2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

A Convex Approach to Inverse Optimal Control and its Application to Modeling Human Locomotion

Anne-Sophie Puydupin-Jamin, Miles Johnson and Timothy Bretl

***Abstract*—Inverse optimal control is the problem of computing a cost function that would have resulted in an observed sequence of decisions. The standard formulation of this problem assumes that decisions are optimal and tries to minimize the difference between what was observed and what would have been observed given a candidate cost function. We assume instead that decisions are only approximately optimal and try to minimize the extent to which observed decisions violate first-order necessary conditions for optimality. For a discrete-time optimal control system with a cost function that is a linear combination of known basis functions, this formulation leads to an efficient method of solution as an unconstrained least-squares problem. We apply this approach to both simulated and experimental data to obtain a simple model of human walking trajectories. This model might subsequently be used either for control of a humanoid robot or for predicting human motion when moving a robot through crowded areas.**

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. This inverse problem has been a topic of study for more than half a century [1]–[4], and has a variety of applications both inside and outside the field of robotics. Ziebart et al [5] apply inverse optimal control to explain why taxi drivers make specific route choices, based on GPS observations. Nielsen and Jensen [6] learn the utility of a human decision maker given inconsistent observations of behavior. Yepes et al [7] predict trajectories flown by human pilots in air traffic control. Trautman and Krause [8] focus on predicting future trajectories taken by human walkers to enable robot navigation in crowds. Work on related problems like apprenticeship learning [9]–[11] has been applied to learn control policies for aerobatic flight based on observations of human experts.

The standard formulation of inverse optimal control assumes that decisions are optimal and tries to minimize the difference between what is observed and what would have been observed given a candidate cost function. This approach is used for example in [12] and [13], two seminal recent papers that have largely motivated our own work. In these works, the cost function is represented as a linear combination of basis functions weighted by an unknown parameter vector. Their solution approach is to infer the parameter vector, solve the corresponding optimal control problem, predict what the resulting observations would be, and then apply derivative-free optimization to minimize the difference between predicted and observed trajectories. This approach can work well, but is computationally very expensive because it requires solving an optimal control problem at each iteration. One approach that differs from this standard formulation is presented in [14]. In this recent paper, Dvijotham and Todorov implement several algorithms that do not require solving the forward problem. Instead, they infer the value function using maximum likelihood. The inverse optimal control problem becomes an unconstrained optimization problem, and their algorithms can be applicable to linearly solvable MDPs with discrete and continuous states.

The method proposed in this paper is inspired from [15]. This new formulation of inverse optimal control assumes that the observations are perfect, while the system is considered to be only approximately optimal. This change in assumption allows us to define residual functions based on the Karush-Kuhn-Tucker (KKT) necessary conditions for optimality [16], [17]. The inverse optimal control problem then simplifies to minimizing these residual functions in order to recover the parameters that govern the cost function. As a result, the inverse optimal control problem reduces to a simple least-squares minimization, which can be solved very efficiently. We also note that it is unnecessary that the underlying optimal control problem be convex. In [15], the authors restrict their attention to convex optimization problems only, but in this paper, we apply a similar approach to solve an optimal control problem that is non-convex. Our approach can be extended to a wide range of discrete-time nonlinear problems, with the assumption that the unknown parameter vector needs to enter the cost function linearly. Note that this technique of approximating a cost function using linear combinations of basis functions is common to most inverse optimal control methods.

Our paper proceeds as follows. In Section II we will introduce the approach used to solve inverse optimal control problems. In Section III we will apply the approach to data generated in simulation, in order to validate it. In Section IV we will apply the approach to experimental data to model human locomotion, and finally Section V will present the conclusions and future works.

II. THE APPROACH

A. Problem Formulation

Consider the following optimization problem with equality constraints:

The authors are with the Department of Aerospace Engineering at the University of Illinois at Urbana Champaign, Urbana, IL, 61801, USA. {puydupi2,mjohns5,tbretl}@illinois.edu

$$\begin{aligned} \min_{x} \quad & f(x,c) = \alpha(x) + c^T\beta(x) \\ \text{subject to} \quad & g_i(x) = 0, \qquad i = 1,...,m \end{aligned} \tag{1}$$

where $x \in R^n$ is the variable, f is the objective function composed of the basis functions α and β such that $\alpha(x)\colon R^n \to R$ and $\beta(x)\colon R^n \to R^k$, g is the set of m equality constraints such that $g(x)\colon R^n \to R^m$, and $c \in R^k$ is an unknown parameter vector. The functions f and g_i are continuously differentiable. Recall that the assumption in this approach is that the observations of x are perfect while the system itself might be imperfect, e.g. $x \sim \mathcal{N}(x^*, \Sigma)$. Note that inequality constraints could be added to problem (1) but are not considered here.

