UNIVERSITY OF CALIFORNIA, SAN DIEGO

A Computational Approach to the Study of Social Interaction

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Paul Lundy Ruvolo

Committee in charge:

Garrison W. Cottrell, Chair
Serge Belongie
Christine Harris
Javier R. Movellan
Lawrence Saul
Terrance J. Sejnowski

2012

The dissertation of Paul Lundy Ruvolo is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

_____

Chair

University of California, San Diego

2012

DEDICATION

To my parents Stephen and Francine, my sister Julia, and my girlfriend Kimberly Ferguson.

EPIGRAPH

Where, then, should we look for a satisfactory theory of behavior? Intentional theory is vacuous as psychology because it presupposes and does not explain rationality or intelligence. The apparent successes of Skinnerian behaviorism, however, rely on hidden Intentional predictions. Skinner is right in recognizing that Intentionality can be no foundation for psychology, and right also to look for purely mechanistic regularities in the activities of his subjects, but there is little reason to suppose they will lie on the surface in gross behavior (except, as we have seen, when we put an artificial straitjacket on an Intentional regularity). Rather, we will find whatever mechanistic regularities there are in the functioning of internal systems whose design approaches the optimal (relative to some ends). In seeking knowledge of internal design our most promising tactic is to takeout intelligence-loans, endow peripheral and internal events with content, and then look for mechanisms that will function appropriately with such "messages" so we can pay back the loans.

–Daniel Dennett, "Intentional Systems", Journal of Philosophy 1971.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost I want to thank my advisor Javier Movellan for continuously pushing me, through his example and through his mentorship, to become the best scientist that I can be. I want to thank Ian Fasel for invaluable mentorship as I began the journey through graduate school. I thank Daniel Messinger for opening my eyes to the world of developmental psychology. I would like to thank Gary Cottrell for recruiting me to UCSD on an IGERT trainee grant to study learning and visions in humans and machines. I would like to thank my committee members for valuable input during the dissertation process.

In addition, I wish to thank my other mentors in the Machine Perception Laboratory (MPLab): Marian Bartlett and Gwen Littlewort-Ford. The MPLab was a wonderful intellectual environment for new ideas to develop and flourish. I want to thank the following MPLab members, that I have not thanked already, for helping to create this stimulating environment: Nicholas Butko, Dave Deriso, He Huang, Kuen-han Lin, Quentin Quarles, Josh Susskind, Walter Talbott, Esra Vural, Jacob Whitehill, Tingfan Wu, and Yu-Hsin Yang.

I wish to thank my coauthors outside of my home lab, including: Andrea Chiba, Alan Fogel, Galit Hofree, Whitney Mattson, Adrienne Moore, Megan O'Rorke, and Piotr Winkielman. In addition to these formal collaborations, I have been lucky enough to be a part of some wonderful scientific groups in particular the Sequential Analysis Weekly Reading Group which helped shape the behavioral studies presented in this dissertation.

Much of the interdisciplinary work I have undertaken during the last several years would not have been possible without the NSF Temporal Dynamics of Learning Center. Through the center, I have forged new collaborations outside of my home discipline of computer science and received invaluable feedback that made existing projects better.

I would like to acknowledge the organizations that funded me during my time at UCSD: the Temporal Dynamics of Learning Center NSF Grant #SBE-0542013, NSF IGERT Grant #DGE-033345, the NSF Project One Grant #SBE-0808767, and the UC Discovery Grant UC dig03-10158.

Without the friendship of many people I wouldn't have made it through the Ph.D. process. In particular I wish to thank my close friends Stephen Checkoway, Brian McFee, Michael Stepp, Cynthia Taylor, and Matthew Tong.

Last but not least, I want to thank my parents Stephen and Francine, my sister Julia, and my girlfriend Kimberly Ferguson for unwavering support during this process.

The text of Chapter 3 is unpublished work to be submitted with authors P. Ruvolo, T. Wu, D. Messinger, A. Fogel, and J.R. Movellan. I was the primary author and researcher on this project, constructing the models, analyzing the data, and drafting the manuscript. Fogel and Messinger collected the dataset used for the analysis presented in this chapter. Wu created the software and hardware infrastructure necessary for running the human robot interaction experiment. Movellan supervised the research presented in this chapter.

The text of Chapter 4 is unpublished work to be submitted as two separate manuscripts. The first manuscript will include the research presented in Sections 4.1-4.13 with authors P. Ruvolo and J.R. Movellan. I was the primary author and researcher on the work contained in these sections, providing the mathematical derivations and drafting the manuscript. Movellan supervised the research in these sections. The second manuscript will include the research in Section 4.14 with authors P. Ruvolo, T. Wu, W. Mattson, D. Messinger, and J.R. Movellan. I was the primary author and researcher on this project, deriving the models, performing the analysis, helping to design the motion capture setup, and drafting the manuscript. Mattson collected the dataset and helped design the motion capture setup used in the experiment. Wu helped with design of the motion capture setup. Messinger and Movellan supervised this research.

VITA

2003	B. S. in Computer Science *Highest Honors*, Harvey Mudd College, Claremont, CA
2008	M. S. Computer Science, UC San Diego, La Jolla, CA
2012	Ph. D. Computer Science, UC San Diego, La Jolla, CA

PUBLICATIONS

P. Ruvolo, I. Fasel, and J. Movellan. Auditory mood detection for social and educational robots. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 3551–3556. IEEE, 2008.

P. Ruvolo, I. Fasel, and J. Movellan. Optimization on a budget: A reinforcement learning approach. *Advances in Neural Information Processing Systems*, 2009.

P. Ruvolo, I. Fasel, and J. Movellan. Tuning optimizers for time-constrained problems using reinforcement learning. *Proceedings of the Workshop on Optimization for Machine Learning Neural Information Processing Systems*, 2009.

