(t)] + d^T \rho[x(t), u(t)] dt. \tag{18}$$

where $\rho : \mathcal{X} \times \mathcal{U} \to [0, 1]^l$ are perturbation basis functions and $d \in \mathbb{R}^l$ are perturbation weights such that $\|d\| < \epsilon$ for some $\epsilon > 0$. In particular, we model a general perturbation with a linear combination of k-th order multivariate Fourier basis functions. The multivariate basis functions are defined as

$$\rho_i[z(t)] = \begin{cases} 1 & \text{if } i = 0 \\ 1 + \cos\left(2\pi a^i \cdot z\right) & \text{for odd } i \\ 1 + \sin\left(2\pi a^i \cdot z\right) & \text{for even } i \end{cases} \tag{19}$$

for $i = 1, \ldots l$, where $a^i = [a_1, \ldots, a_{n+m}]$, each $a_j \in [0, \ldots, l]$. Here z is the concatenation of the state and input vectors at time t, $z(t) = [x(t), u(t)]$. A particular set of basis functions is formed by systematically varying the elements in each a^i, and assuming only one nonzero element of a^i for each i.

C. Three Example Systems

The three systems we use are (a) linear quadratic regulation, (b) regulation of a kinematic unicycle, (c) calibration of an elastic rod. We now describe the forward optimal control problem of each of these systems.

1) Linear Quadratic Regulation: In our first system, we consider a linear system with quadratic cost

$$\begin{aligned} \underset{x,u}{\text{minimize}} \quad & \int_{t_0}^{t_f} x^T Q x + u^T R u \\ \text{subject to } & \dot{x}(t) = Ax(t) + Bu(t), \\ & x(0) = x_{start} \\ & x(t_f) = \text{Free}, \end{aligned} \tag{20}$$

where states are denoted by $x(t) \in \mathbb{R}^n$ and control inputs are denoted by $u(t) \in \mathbb{R}^m$. The matrices A and B are assumed time-invariant, with elements drawn from a $N(0, 1)$ Gaussian distribution for each trial. Matrix A is scaled such that $|\lambda_{max}(A)| < 1$, and controllability of the system is verified manually. The initial state of the system x_0 for each trial is drawn from a $N(0, 5)$, and the final time $t_f = 10$ is fixed for all trials. Moreover, for each trial we select cost matrices Q and R, with diagonal elements generated according to the uniform distributions of $U[0, 1]$ and $U[\epsilon, 1]$ respectively, to obtain nonnegative-definite and positive-definite matrices Q and R.

2) Quadratic Regulation of the Kinematic Unicycle: As our second test system, we consider quadratic regulation of the kinematic unicycle

$$\begin{aligned} \underset{x,u}{\text{minimize}} \quad & \int_{t_0}^{t_f} x^T Q x + u^T R u \\ \text{subject to } & \dot{x}(t) = \begin{bmatrix} \cos x_3(t) \\ \sin x_3(t) \\ u(t) \end{bmatrix}, \\ & x(0) = x_{start} \\ & x(t_f) = \text{free}, \end{aligned} \tag{21}$$

where states are denoted by $x(t) \in \mathbb{R}^3$ (with $x_i(t)$ representing the i-th element of the vector $x(t)$), and control inputs are denoted by $u(t) \in \mathbb{R}$. We generate initial conditions and cost parameters in a similar manner to the linear quadratic regulation problem.

3) Calibrating an Elastic Rod: Our third test system considers a thin, flexible wire of fixed length that is held at each end by a robotic gripper, which we call an elastic rod [37]. Assuming that it is inextensible and of unit length, we describe the shape of this rod by a continuous map $q\colon [0, 1] \to G$, where $G = SE(3)$. As defined in [37], let L_q denote the left translation map $L_q : G \to G$. Let e denote the identity element of G, and let $\mathfrak{g} = T_e G$ and $\mathfrak{g}^* = T_e^* G$. Abbreviating $T_e L_q(\zeta) = q\zeta$ as usual for matrix Lie groups, we require this map to satisfy

$$\dot{q} = q(u_1 X_1 + u_2 X_2 + u_3 X_3 + X_4) \tag{22}$$

for some $u\colon [0, 1] \to U$, where $U = \mathbb{R}^3$ and X_i are the usual basis for $\mathfrak{g}$. We refer to q and u together as $(q, u)\colon [0, 1] \to G \times U$ or simply as (q, u). Each end of the rod is held by a robotic gripper. We ignore the structure of these grippers, and simply assume that they fix arbitrary $q(0)$ and $q(1)$. We further assume, without loss of generality, that $q(0) = I_{4\times4}$. Finally, we assume that the rod is elastic in the sense of Kirchhoff [38], so has total elastic energy