The inverse optimization problem associated with (1) is to recover the unknown parameter vector c given observations of x, and having prior knowledge of α, β and g.

B. Necessary Conditions for Optimality

For a given c, assuming that x^* is a local minimum of the problem (1) and is regular, there exist unique Lagrange multiplier vectors $\lambda^* \in R^m$ such that [17], [18]:

$$\begin{aligned} \nabla_x f(x^*,c) + \sum_{i=1}^{m} \lambda_i^{*T} \nabla_x g_i(x^*) = 0 \\ g_i(x^*) = 0, \qquad i = 1,...,m \end{aligned} \tag{2}$$

where f and g_i are continuously differentiable functions. The two equations in (2) are known as the KKT necessary (and sufficient) conditions for equality constraint optimization problems. The first one is the stationarity condition while the second equation ensures primal feasibility. If the Lagrangian of problem (1) is defined to be:

$$L(x,c,\lambda) = f(x,c) + \sum_{i=1}^{m} \lambda_i^T g_i(x) \tag{3}$$

then for a given c, the necessary conditions in (2) can be rewritten as:

$$\nabla_{(x,\lambda)} L(x^*,c,\lambda^*) = 0 \tag{4}$$

C. Residual Functions

As stated previously, the system is assumed to be only approximately optimal. Residual functions are defined in order to represent what *approximately optimal* means in a manner similar to [15]:

$$\begin{aligned} r_{eq} &= g(x) \\ r(x,c,\lambda) &= \nabla_x f(x,c) + \sum_{i=1}^{m} \lambda_i \nabla_x g_i(x) \end{aligned} \tag{5}$$

The necessary conditions of optimality are satisfied when the two residual functions in (5) are equal to zero. The method consists then in minimizing the extent to which observed decisions violate the KKT necessary conditions, i.e. minimizing the extent to which they are not equal to zero. The candidate solutions are obtained from the stationarity residual function, while the first residual function is used to check that the solutions obtained are feasible.

Using the Lagrangian defined as above, the stationarity residual function in (5) becomes:

$$r(x,c,\lambda) = \nabla_x L(x,c,\lambda) \tag{6}$$

which for problem (1) corresponds to:

$$r(x,c,\lambda) = \nabla_x \alpha(x) + c^T \nabla_x \beta(x) + \lambda^T \nabla_x g(x) \tag{7}$$

Given observations of x that are assumed to be perfect, the inverse optimization problem becomes to minimize the residual function defined in (7) where the unknowns are the Lagrangian multipliers λ and the parameter c:

$$\min_{c,\lambda} \qquad ||r(x,c,\lambda)||^2 \tag{8}$$

One can see that the initial constrained optimization problem in (1) has been modified into an unconstrained optimization problem in (8), with the only limitation that the objective function needs to be composed of a linear combination of known basis functions. Note also that the residual function shown in (7) is linear with respect to the unknown parameter c and the Lagrange multipliers λ. The problem therefore becomes a convex unconstrained least-squares optimization problem, which is easier to solve than the initial constrained optimization problem in (1).

It is important to note that this method can be extended to several variations of the problem presented in (8). For example, one could consider the case where multiple observations of the same system are given, or one could also consider the case where the unknown parameter c changes with time. We believe our approach can be used to solve discrete-nonlinear problems that consist of a continuously differentiable cost function and continuously differentiable constraints, but with the limitation that the unknown parameter vector needs to enter the cost function linearly.

Least-squares problems arise in many areas of applications, and are one of the most commonly encountered unconstrained optimization problems. Their structure makes them easier to solve than general unconstrained minimization problems, and solution techniques can be found in [3] and [16]. In particular, linear least-squares problems can be solved very easily by rewriting the objective function in the following manner:

$$r(x,z) = \frac{1}{2}||Jz - b||^2 \tag{9}$$

where z is the vector of unknown parameters: $z = [c \ \ \lambda]^T$. Since the problem is convex, we know that a global minimizer z^* of $r(x,z)$ must satisfy $\nabla_x r(x,z^*) = 0$, which leads to the *normal equations*:

$$J^T J z^* = J^T b \tag{10}$$

Several algorithms can be used to solve the normal equations, the major ones being Cholesky factorization, QR factorization and Singular-Value Decomposition (SVD). These algorithms are easy to implement, and their computation time is negligible in comparison to algorithms that need to solve the forward optimal control problem at each iteration.