P. Ruvolo, I. Fasel, and J.R. Movellan. A learning approach to hierarchical feature selection and aggregation for audio classification. *Pattern Recognition Letters*, 2010.

P. Ruvolo and J Movellan. Auditory cry detection in early childhood education settings. In *Proceedings of IEEE International Conference on Development and Learning*, 2008.

P. Ruvolo and J.R. Movellan. An alternative to low-level-synchrony-based methods for speech detection. *Advances in Neural Information Processing Systems*, 2010.

P Ruvolo, J Whitehill, and J. Movellan. Building a more effective teaching robot via apprenticeship learning. In *Proceedings of IEEE International Conference on Development and Learning*, 2008.

P. Ruvolo, J. Whitehill, and J.R. Movellan. Exploiting structure in crowdsourcing tasks via latent factor models. Machine Perception Laboratory Technical Report, 2010.

J. Artigas, W. Mattson, D. Messinger, P. Ruvolo, T. Wu, and J. Movellan. Rethinking motor development and learning. *Proceedings of IEEE International Conference on Development and Learning*, 2011.

I. Fasel, P. Ruvolo, T. Wu, and J. Movellan. Infomax control for social robots. In *NIPS Workshop on Probabilistic Approaches for Robotics and Control*, 2009.

H. Finger, S.C. Liu, P. Ruvolo, and J.R. Movellan. Approaches and databases for online calibration of binaural sound localization for robotic heads. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4340–4345. IEEE, 2010.

G. Littlewort, M. Bartlett, J. Whitehill, T. Wu, N. Butko, P. Ruvolo, and J. Movellan. The motion in emotion – a CERT based approach to the FERA emotion challenge. *Proceedings of FERA Workshop at Face and Gesture*, 2011.

D.M. Messinger, N.V. Ekas, P. Ruvolo, and A.D. Fogel. "Are you interested, baby?" Young infants exhibit stable patterns of attention during interaction. *Infancy*, 2011.

D.M. Messinger, P. Ruvolo, N.V. Ekas, and A. Fogel. Applying machine learning to infant interaction: The development is in the details. *Neural Networks*, 23(8-9):1004–1016, 2010.

J.R. Movellan, F. Tanaka, I.R. Fasel, C. Taylor, P. Ruvolo, and M. Eckhardt. The RUBI project: a progress report. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 333–339. ACM, 2007.

J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems*, 22:2035–2043, 2009.

T. Wu, N. Butko, P. Ruvolo, J. Whitehill, M. Bartlett, and J. Movellan. Between dataset AU recognition transfer. *Proceedings of FERA Workshop at Face and Gesture*, 2011.

T. Wu, N. Butko, P. Ruvolo, J. Whitehill, M. Bartlett, and J. Movellan. Multilayer architectures for facial action unit recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 2012.

T. Wu, N.J. Butko, P. Ruvolo, M.S. Bartlett, and J.R. Movellan. Learning to make facial expressions. *Proceedings of IEEE International Conference on Development and Learning*, 2009.

T. Wu, W. Mattson, J. Artigas, P. Ruvolo, J. Movellan, and D. Messinger. Collecting a developmental dataset of reaching behavior: First steps. *IROS Workshop on Cognitive Neuroscience Robots*, 2011.

ABSTRACT OF THE DISSERTATION

A Computational Approach to the Study of Social Interaction

by

Paul Lundy Ruvolo

Doctor of Philosophy in Computer Science

University of California, San Diego, 2012

Garrison W. Cottrell, Chair

For scientists, explanations of natural phenomenon based on optimality principles are critical tools for understanding the phenomena that shape the solutions the brain devises for the complex perceptual and motor problems of daily life. The neuroscientist David Marr called this type of analysis the "computational approach". While the computational approach has been applied with a great deal of success to phenomena such as neural coding and human motor control, the success of the computational approach for studying interactive behavior, particularly social behavior, has been more modest.

The purpose of this dissertation is threefold: to make the case for the study of social interaction from the computational perspective; to understand the challenges involved in this study and provide computational tools to address these

challenges; and to apply the computational approach to the study of social behavior in the real world. Our principle contributions are: (1) developing a framework for both analyzing and synthesizing behaviors in continuous state, action, and time from the perspective of the intentions that these behaviors appear to be realizing (our approach is well-suited for many motor-control and social-interaction problems), and (2) carrying out two computational studies of early infant social behavior that shed light on the computational forces that shape development. Our empirical results provide a new view of early infant social behavior as intentional, with the surprising intention of maximizing time spent with mother smiling at infant and the infant not smiling herself.

Chapter 1

Introduction

We are a highly social species. Our ability to flexibly organize ourselves into large groups capable of cooperating in a highly competitive world is responsible for our domination of the planet. However, while the behavioral sciences have concerned themselves with describing and cataloguing various characteristics of social interactions, little attempt has been made to understand these interactions from a computational perspective. Here, the term "computational perspective" invokes David Marr who advocated a careful study of the computational problems faced by the human brain as a means to both better understand its structure as well as to understand how an artificial brain might be developed [34]. Marr's program for computational analysis involves formalizing, in mathematical terms, a particular problem the brain appears to be solving, employing engineering methods to derive a solution, and finally comparing the engineered solution with the one developed by the brain as a means to illuminate the principles that might have shaped its development.

The purpose of this dissertation is to both address many of the difficult challenges of studying social interaction from the computational perspective and provide examples of the power of this approach. To this end we: (1) propose, develop, and extend algorithms to facilitate the study of social behavior from the computational perspective, and (2) apply these techniques to understanding, predicting, and imitating real-world social behavior.