$$\frac{1}{2} \int_0^1 \left(c_1 u_1^2 + c_2 u_2^2 + c_3 u_3^2 \right) dt$$

for given constants $c_1, c_2, c_3 > 0$. For fixed endpoints, the rod will be motionless only if its shape locally minimizes the total elastic energy. In particular, we say that (q, u) is in static equilibrium if it is a local optimum of

$$\begin{aligned} \underset{q,u}{\text{minimize}} \quad & \frac{1}{2} \int_0^1 \left(c_1 u_1^2 + c_2 u_2^2 + c_3 u_3^2 \right) dt \\ \text{subject to} \quad & \dot{q} = q(u_1 X_1 + u_2 X_2 + u_3 X_3 + X_4) \\ & q(0) = e, \qquad q(1) = b \end{aligned} \tag{23}$$

for some $b \in \mathcal{B}$.

As mentioned in Section II, recent work [21] has tackled a similar problem which used a different model and a solution method analogous to the method of Mombaur, et al.

VII. Results and Discussion

A. Perfect Observations with Known Basis Functions

In this set of experiments, each algorithm was given one perfect observation of an optimal trajectory and learned the unknown cost function parameters c. After learning the cost function, predicted trajectories are computed. This allows us to compute other statistics such as the error in total cost, error in feature vectors, and sum squared error between observed and predicted trajectories.

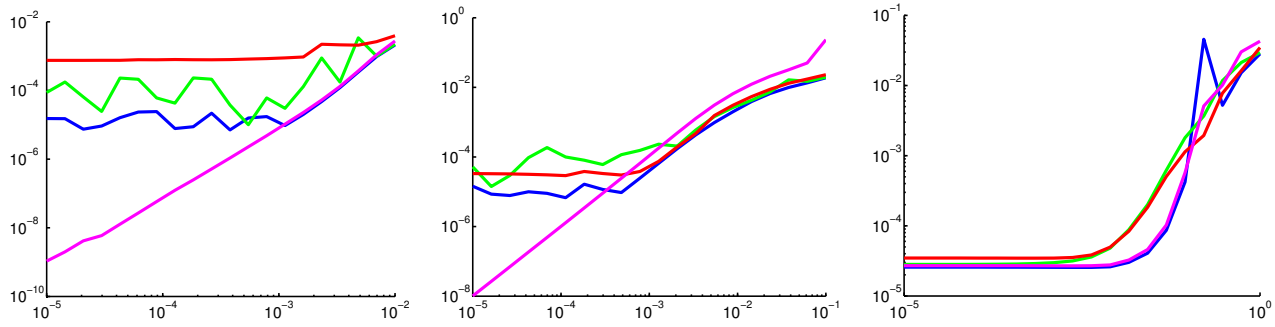
Table I shows results averaged over 50 trials with randomly selected boundary conditions in each trial. In the method of Mombaur, the sum-squared error between predicted and observed trajectories converges near zero as the number of iterations increases. However the inferred cost function parameters are not learned perfectly. Similarly, upon termination of the methods of Abbeel and Ratliff, the error between predicted and observed feature vectors is small, but the cost function parameters are not learned perfectly.

The new method developed in this paper also performs as expected – learning the unknown parameters perfectly (within the accuracy and precision tolerances of ODE and least squares solvers).

TABLE I

RESULTS FOR PERFECT OBSERVATIONS WITH KNOWN BASIS FUNCTIONS.

System	Error Type	Mombaur	Abbeel	Ratliff	New
LQR	computation (s)	280	68	117	4
	forward problems	129	28	48	0
	parameter error	7.03e-2	1.71e-1	6.99e-1	6.35e-8
	feature error	2.30e-3	3.07e-3	1.15e-1	2.81e-9
	trajectory error	1.36e-5	1.04e-4	2.64e-2	1.04e-16
Unicycle	computation (s)	448	63	280	2
	forward problems	133	20	100	0
	parameter error	3.27-2	5.12e-1	5.23e-1	2.54e-5
	feature error	3.53e-3	1.69e-2	1.42e-2	1.03e-5
	trajectory error	1.55e-5	1.12e-3	4.64e-3	8.09e-10
Elastic Rod	computation (s)	95	9	15	1
	forward problems	71	5	10	0
	parameter error	3.38e-2	8.92e-1	9.71e-1	3.96e-5
	feature error	6.77e-7	6.24e-3	4.48e-3	4.87e-7
	trajectory error	1.94e-5	7.95e-3	8.82e-3	6.14e-6



(a) LQR regulation, trajectory error. (b) Unicycle regulation, trajectory error. (c) Elastic rod calibration, trajectory error.