III. APPLICATION TO DATA GENERATED IN SIMULATION

In this section we apply the approach introduced in Section II to data generated in simulation in order to validate it. The model used for the simulation is chosen to be the unicycle model, since it is the one used in Section IV to model human locomotion using experimental data.

A. The Unicycle Model

Consider the following discretized version of the unicycle model:

$$\begin{aligned} \min_{x^i,u^i} \quad & \frac{1}{2}\tau \sum_{i=0}^{N-1} (c(u_1^i)^2 + (u_2^i)^2) \\ \text{subject to} \quad & x_1^{(i+1)} = x_1^i + \tau u_1^i \cos(x_3^i) \\ & x_2^{(i+1)} = x_2^i + \tau u_1^i \sin(x_3^i) \\ & x_3^{(i+1)} = x_3^i + \tau u_2^i \\ & x^{(0)} = x_{start} \\ & x^{(N-1)} = x_{goal} \end{aligned} \tag{11}$$

where τ is the discretization rate, and i is the time step, going from 0 to $N-1$. The position and the orientation are (x_1, x_2) and x_3, respectively. The two inputs are the forward speed u_1 and the turning rate u_2. The start and end points are set fixed to match the experimental setup described in Section IV. The unknown parameter c governs how much we penalize control effort u_1 relative to control effort u_2. The inverse optimal control problem consists in recovering the value of the parameter c, using data generated in simulation. Note that the fixed end-point constraints do not appear in the derivation of residual function, and the implications are discussed in Section IV-C.

B. Derivation of the Residual Function for the Unicycle Model

Based on (3), the Lagrangian associated with the unicycle model in (11) is:

$$\begin{aligned} L(x^i, u^i, c, \lambda^{i+1}) = & \frac{1}{2}\tau \sum_{i=0}^{N-1} (c(u_1^i)^2 + (u_2^i)^2) \\ & + \sum_{i=0}^{N-1} (g^i(x^i, u^i) - x^{i+1})^T \lambda^{i+1} \end{aligned} \tag{12}$$

where g is the set of equality constraints in (11).

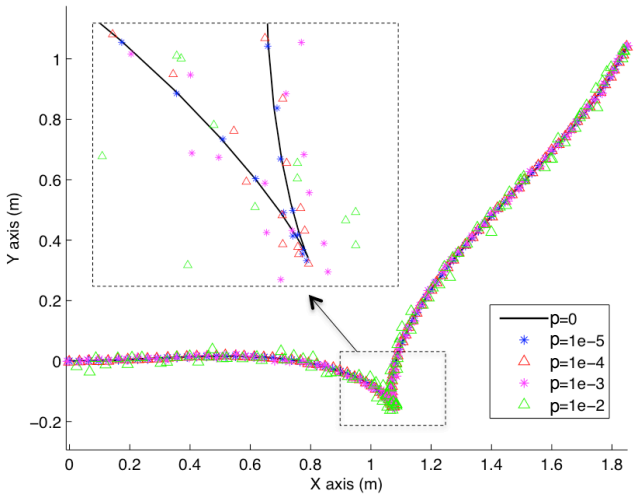

Fig. 1. Example of a noiseless trajectory simulated using the unicycle model (black line), along with four different noisy measurements. Each set of measurements has been obtained by averaging 60 noisy trajectory measurements. The amount of noise varies and is characterized by the value of the standard deviation p = $1e^{-5}$, $1e^{-4}$, $1e^{-3}$ and $1e^{-2}$.