The computational approach has been applied with considerable success to

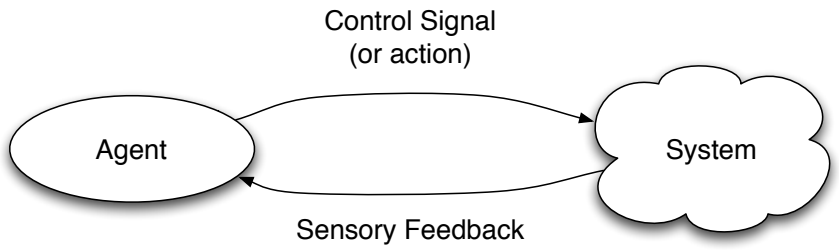

Figure 1.1: A schematic of the feedback control problem. An agent executes actions and receives sensory feedback from the system.

understanding sensory coding in the brain. For instance, Bell and Sejnowski used a computational-level analysis to show that the receptive fields of neurons in the primary visual cortex are approximately optimal for encoding visual information given the statistics of natural images [5, 50]. Others have applied similar methods to explain the brain's encoding of modalities such as audio [52]. There is also a significant body of work in the motor-control literature on using optimality principles to understand why, out of all the possible forms of human motor movements capable of achieving some task, the brain chooses to execute a particular sequence of movements [58, 25, 20, 29, 6]. Some of these models, most notably [58] and [12], consider the role of sensory feedback *during* task execution as a means of shaping on-the-fly the form of an unfolding motor movement. In this viewpoint, the problems of perception and control are inextricably linked, demanding that computational explanations of human behavior take into account the fact that the brain's generation of efficient motor movements unfolds within the context of sensory signals from the environment.

The mathematical framework used to formalize models of motor movements that account for the role of perception is known as optimal feedback control. In this framework an organism (or more generally an "agent") receives sensory feedback from the environment and in response specifies a control signal with the aim of achieving some task in an optimal fashion (see Figure 1.1). The schematic shown in the figure depicts an agent that interacts with the environment in a circular feedback loop where perception influences action, which influences perception, and

so on. In order to determine the optimal control signal that an agent should specify in response to a particular stream of sensory information, we must formalize the agent's performance criterion. This performance criterion might be defined either as some explicit reward given to the agent by the environment, for instance the delivery of food to a rat by a human experimenter, or could be determined by an internal evaluative process of the agent itself, for instance by the dopaminergic system of the brain. Finally, in order to determine the optimal solution, one must specify the constraints imposed by the environment (or "system"). For instance, in the case of modeling human-reaching-movements, the constraints of the environment are the kinematic and dynamical properties of the human musculoskeletal system. The beauty of this framework lies in its generality. In order to analyze any system with circular causality at the computational-level, one need only describe the system as an optimal feedback-control problem.

While the framework of feedback control has greatly advanced the ability to study systems at a computational level, the program of computational analysis of behavior goes back much further, most notably to the early Cyberneticians such as Wiener, Rosenblueth, and Von Neumann. Cybernetics was conceived as a systematic study of purposive or goal-directed behavior in mechanical and biological systems. The intellectual inspiration for this movement was partially derived from work that some of the pioneers of the field had done during World War II to design automatic weapons-targeting systems. For Wiener [62], the best way to understand systems that exhibited circular causality (such as the one depicted in Figure 1.1) was through the identification of the purpose, or ends, that the system's behavior achieved in the world.

What are the principal motivations for studying social behavior from the computational perspective? In a similar spirit to the early Cyberneticians, we view the identification of the goal of a social behavior as a fruitful way of understanding and predicting the social behavior of natural agents, as well as for providing approaches for artificial agents to effectively cooperate with and flexibly learn from natural agents. Our view is similar to that of Daniel Dennett, who advocated a strategy called the "Intentional Stance" for understanding behavior.

> We will find whatever mechanistic regularities there are in the functioning of internal systems whose design approaches the optimal (relative to some ends). In seeking knowledge of internal design our most promising tactic is to takeout intelligence loans, endow peripheral and internal events with content, and then look for mechanisms that will function appropriately with such "messages" so we can pay back the loans [17].

Here, Dennett draws a contrast between the intentional stance as a methodology for understanding behavior and Skinner's Behaviorism. Skinner, in the pursuit of characterizations of behavior, rigidly eschewed the usage of intentional terms such as "perceptions", "desires", and "beliefs", instead focusing on the development of laws that map environmental stimuli to an agent's behavioral response. Dennett's intentional stance is not meant to devalue this pursuit. On the contrary, Dennett believes that using intentional language to describe behavior (i.e. taking out an "intelligence loan" by assuming intelligence on the part of the agent under investigation), is the most effective way of illuminating the mechanisms that function in the support of that behavior. Once we permit ourselves intentional language, events in the world become endowed with content (e.g. photons hitting the retina become information, vibrations of the vocal chords become verbal communication). Finally, Dennett says that if our goal is to understand an agent's behavior at the mechanistic level, then ultimately we must explain, without the use of intentional language (i.e. repay our loans), how the agent is able to generate the behavior that we have characterized up to now as intentional.

Our aim in this dissertation is precisely to take out these "intelligence loans". These loans provide tantalizing hints as to the mechanisms that might be implicated in achieving the intentional behavior of the agent under investigation. Additionally, even without repaying these intelligence loans, we can reason about how behavior might change if something about the system under investigation is modified. This modification could consist of either placing an observed agent in a new environment, or reasoning about how another agent with the same intentions as the first (but with potentially very different sensorimotor characteristics) might behave in the original environment. Here, these predictions can be used for two purposes: (1) to suggest how the behavior of a natural system might

change in response to being placed in a new environment, and (2) to synthesize artificial behavior (e.g. on a robot) that achieves the same intentions as the natural system.