Fig. 1. This figure shows how the sum-squared error between observed and predicted trajectories vary under perturbations of increasing magnitude. Blue, Green, Red, and Magenta curves correspond to the methods of Mombaur, Abbeel, Ratliff, and the new technique developed in this paper, respectively.

B. *Perfect Observations with Perturbed Cost*

In this set of experiments, the true cost function consists of a linear combination of known basis functions plus a bounded deterministic perturbation (see Section VI-B). For each system, one particular set of boundary conditions was selected, and observations of optimal trajectories are gathered for a range of perturbation magnitudes. Figure 1 shows the performance of each method over varying magnitude perturbations. These results generally show:

- All of the methods learn cost functions which are able to approximate the observation in terms of feature vector and trajectory errors.
- The performance of the iterative methods remains close to the results obtained with known basis functions for small perturbations, and then degrades at larger perturbations,
- The performance of our new method (KKT) continues to improve as the perturbation descreases, reflecting exact recovery of the cost function (to specified numerical method tolerances).

Note that in the case of the elastic rod, all of the methods, including our new approach, stop improving as the perturbation magnitude gets small. This trend occurs because the numerical method for solving the forward optimal control problem terminates before reaching the observed local minima under our standard convergence and tolerance parameters, which are fixed for all experiments.

VIII. CONCLUSION

In this paper, we presented a new method of inverse optimal control, and compared the method to existing approaches using a set of canonical example systems. We compared our new approach with the following methods: inverse reinforcement learning by Abbeel and Ng [1], maximum margin planning by Ratliff, et al. [2], and inverse optimal control by Mombaur, et al. [3]. These existing solution approaches search for values of the parameters that minimize the difference between predicted and observed trajectories (or state-action features). These approaches require solving a forward optimal control problem at each iteration. The approach presented in this paper does not require the solution of any forward optimal control problem, and instead minimizes residual functions derived from first order necessary conditions for optimality. Our results show that the new method we develop is better able to recover unknown parameters and is less computationally expensive than the

existing methods. While our new method behaved well for the canonical systems used in this paper, we acknowledge that there are a variety of situations in which it is not clear which method would be best.

There are opportunities for future work, which include (a) investigating the existence and uniqueness of solutions under the new approach developed in this paper, and (b) investigating how our method performs under additional forms of model perturbation. It is not yet clear exactly what types of observations might cause the method to fail to recover the unknown cost function. Second, in this paper we considered perturbations in the true cost function, i.e. the modeled basis functions are only approximations of the true underlying cost function. There are at least two additional forms of model perturbation of interest. The first is to consider inaccurate system dynamics. Second, we will consider system trajectories which are only measured at sampled points which contain noisy partial state information.

REFERENCES

[1] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. New York, NY, USA: ACM, 2004, p. 1.

[2] N. Ratliff, J. A. D. Bagnell, and M. Zinkevich, "Maximum margin planning," in *International Conference on Machine Learning*, July 2006.

[3] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—an inverse optimal control approach," *Autonomous Robots*, vol. 28, pp. 369–383, 2010.

[4] A. Keshavarz, Y. Wang, and S. Boyd, "Imputing a convex objective function," in *IEEE International Symposium on Intelligent Control (ISIC)*, sept. 2011, pp. 613 –619.

[5] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469 – 483, 2009.

[6] A.-S. Puydupin-Jamin, M. Johnson, and T. Bretl, "A convex approach to inverse optimal control and its application to modeling human locomotion," in *IEEE International Conference on Robotics and Automation*, 2012.

[7] A. Terekhov, Y. Pesin, X. Niu, M. Latash, and V. Zatsiorsky, "An analytical approach to the problem of inverse optimization with additive objective functions: an application to human prehension," *Journal of Mathematical Biology*, vol. 61, pp. 423–453, 2010.

[8] A. Terekhov and V. Zatsiorsky, "Analytical and numerical analysis of inverse optimization problems: conditions of uniqueness and computational methods," *Biological Cybernetics*, vol. 104, pp. 75–93, 2011.

[9] J. Park, V. M. Zatsiorsky, and M. L. Latash, "Finger coordination under artificial changes in finger strength feedback: A study using analytical inverse optimization," *Journal of Motor Behavior*, vol. 43, no. 3, pp. 229–235, 2011.

[10] T. D. Nielsen and F. V. Jensen, "Learning a decision maker's utility function from (possibly) inconsistent behavior," *Artificial Intelligence*, vol. 160, pp. 53–78, 2004.