The KKT conditions defined in (2) and (4) for the unicycle model lead to:

$$\begin{aligned} \nabla_{(x^i)} L(x^i, u^i, \lambda^i, \lambda^{i+1}) &= -\lambda^i + \nabla_{(x^i)} g^i(x^i, u^i)^T \lambda^{i+1} \\ &= 0 \end{aligned} \tag{13}$$

and

$$\begin{aligned} \nabla_{(u^i)} L(x^i, u^i, c, \lambda^{i+1}) =& \frac{1}{2}\tau \nabla_{(u^i)} (c(u_1^i)^2 + (u_2^i)^2) \\ & + \nabla_{(u^i)} g^i(x^i, u^i) \lambda^{i+1} \\ =& 0 \end{aligned} \tag{14}$$

The residual function defined in (6) then becomes:

$$\begin{aligned} r(x^i, u^i, c, \lambda^i, \lambda^{i+1}) &= \begin{bmatrix} \nabla_{(x^i)} L(x, u, \lambda^{i+1}) \\ \nabla_{(u^i)} L(x, u, \lambda^{i+1}) \end{bmatrix} \\ &= \begin{bmatrix} -\lambda^i + \nabla_{(x^i)} g^i(x^i, u^i)^T \lambda^{i+1} \\ \frac{1}{2}\tau \nabla_{(u^i)} (c(u_1^i)^2 + (u_2^i)^2) + \nabla_{(u^i)} g^i(x^i, u^i) \lambda^{i+1} \end{bmatrix} \end{aligned} \tag{15}$$

and is linear as a function of the unknown parameters $[c \;\; \lambda^i \;\; \lambda^{i+1}]^T$. It can therefore be rewritten in a similar manner to (9) and solved using the algorithms for linear least-squares unconstrained problems presented in [16]. Note that the residual vector has dimensions $(5N, 1)$, where N is the total number of time steps, since $x \in R^3$ and $u \in R^2$.

C. Results for Data Generated in Simulation

Matlab 7.11.0 is used to implement the algorithm presented in this paper. Paths are simulated using the unicycle model described in (11) by arbitrarily choosing a value for the parameter c (c = 3.7 was chosen). Once the trajectories are generated, we can use the observations of the states

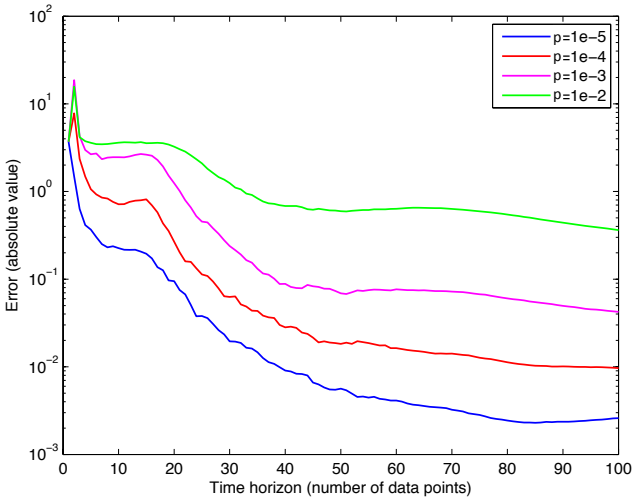

Fig. 2. Error between the predicted value of the parameter c obtained for each of the four averaged noisy measurements, and the value used during simulation. The approach used is able to predict the correct value of c ($c = 3.7$), even when very noisy measurements are used (i.e. for $p = 1e^{-2}$).

x and control inputs u to solve the unconstrained least-squares optimization problem defined in (8) where the residual function is given by (15). Recall that the aim of solving the inverse optimal control problem is to recover the value of the unknown parameter c. Since the trajectories are obtained from simulation, we can compare the value of c predicted using the approach introduced, to the actual value of the parameter used during the simulation.

To test the robustness of the approach, noise is added to the measurements. Gaussian white noise normally distributed with zero mean and variable standard deviation is added to the states x such that: $x \sim \mathcal{N}(0, p)$. The noise variable p is chosen to take the following values: $1e^{-5}$, $1e^{-4}$, $1e^{-3}$ and $1e^{-2}$. It is important to note that since the unknown parameter is assumed to be constant (non-time-varying) in our model, we do not need to consider multiple trajectories to predict the correct value of c. Instead, we are looking at how fast the predicted value of c converges as a function of the number of data points constituting the trajectory, i.e. the time horizon of the trajectory. A total of 60 trajectory measurements are generated for each values of p, which are then averaged. In other words, for each value of noise standard deviation p, we have one corresponding averaged set of measurements. Fig. 1 shows a simulated noiseless trajectory, along with the four different averaged noisy measurements obtained from the four values of p.