The idea of studying social interaction at the computational level has been proposed before [63], however, to date little progress has been made. What then are the principle obstacles in pursuing a computational-level analysis of social interactions? To understand the particular difficulties of the problem it is best to contrast the study of social behavior at the computational level with the study of motor movements at the computational level. For example, take the well-studied domain of computational explanations of human point-to-point reaching movements. Prominent computational-level models in this area include the "minimum-jerk model" of Flash and Hogan [20] and the minimum torque change model of Kawato *et. al.* [29]. Each approach articulates a different performance function to explain the generation of reaching movements. In order to test whether or not real human reaching movements support or discredit these possible performance functions, the solution generated by optimal control theory must be compared to empirical data. However, in order to compute the optimal behavior one must specify the constraints the human motor system places on the potential solution. Fortunately, in this situation these constraints are well understood given the sciences of Kinesiology and Newtonian Physics. In contrast when studying social behavior, there is no equivalently rigorous framework for deriving the constraints placed on potential solutions (i.e. there is no exact science of "Social Physics"). In this dissertation we take an empirical approach to learning these constraints by using techniques from machine learning, which provides a rich set of tools for learning structure from data.

Another difficulty for developing computational-level explanations of social behavior, is that in contrast to the study of point-to-point reaching movements, it may be difficult to determine *a priori* a reasonable performance function that explains an observed behavior. For instance, while the particular form of the performance function of point-to-point reaching movements is not settled, most everyone agrees that it has something to do with either maximizing the probability

of contacting a target, minimizing end-point variance [25], minimizing movement time, or some combination of these factors [6]. In a sense, it is easy to determine a proximal goal to explain these movements. However, what is the performance objective of an infant smiling at his mother? While we could say that the infant's goal is to maximize his probability of surviving into adolescence; trying to link behavior with such a distal goal is futile. What then is a useful proximal goal that might adequately describe this behavior? Again we turn to empirical techniques to solve this problem by employing new methods from machine learning to systematically search over a large space of possible performance functions until we find one that fits the data optimally. Again, one can see the contrast between the analytical nature of the formulation of computational explanations for point-to-point movement with the empirical nature of the computational explanations for social behavior.

In this dissertation we provide new computational algorithms for solving the challenges of studying social behavior from a computational perspective. We propose machine-learning based methods to provide approximate models of social behaviors, not unlike Newtonian physics provides models of physical behavior. In addition, we specify a model, based on Bayesian Inference, for determining which intention best characterizes observed behavior of two interacting agents. Finally, we extend this method of characterizing behavior based on intentions to a particular class of continuous time, continuous state, and continuous action control problems that can be used to model socially interactive behavior as well as more traditional single-agent behavior. Notably, our contribution to the determination of intentions for this class of systems allows for a computationally efficient algorithm for the synthesis of optimal behavior for a new agent (either a natural agent in a new domain, a robot operating in the same domain, or even a robot operating in a new domain).

In terms of applications of our techniques, we principally concern ourselves with the study of social interactions that occur in early life between mothers and their infants. These nonlinguistic interactions contain rich temporal structure and unfold across many modalities (including vocalizations, facial expressions, and

touch). The motivation for studying these social interactions from the computational perspective is to both gain an understanding of the computational problems infants solve early in life, and also to inform the development of robots that learn to successfully interact with the physical and social world through extended trial-and-error interactions with both social and nonsocial objects. We provide the first ever study of infant facial expressions from a computational perspective. We demonstrate that our findings not only facilitate the synthesis of social behavior on a humanoid robot, but suggest potential experiments for understanding the impact of various factors on normal social development. In addition, our findings suggest strategies for developing interventions to achieve better developmental outcomes for atypical infant populations. The contributions of each original chapter are summarized below:

Chapter 3

1. We develop a model for inferring goals of mothers and infants from their nonverbal behavior.

2. We show that mother's smiling can be accurately predicted by ascribing her the intentions of maximizing mutual smiling time with her infant.

3. We show that infant's smiling can be accurately predicted by ascribing him the intentions of maximizing time where mother is smiling and he is not smiling himself.

4. We implement our model of infant smiling on a humanoid robot with an infant-like appearance.

5. We perform a human robot interaction study that shows that undergraduates interacting with the robot have a similar intention as mothers do when interacting with their infants. Additionally, we show that the control policies derived from real infant smiling behavior have the expected effects on the smiling behavior of undergraduates interacting with the robot.

Chapter 4

1. We present a method for determining goals from behavior for biological and mechanical systems in continuous state, action, and time.

2. We show that various types of uncertainty can be handled by our technique in a principled fashion.

3. We show how to handle control problems with partial observability.

4. We illuminate the conditions under which the problem of inferring goals from behavior is underconstrained.

5. We provide two methods for incorporating prior knowledge about the goals of an agent in order to make the problem of goal-inference well-posed.

6. We provide a method for imitating the goal of a demonstrator in a novel situation (either with a different agent, a different environment, or both). We show that, for certain systems the imitation problem is well-posed even when the goal-inference problem is underconstrained.

7. We show how to determine goals from the behavior of two agents interacting in a game-theoretic setting.

8. We provide connections between the continuous and discrete inverse optimal-control problem.

9. We apply our techniques to providing principled and flexible methods for inferring the intentions behind infant head movements. We also implement a system based on computer vision to infer the intention behind human head movements from video in realtime that may have interesting applications in the field of Human Computer Interaction.

We begin with a presentation of the mathematical background that will allow the contributions of the dissertation to be understood more clearly.

Chapter 2

Mathematical Background

In order to pursue a computational approach to understanding social interaction, we must formalize several key concepts. We must define concretely the space of possible social behaviors, enumerate performance criteria which may explain an observed social behavior as optimal, and specify a framework for defining the constraints that the environment places on social behavior. To formalize these notions we invoke the frameworks of Bayesian Inference and Optimal Control Theory. These frameworks are naturally suited to precisely formulate what we mean for a social system to be optimal with respect to some goal, and also to provide data-driven methods for determining which, among a large space of goals, best characterizes real-world social behavior. The purpose of this chapter is to provide a crash course in the relevant theory. For more extensive treatments of these subjects consult [9] and [8].