[11] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 663–670.

[12] P. Abbeel, "Apprenticeship learning and reinforcement learning with application to robotic control," Ph.D. dissertation, Stanford University, 2008.

[13] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous Helicopter Aerobatics through Apprenticeship Learning," *The International Journal of Robotics Research*, 2010.

[14] J. Tang, A. Singh, N. Goehausen, and P. Abbeel, "Parameterized maneuver learning for autonomous helicopter flight," in *IEEE International Conference on Robotics and Automation*, may. 2010, pp. 1142 –1148.

[15] U. Syed, M. Bowling, and R. E. Schapire, "Apprenticeship learning using linear programming," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1032–1039.

[16] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," in *Proceedings of the 20th international joint conference on Artifical intelligence*, ser. IJCAI'07. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007, pp. 2586–2591.

[17] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. AAAI*, 2008, pp. 1433–1438.

[18] ——, "Human behavior modeling with maximum entropy inverse optimal control," *AAAI Spring Symposium on Human Behavior Modeling*, 2009.

[19] K. Dvijotham and E. Todorov, "Inverse optimal control with linearly-solvable mdps," in *International Conference on Machine Learning*, 2010.

[20] N. Aghasadeghi and T. Bretl, "Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, sept. 2011, pp. 1561 –1566.

[21] S. Javdani, S. Tandon, J. Tang, J. O'Brien, and P. Abbeel, "Modeling and perception of deformable one-dimensional objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, may 2011, pp. 1607 –1614.

[22] N. Aghasadeghi, A. Long, and T. Bretl, "Inverse optimal control for a hybrid dynamical system with impacts," in *Robotics and Automation (ICRA), 2012 IEEE/RSJ International Conference on*. IEEE, 2012.

[23] K. Hatz, J. Schlöder, and H. Bock, "Estimating parameters in optimal control problems," *SIAM Journal on Scientific Computing*, vol. 34, no. 3, pp. A1707–A1728, 2012.

[24] M. Knauer and C. Büskens, "Bilevel optimization of container cranes," in *Progress in Industrial Mathematics at ECMI 2008*, ser. Mathematics in Industry, A. D. Fitt, J. Norbury, H. Ockendon, and E. Wilson, Eds. Springer Berlin Heidelberg, 2010, pp. 913–918.

[25] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning movement primitives," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, P. Dario and R. Chatila, Eds. Springer Berlin / Heidelberg, 2005, vol. 15, pp. 561–572.

[26] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *IEEE International Conference on Robotics and Automation*, May 2009, pp. 763 –768.

[27] G. Konidaris, S. Kuindersma, A. Barto, and R. Grupen, "Constructing skill trees for reinforcement learning agents from demonstration trajectories," in *Advances In Neural Information Processing Systems*, 2010.

[28] J. L. Yepes, I. Hwang, and M. Rotea, "New algorithms for aircraft intent inference and trajectory prediction," *Journal of Guidance Control and Dynamics*, vol. 30, pp. 370–382, 2007.

[29] D. Lee and Y. Nakamura, "Mimesis model from partial observations for a humanoid robot," *The International Journal of Robotics Research*, vol. 29, no. 1, pp. 60–80, 2010.

[30] D. H. Grollman and O. C. Jenkins, "Incremental learning of subtasks from unsegmented demonstration," in *International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010.

[31] W. Li, E. Todorov, and D. Liu, "Inverse optimality design for biological movement systems," in *IFAC*, 2011.

[32] H. K. Khalil, *Nonlinear Systems*. Prentice-Hall, Inc., 2002.

[33] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2011.

[34] A. V. Rao, D. A. Benson, C. Darby, M. A. Patterson, C. Francolin, I. Sanders, and G. T. Huntington, "Algorithm 902: Gpops, a matlab software for solving multiple-phase optimal control problems using the gauss pseudospectral method," *ACM Trans. Math. Softw.*, vol. 37, no. 2, pp. 22:1–22:39, Apr. 2010.

[35] M. Athans and P. L. Falb, *Optimal Control: An Introduction to the Theory and Its Applications*. McGraw-Hill, 1966.

[36] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control*. Hemisphere Publishing Co., 1975.

[37] T. Bretl and Z. McCarthy, "Equilibrium configurations of a kirchhoff elastic rod under quasi-static manipulation," in *WAFR*, 2012.

[38] J. Biggs, W. Holderbaum, and V. Jurdjevic, "Singularities of optimal control problems on some 6-d lie groups," *IEEE Trans. Autom. Control*, vol. 52, no. 6, pp. 1027–1038, June 2007.