The inverse optimal control algorithm is used to recover the value of c for each set of measurements. The computation time required to run the algorithm is 0.032 seconds (on a 2.5 GHz Core 2 Duo processor). The results showing the comparison between the predicted value of c and the value used during simulation are shown in Fig. 2.

The results in Fig. 2 show that even with large Gaussian white noise (i.e. $p = 1e^{-2}$) the algorithm is able to predict the correct value of c. It is obvious from Fig. 2 that when the amount of white noise is increased, the algorithm necessitates more points from the trajectory (i.e. a longer time horizon) to correctly recover the unknown parameter. Therefore results in Fig. 2 show that our approach also performs well in presence of noisy measurements.

IV. APPLICATION TO EXPERIMENTAL DATA FOR MODELING OF HUMAN LOCOMOTION

In this section we apply the approach introduced in Section II to experimental data, in order to find a model for human locomotion. The reason why we would want to model human locomotion, in the context of humanoid robot control, is to obtain a model based on observations that can be implemented to humanoid robots to generate locomotion trajectories similar to human trajectories, and to predict actions or trajectories of robots based on human observations.

A. The Experiment

The data used were collected for the experiment described in [12]. In summary, subjects were asked to walk in a gymnasium from a starting point to a final destination represented by a porch. The starting point was always the same, but the final position and final orientation of the porch were varying. The subjects were asked to walk from one point to another freely, without time or velocity constraints, and the trajectories were recorded using motion capture technology. An example of a subset of 6 observed trajectories is presented in Fig. 4 (blue lines) for one subject.

B. Notes on the Choice of the Model

The initial assumption in [12] was that human walking data can be modeled using a simple nonholonomic unicycle model along with an objective function that minimizes input energy. However, it was found that more complicated dynamic models could be used to better fit the experimental data. For example, [12] defines the turning input to be the derivative of the curvature. In [13], the authors not only use a more complex dynamic model, but they also define a more complex objective function that takes into account the initial tendency of the subjects to adjust the orientation of their bodies towards the target.

However, in this paper, we want to focus on validating our inverse optimal control approach by showing results for the standard unicycle model and with a simple objective function defined in (11). As we will show in the rest of this section, our approach leads to promising results. We expect our fitting results to improve in future work when we will consider more complex system dynamics and objective functions.

C. Recovered Trajectories

Once the inverse optimal control problem is solved and the unknown parameter c is predicted, it is necessary to generate the trajectory obtained using the model and the recovered value of c, and compare it to the experimental data. Instead of using a

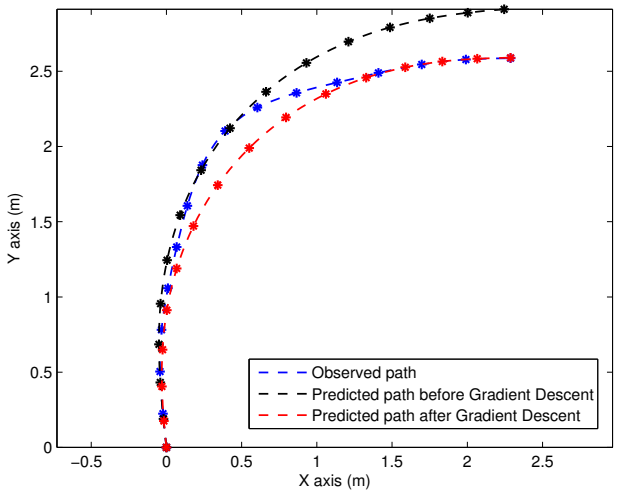

Fig. 3. Recovered trajectories before and after performing gradient descent to match the end points. The blue trajectory is the observed one obtained from experimental data. The black trajectory is the predicted result obtained before performing gradient descent and the red trajectory is the predicted result after performing gradient descent.

shooting method to generate the recovered trajectories, a faster algorithm is employed, based on the same KKT conditions for optimality. From (13) and (14), the following relationships can be derived:

$$\begin{aligned}(\lambda^i)^T &= (\lambda^{i+1})^T J^i \\ (u^i)^T &= -(\lambda^{i+1})^T G^i\end{aligned} \tag{16}$$

where for the unicycle model:

$$J^i = \begin{bmatrix} 1 & 0 & -\tau u_1^i \sin x_3^i \\ 0 & 1 & \tau u_1^i \cos x_3^i \\ 0 & 0 & 1 \end{bmatrix}, \quad G^i = \begin{bmatrix} \frac{1}{c}\cos x_3^i & 0 \\ \frac{1}{c}\sin x_3^i & 0 \\ 0 & 1 \end{bmatrix} \tag{17}$$