Notation: Unless otherwise stated, capital letters are used for random variables, small letters for specific values taken by random variables. When the context makes it clear, we identify probability functions by their arguments: e.g., p(a,b) is shorthand for the joint probability mass or joint probability density that the random variable A takes the specific value a and the random variable B takes the value b. We use subscripted colons to indicate collections or sequences: e.g. $A_{1:t} \stackrel{def}{=} \{A_1 \dots A_t\}$. Symbols will be defined when they are first referenced in the text. For a listing of symbols used regularly throughout the text see Table A.1.

2.1 Bayesian Inference and Probabilistic Generative Models

Bayes' rule, while rather simple in form, has been instrumental in the development of modern machine learning techniques. At a purely syntactic level Bayes' rule specifies the relationship between two conditional probability distributions:

$$p(b \mid a) = \frac{p(a \mid b)p(b)}{p(a)} \tag{2.1}$$

However, in order to gain an appreciation of the semantic meaning of the preceding equation it helps to consider the role that this formula plays in probabilistic generative models. Probabilistic generative models specify a probability distribution over a set of random variables. The structure of the generative model typically suggests some sort of logical or plausible causal process by which these random variables are generated. For instance, a generative model might specify a distribution over some latent category label associated with an image, and then specify a probability distribution over the pixels that comprise the image given the previously generated category label. Given a probabilistic generative model, if we observe some subset of the random variables, then we can use the structure of the generative model to infer the values of the random variables that we do not observe (also called hidden variables). Bayes' rule is what allows us to make these inferences.

For instance, let X by a random variable representing some pixels in an observed image, and Y be a binary random variable which takes value 1 if the image contains a face and 0 if it does not. Suppose we observe a particular pattern of pixels x. We can use Bayes' rule to determine the probability distribution over the latent class label, Y, given the observed pixels.

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{2.2}$$

Which by the law of total-probability and the product rule can be rewritten as:

$$p(y|x) = \frac{p(x|y)p(y)}{\sum_{y \in \{0,1\}} p(x|y)p(y)} \tag{2.3}$$

Where the distributions $p(y)$ and $p(x|y)$ are given by the generative model.

2.2 Stochastic Optimal Control

The problem of stochastic optimal control is to determine a controller for an agent that achieves a specified performance criterion as well as possible. We use the term "possible" to indicate that the algorithm must only consider controllers that obey the constraints imposed by the agent's environment. At its most general, we imagine an agent receives an observation at each point in time and in turn specifies a control signal that probabilistically affects the agent's future observations (see Figure 1.1). The goal of the optimal control algorithm in this setting is to determine a mapping between any sequence of possible observations and the optimal control signal to execute in response. We refer to this mapping as a controller (or equivalently a behavior or policy).

Here, we assume the optimal control algorithm has access to a model of how the agent's control signals affect the likelihood of future observations given the past observations and the agent's executed control signals. The most basic form of the problem is to assume that the observations consist of a system state that encodes all the information needed to predict the system's future behavior. Such a system is known as a fully-observable first-order Markovian system. Here, the model of how the agent's control signals affect future observations is given by a probability distribution of the next system state given the current state as well as the agent's control signal (although having such model is not required by some solution techniques, for example, algorithms based on reinforcement learning [54]). Our objective will be to compute a controller that specifies which control signal the agent should execute in each state that maximizes the expected value of a given performance function over the longterm.

Throughout this document we use different terminology to refer to similar concepts (depending on the context). For the purposes of this dissertation we consider the following terms to be equivalent:

1. performance function = reward function = objective function = intention = -cost

2. action = control signal = decision

3. controller = policy = behavior

2.3 Discrete State, Action, and Time Optimal Control

We consider the problem of optimally controlling a dynamical system in discrete time. Further, we assume that the set of possible states, $\mathcal{X}$, the system can occupy is of finite size. Additionally, we restrict ourselves to the case where the set of actions, $\mathcal{A}$, available to the agent is of finite size. We assume that we are given a performance function, $r : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$, that assigns a scalar performance value to each possible state-action combination. As in the general formulation of the Stochastic Optimal Control problem given in the previous section, our goal will be to compute a policy, denoted as π (which in this case is a mapping from the current state of the system to an action), that maximizes the expected sum of the performance function over some time-horizon. In order to solve the control problem, we require access to the system's transition dynamics. Using the terminology of Chapter 1, these transition dynamics can be thought of as the constraints that the environment places on the space of potential solutions (in this case policies).

2.3.1 Problem Formulation

While there are many specific formulations of the optimal control, here we treat the popular discounted infinite-horizon variant (see [8] for a more complete treatment of discrete-time control). For the infinite-horizon discounted case, our goal will be to compute the policy, $\pi^{\star}$, that satisfies the following equation:

$$\pi^{\star} \quad = \quad \arg\max_{\pi} E\left[\sum_{i=0}^{\infty} \gamma^{i} R_{i} \;\middle|\; \pi\right] \tag{2.4}$$

Where $\gamma \in [0, 1)$ is a discount factor that specifies the agent's preference for achieving high performance in the short versus longterm (as $\gamma \rightarrow 0$ the agent behaves myopically, as $\gamma \rightarrow 1$ the agent values future performance as much as present

performance), and each R_i is a random variable which represents the reward the agent accrues at the ith timestep.

2.3.2 Solution By Dynamic Programming

There are several methods for computing an optimal policy for the problem formulation given in the preceding section. Here, we briefly sketch a very popular solution approach based on the principle of dynamic programming called policy iteration. The policy iteration algorithm iterates two steps until convergence: (1) evaluating a given candidate policy, (2) improving the candidate policy. To start, the candidate policy can be initialized to any value (random is a popular choice). The policy evaluation step involves constructing the value function, $v^\pi : \mathcal{X} \to \mathbb{R}$ for the candidate policy, π. Intuitively, the value function for a particular policy specifies the expected performance over the longterm when the agent begins in a particular state and follows the given policy. Formally,

$$v^\pi(x) = E\left[\sum_{i=0}^{\infty} \gamma^i R_i \;\middle|\; \pi, X_0 = x\right], \forall x \in \mathcal{X} \tag{2.5}$$

$$= r(x, \pi(x)) + \gamma \sum_{x' \in \mathcal{X}} v^\pi(x') p(s'|s, \pi(x)), \forall x \in \mathcal{X} \tag{2.6}$$

Where we went from the first step to the second step by applying linearity of expectations, using our transition model, and substituting the definition of v^π on the right-hand side. By consolidating the terms involving v^π on the left-hand side of Equation 2.6 we arrive at:

$$v^\pi(x) - \gamma \sum_{x' \in \mathcal{X}} v^\pi(x') p(s'|s, \pi(x)) = r(x, \pi(x)), \forall x \in \mathcal{X} \tag{2.7}$$

Equation 2.7 defines a linear equation for each state. There are exactly as many unknowns as equations. Additionally, the system of linear equations is guaranteed to have a single solution provided $\gamma \in [0, 1)$. The value function can be determined by solving this system of linear equations.

Once the value function is computed, we proceed to the second step of the policy iteration algorithm: policy improvement. In order to improve the candidate policy, π, we first construct the state-action value function, $q^\pi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$, which

specifies the expected performance over the longterm for executing a particular action in a given starting state and then behaving according to the given policy, π, thereafter. Formally,

$$q^{\pi}(x,a) = E\left[\sum_{i=0}^{\infty}\gamma^{i}R_{i} \;\middle|\; \pi, X_0 = x, A_0 = a\right], \forall x \in \mathcal{X}, \forall a \in \mathcal{A} \quad (2.8)$$

$$= r(x,a) + \gamma \sum_{x' \in \mathcal{X}} v^{\pi}(x')p(x'|x,a) \quad (2.9)$$

Since we already have computed the value function v^{π} we can easily compute the function q^{π} as well. The improved policy for the next iteration of the algorithm is given by maximizing q^{π} for each state as a function of the action.

$$\pi'(x) = \arg\max_{a} q^{\pi}(x,a) \quad (2.10)$$

Where π' indicates the improved policy. The two steps of policy evaluation and policy improvement are repeated until the policy improvement step does not change the candidate policy. Once this condition has been achieved the candidate policy is guaranteed to be optimal [8].

2.4 Optimal Control for Systems in Continuous State, Action, and Time

We now consider the problem of optimally controlling a continuous time, continuous state, continuous action system. We focus on sketching solution methods for a particular class of continuous time dynamical systems that have been used extensively for modeling biological and mechanical motor control and have great potential to model socially interactive systems (see Section 4.12 for potential applications).

2.4.1 Problem Formulation

First, consider the general case of controlling a dynamical system where $U_t \in \mathbb{R}^m$ is a random process that specifies the vector-valued control signal at time

t and $X_t \in \mathbb{R}^n$ is a random process that specifies the vector-valued state at time t. The dynamics of the system have the following form:

$$dX_t = a(X_t, U_t)dt + c(X_t, U_t)dB_t \tag{2.11}$$

Where dt is the time differential, dB_t is a vector of standard Brownian motion differentials, and the functions $a(\cdot)$ and $c(\cdot)$ control the evolution of the system state according to its deterministic and stochastic components respectively. For example, the state vector, X_t, might encode the positions and angular velocities of a robot's joint angles and the control signal, U_t, might specify torques exerted by a number of electric motors about those joints.

In this dissertation we consider a restricted form of the dynamical systems in Equation 2.11 that have the following form:

$$dX_t = \left(a(X_t) + b(X_t)U_t\right) dt + c(X_t)dB_t \tag{2.12}$$

This type of dynamical system is known in the literature as a "control-affine diffusion" since the deterministic component of the state-dynamics is an affine function of the control signal U_t. This particular dynamical system is widely studied in the motor control literature for its ability to model the movement of Newtonian systems (such as the motions of the human body or multi-segment robots). Another key difference between this formulation and Equation 2.11 is that the gain on the Brownian-motion term only depends on the state and not on the control signal. A very similar formulation that involves noise that scales with the magnitude of the control signal can also be solved using the techniques presented later in this chapter, however, the treatment of such a system is outside the scope of this dissertation.

Solving the optimal control problem involves computing a policy, π^*, that maps the state-time tuple (x, t) to an optimal action u^*. Here, we consider control problems over finite-time horizons with (optional) exponential discounting of performance over time. Our goal is to compute the policy, π, that maximizes the expected performance achieved during the time horizon. Specifically, $\pi^\star$ must satisfy the following equation:

$$\pi^* = \arg\max_{\pi} E\left[\psi(X_T) + \int_0^T e^{-\frac{t}{\tau}} r_t(X_t, U_t)dt|\pi\right] \tag{2.13}$$

Where T is the terminal time, r_t defines the instantaneous performance rate, τ specifies temporal discounting of the performance rate, ψ specifies the performance function at the terminal time, and the random processes X_t and U_t evolve based on the given policy π and the SDE in Equation 2.12. It can be shown through a straightforward application of Ito's rule, that finding the policy $\pi^\star$ that satisfies this equation can be reduced to find the function v that satisfies the following Partial Differential Equation (PDE) for all times t and states x (see [40] for a full derivation):

$$\begin{aligned} -\nabla_t v_t(x) &= \max_u \Big\{ -\frac{1}{\tau} v_t(x) + r_t(x,u) + \left(a(x) + b(x)u\right)^\top \nabla_x v_t(x) \\ &\qquad + \frac{1}{2} trace\left(c(x)c(x)^\top \nabla^2_{xx} v_t(x)\right) \Big\} \qquad (2.14) \\ v_T(x) &= \psi(x) \qquad (2.15) \end{aligned}$$