which leads to:

$$\begin{aligned}\lambda^{i+1} &= (J^i)^{-T} p^i \\ (u^i)^T &= -(\lambda^i)^T (J^i)^{-1} G^i\end{aligned} \tag{18}$$

where we note that $(J^i)^{-1}$ always exists. Therefore using the state equations from (11) and the set of equations from (18), it is possible to reconstruct the entire trajectory. The recovered trajectory depends only on the initial position $x^{(0)}$, which is given, and the initial set of Lagrange multipliers $\lambda^{(0)}$, which is predicted from the inverse optimal control problem. The initial recovered trajectory obtained from $\lambda^{(0)}$ and $x^{(0)}$ does not necessarily satisfy the terminal constraint. This is because the end-point constraints are not taken into account in our residual function. Therefore, we use gradient descent to search for a $\lambda^{(0)}$ that results in a trajectory satisfying the terminal constraint.

D. Results for the Experimental Data

The unconstrained linear least-squares problem being solved is defined in (8) where the residual function is given by (15) for the unicycle model presented in (11). Once the least-squares problem is solved and the unknown parameter c is predicted along with the Lagrange multipliers λ, the set of equations in (18) and (11) are used to construct the predicted trajectory.

Figure 3 shows the recovered trajectory before and after gradient descent. The total computation time to solve the inverse optimal control problem, construct the predicted trajectory and perform gradient descent is 0.076 seconds. One can see from Fig. 3 that the recovered trajectory does not perfectly fit the experimental data, and this is believed to be due to the limitations of the unicycle model.

Beyond considering only one trajectory, the experimental data consists of a set of multiple trajectories for a variety of boundary conditions. In particular, we consider a subset of the data which consists of 15 trajectories for 6 different boundary conditions (i.e. the same terminal position and orientation). Figure 4 shows results of recovered trajectories when using only one observed trajectory (dashed red lines) and when using multiple observed trajectories (dashed black lines). One can see that better results are obtained when using multiple observations to recover the parameter c. Note that in the case of multiple observations, the linear least-squares problem becomes:

$$\min_{c,\lambda^1,\ldots,\lambda^{N_{traj}}} \quad \sum_{i=1}^{N_{traj}} ||r(x^i, u^i, c, \lambda^i, \lambda^{i+1})||^2 \tag{19}$$

where $x^1, x^2, ..., x^{N_{traj}}$ are N_{traj} perfectly observed trajectories and c is constant across the N_{traj} trajectories. Our results suggest that more observations used to predict c will result in a better recovered cost function, i.e. one which better predicts observed trajectories.

The total computation time to solve the inverse optimal control problem using 15 observed trajectories, construct the predicted trajectory and perform gradient descent was 16 seconds. This increase in computation time is due to the fact that the size of the residual vector is no longer $(5N, 1)$, but $(5N_{traj}N)$, where N_{traj} is the number of observed trajectories. Despite the increased computation time, we believe the complexity of solving least-squares problem is less than the complexity of iterative techniques which require solving the forward optimal control problem at each iteration. A full comparison of running time and complexity is the subject of future work.

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper we present a new approach for solving inverse optimal control problems that differs from already existing methods. The standard formulation of this problem assumes that decisions are optimal and consists in minimizing the difference between what is predicted and what is observed given a candidate cost function. We assume instead that decisions are only approximately optimal and we minimize the extent to which observed decisions violate the necessary conditions of optimality. This approach can be applied to

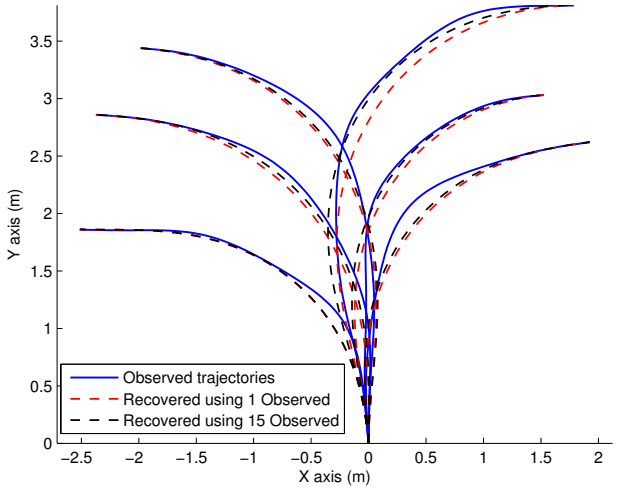

Fig. 4. In this figure, observed and predicted trajectories are projected on the x-y plane. Blue curves represent observed trajectories obtained from experimental data. Red dashed curves show predicted results obtained when using only a single observation to recover the value of parameter c. Black dashed curves show predicted results obtained when using 15 observations to recover the value of c.