Where the action for the optimal policy, π^*, for a particular state-time tuple, (x,t), is given by the u that maximizes the right-hand side of Equation 2.14. Equations 2.14 and 2.15 are known as the Hamilton-Jacobi-Bellman (HJB) equations. The function v_t is known as the *optimal value function* and specifies the expected performance accrued when starting in state x and then running the optimal policy $\pi^\star$ starting from time t until the terminal time T. That is:

$$v_t(x) = \max_\pi E\left[\psi(X_T) + \int_t^T e^{-\frac{s-t}{\tau}} r_s(X_s, U_s) ds \;\middle|\; \pi, X_t = x\right] \qquad (2.16)$$

Next, we make the additional assumption that the performance rate is quadratic in u, i.e. $r_t(x,u) = \rho_t(x) - \frac{1}{2}u^\top q u$. Where q is a known symmetric positive-definite matrix (which can optionally depend on the state and time). This decomposition of the reward function involves two terms: an arbitrary state-desirability rate, ρ_t, and a quadratic cost-rate that penalizes large control signals. This decomposition allows us to perform the maximization over the control signal, u, in Equation 2.14 analytically. Following some simple algebra, the maximizer is given by:

$$\pi^*_t(x) = q^{-1} b(x)^\top \nabla_x v_t(x) \qquad (2.17)$$

This equation has a meaningful interpretation. In a particular state the agent should choose an action that follows the steepest ascent direction with respect to

the value function. The term $q^{-1}b(x)^\top$ modifies the notion of steepest ascent to account for both the relative "ease" of modifying the state-differential through a particular component of the control signal as well as differential costs that are incurred for using different components of the control signal. Next, we substitute the maximizer from Equation 2.17 into Equation 2.14 in order to obtain the HJB equation without the maximization operator.

$$\begin{aligned} -\nabla_t v_t(x) &= -\frac{1}{\tau}v_t(x) + \rho_t(x) + \frac{1}{2}\nabla_x v_t(x)^\top b(x) q^{-1} b(x)^\top \nabla_x v_t(x) \\ &\quad + a(x)^\top \nabla_x v_t(x) \\ &\quad + \frac{1}{2} trace\left(c(x)c(x)^\top \nabla^2_{xx} v_t(x)\right) \qquad (2.18) \\ v_T(x) &= \psi(x) \qquad (2.19) \end{aligned}$$

The preceding equations give the Hamilton Jacobi Bellman (HJB) equations for the stochastic systems with dynamics given by Equation 2.12 and optimality criterion given by Equation 2.13. Next, we turn our attention to methods for solving these equations in order to obtain an optimal controller.

2.4.2 Collocation Methods for Solving the Optimal Control Problem

For all but a few special cases, such as those resulting from linear control problems with quadratic state costs, it is computational difficult to solve the HJB equations exactly. In this dissertation, we consider solutions to these equations based on collocation. Collocation methods have become a popular approach for solving continuous state control problems due to their flexibility and natural connections to machine learning approaches [51, 59, 56].

The basic procedure for using collocation methods to solve the HJB is to first choose a set of times $0 = t_1 < t_2 < \ldots < t_l = T$ and corresponding sets of states $\mathbf{x_1} \ldots \mathbf{x_l}$. Next, we seek to satisfy the HJB as closely as possible at the resulting state-time tuples. Precisely, we seek to compute a v such that sum of squared differences between the left-hand and right-hand sides of the HJB equation

are as small as possible.

$$\begin{aligned} v^{\star} &= \arg\min_{v} \sum_{i=1}^{l-1} \sum_{x \in \mathbf{x_i}} \Bigg(\nabla_t v_{t_i}(x) - \frac{1}{\tau} v_{t_i}(x) + \rho_{t_i}(x) \\ &\quad + \frac{1}{2} \nabla_x v_{t_i}(x)^{\top} b(x) q^{-1} b(x)^{\top} \nabla_x v_{t_i}(x) \\ &\quad + a(x)^{\top} \nabla_x v_{t_i}(x) \\ &\quad + \frac{1}{2} trace\left(c(x) c(x)^{\top} \nabla^2_{xx} v_{t_i}(x) \right) \Bigg)^2 \\ &\quad + \sum_{x \in \mathbf{x_l}} \left(v_{t_l}(x) - \psi(x) \right)^2 \end{aligned} \tag{2.20}$$

We seek to minimize the preceding equation over all value functions from some parameterized family. In particular, we parameterize the value function as a linear combination of non-linear basis functions:

$$v_t(x, w_t) = \sum_{i=1}^{d} \phi_{t,i}(x) w_{t,i} \tag{2.21}$$

Where $\phi_{i,j}$ is the jth basis function at time t_i. There are two main approaches for computing the weights that minimize Equation 2.20. The first is to recognize that the optimization problem is a non-linear least-squares problem, and that tools such as Levenberg-Marquardt Optimization can be applied to compute a local minimizer [39]. However, a more efficient approach which we apply in this dissertation is to solve a sequence of linear least-squares problems backwards in time from the terminal time T to the initial time 0. We begin by showing how to compute the weights at the terminal time T. We seek to compute the weights that minimize the following equation:

$$w^{\star}_{t_l} = \arg\min_{w} \sum_{x \in \mathbf{x_l}} \left(v_T(x, w) - \psi(x) \right)^2 \tag{2.22}$$

Once we have solved for the optimal w at the terminal time we proceed backwards

in time by solving the following optimization problem:

$$
\begin{aligned}
w^{\star}_{t_{i-1}} \quad = \quad \arg\min_{w} \sum_{x \in x_{i-1}} \Bigg(& \frac{v_{t_i}(x, w^{\star}_{t_i}) - v_{t_{i-1}}(x, w)}{t_i - t_{i-1}} - \frac{1}{\tau} v_{t_i}(x, w^{\star}_{t_i}) + \rho_{t_i}(x) \\
& + \frac{1}{2} \nabla_x v_{t_i}(x, w^{\star}_{t_i})^{\top} b(x) q^{-1} b(x)^{\top} \nabla_x v_{t_i}(x, w^{\star}_{t_i}) \\
& + a(x)^{\top} \nabla_x v_{t_i}(x, w^{\star}_{t_i}) \\
& + \frac{1}{2} trace \left(c(x) c(x)^{\top} \nabla^2_{xx} v_{t_i}(x, w^{\star}_{t_i}) \right) \Bigg)^2
\end{aligned} \tag{2.23}
$$