discrete-time optimal control systems with the limitation that the cost function has to be a linear combination of known basis functions. The initial optimal control problem simplifies into an unconstrained linear least-squares problem, which can be solved easily and which does not necessitate iterations.

A discretized unicycle model is used to test the approach with both simulation and experimental data. The application to data generated in simulation successfully validates the approach, and shows that it also performs well in presence of noisy measurements. The approach is then applied to experimental data in order to find a model for human locomotion. Subjects were asked to walk in a gymnasium from a starting point to a final destination and their trajectories were recorded [12]. The approach gives satisfactory results when comparing the recovered trajectories to the experimental data. Furthermore, using multiple observed trajectories to recover the unknown parameter of the cost function is found to improve the quality of the recovered trajectories.

Therefore, it was shown that this approach can be used to obtain a simple model of human locomotion, without solving the forward optimal control problem and without iterations, but by simply minimizing the extent to which observed trajectories violate the necessary conditions of optimality.

B. Future Works

The details of the limitations of our approach still need to clarified. For example, it is not clear what the conditions are for perfectly recovering the cost function, and how sensitive the approach is to model uncertainty. Also, we are aware that better models for human locomotion have been proposed in other studies such as [12] and [13], and an improvement would be to use a more complex dynamic model along with a more complete set of basis functions to define the cost function. Another necessary improvement to this study is to compare our approach to existing methods, in terms of computation time and complexity. And finally, we are hoping to extend this work either for the control of a humanoid robot or for predicting of human motion when moving a robot though crowded areas.

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge Gustavo Arechavaleta, Jean-Paul Laumond, Halim Hicheur and Alain Berthoz for providing the human locomotion database that they used in [12]. This material is based upon work supported by the National Science Foundation under CPS-0931871 and CMMI-0956362.

REFERENCES

[1] R. Kalman, "When is a linear control system optimal?" *ASME Transactions, Journal of Basic Engineering*, vol. 86, pp. 51–60, 1964.

[2] M. Masak, "An inverse problem on decoupling optimal control systems," *IEEE Trans. Autom. Control*, vol. 13, no. 1, pp. 109 – 110, feb 1968.

[3] A. Jameson and E. Kreindler, "Inverse problem of linear optimal control," *SIAM Journal on Control*, vol. 11, no. 1, pp. 1–19, 1973. [Online]. Available: http://link.aip.org/link/?SJC/11/1/1

[4] J. Casti, "On the general inverse problem of optimal control theory," *Journal of Optimization Theory and Applications*, vol. 32, no. 4, pp. 491–497, 12 1980. [Online]. Available: http://dx.doi.org/10.1007/BF00934036

[5] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Human behavior modeling with maximum entropy inverse optimal control," *AAAI Spring Symposium on Human Behavior Modeling*, 2009.

[6] 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.

[7] 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.

[8] A. Trautman, P.; Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference*, pp. 797–803, 2010.

[9] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Twenty-first International Conference on Machine Learning*, 2004.

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

[11] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, 2010.

[12] G. Arechavaleta, J. P. Laumond, H. Hicheur, and A. Berthoz, "An optimality principle governing human walking," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 5–14, Feb. 2008.

[13] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—an inverse optimal control approach," *Autonomous Robots*, vol. 28, no. 3, pp. 369–383, 04 2010. [Online]. Available: http://dx.doi.org/10.1007/s10514-009-9170-7

[14] E. T. K. Dvijotham, "Inverse optimal control with linearly-solvable mdps," *Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel*, 2010.

[15] A. Keshavarz, Y. Wang, , and S. Boyd, "Imputing a convex objective function," *Proceedings IEEE Multi-Conference on Systems and Control*, To appear September 2011.

[16] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer Verlag, 2006.

[17] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*, 3rd ed., ser. International series in operations research & management science. New York: Springer, 2007, vol. 116.

[18] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.