We replaced the temporal derivative of the value function with an approximation based on finite differences. Since we assume a linear parameterization for v_t and that w_i is held fixed before solving Equation 2.23, the optimization variable w only shows up linearly. Thus, the preceding optimization problem can be solved using linear regression. Additionally, if the set of points x_i are constant for all time indices and the same set of basis functions are used for all time indices, one can precompute the pseudo-inverse for the regression problem and solve each step of the backwards pass using a matrix multiplication (rather than having to repeatedly solve a least-squares regression).

Chapter 3

Computational Analysis and Synthesis of Infant Intentions in Early Social Interaction

Abstract: We present a computational study of the intentions of mothers and four-to seventeen-week-old infants engaged in face-to-face interaction. We model sequences of mother and infant smile onsets and offsets as optimal behavior with respect to both the agent's intention and the statistics of the probabilistic responses of his or her partner. We develop a Bayesian model to infer these intentions from a database of mother-infant interactions by identifying key markers of intentionality in the temporal patterns of both mother and infant smiling. Our model shows that the pattern of smiles of mother is consistent with the intention of drawing her infant into prolonged periods of mutual smile. Surprisingly, the patterns of infant smiling reveal an intentional agent that seeks to make his mother smile without smiling himself. Next, we instantiate our model of infant smiling on a highly-expressive humanoid robot with an infant-like appearance. The data from this study exhibited strikingly similar patterns to the mother-infant interaction data.

3.1 Introduction

A key goal of developmental psychology is to characterize the progression of infants' social, cognitive, and motor capacities that lead to the emergence of intentional communication towards the end of their first year [11, 53, 31, 60, 16, 65, 36]. In this endeavor, scientists have looked for markers of "adult-like" intentional communication, such as eye-contact and persistent gestures, as signals of the beginning of the ability of infants to actively pursue their intentions in social interactions. However, some scientists, such as Bates [4], have suggested that infants need not necessarily wait for these milestones to actively enlist others in the pursuit of their intentions. In contrast, intentional infant social behavior is distinguished only by an infant's ability to manipulate social objects, e.g. a caregiver, in a similar manner as an infant might manipulate a physical object. Here, we provide a formal framework for defining, detecting, and characterizing the emergence of intentional social behavior in early infancy. Our framework determines from first principles the specific forms that infants' intentional behavior can assume, providing a rigorous data analysis technique for understanding intentional behavior without constraining ourselves to look for the development of adult-like markers of intentional communication. Our framework is based on adopting what Daniel Dennett calls the "intentional stance" toward understanding the nonverbal behavior that marks face-to-face mother-infant interaction in the first few months of life [17].

Dennett's "intentional stance" is a strategy for an external observer to understand and predict the behavior of an agent by ascribing it intentions (e.g. goals or desires). However, to realize the power of the strategy we should not simply ascribe arbitrary intentions to the agent, but rather those that best explain an observed pattern of behavior as optimal with respect to the constraints imposed on the agent by its environment. For instance, consider the case that we are on a safari and observe over several hours the actions of a lion and a group of zebra. We watch the lion stalk through the high grass slowly approaching the herd. When the lion closes within 20 feet of the herd she springs into action; throwing herself at the closest zebra. How should we understand this lion's behavior? A very compact and powerful description is to adopt the intentional stance toward the lion by

ascribing it the intention of killing a zebra. From this point of view, the very disparate behaviors of stalking through the grass and running full speed toward the zebra suddenly become unified under an intentional explanation. Additionally, if we desire to predict what the lion will do in a hypothetical situation, then we can reason about which of the lion's behaviors would maximize its chance of achieving its intention in this new situation. For instance, if one of the zebras broke from the pack and appeared to be injured, it would be quite easy to guess what the lion would do next. Even if we did not know *a priori* that lions like to hunt zebra, if we assume that the lion is an intentional agent and have a reasonable idea of a set of intentions that lions might have, then it would become obvious quite quickly that, out of all these possible intentions, the lion's movements are best predicted by ascribing to her the intention of trying to bring down one of the herd.

Another example of a system whose movements can be successfully explained by adopting the intentional stance is an anti-aircraft missile. Imagine as you pilot your jet fighter through the skies, that you see on your radar a rapidly approaching anti-aircraft missile. As you bank your plane hard to the right, you notice that the missile abruptly changes course to the right as well. A very compact and powerful understanding of the missile's behavior in this situation is that the missile is an intentional system with the intention of striking your jet. This understanding of the missile as intentional allows you to accurately predict future movements of the missile in response to evasive maneuvers that you might execute.

In the two preceding examples, the mechanisms that lead to an external observer being able to use the intentional stance as a successful means of understanding and predicting behavior are quite different. In the case of the lion, the intentional behavior was likely shaped by millions of years of evolution along with learning processes that occur during the lion's lifetime. In the case of the anti-aircraft missile, the intentional behavior is realized by the efforts of human engineering and technological innovation. From the point of view of the intentional stance, all that matters is the success of the strategy for explaining behavior. The strategy of adopting the intentional stance is agnostic to the underlying processes and mechanisms that bring about the intentional behavior. To put it another way:

if you were a zebra in the herd or the pilot in the preceding example, would you care about the particular mechanisms that brought about the intentional behavior of the lion or the missile? Determining the intentions behind behavior, whether mechanical or biological, is vital to our survival. In fact, we generate intentional explanations of behavior so effortlessly that we probably don't even realize how regularly we engage in the practice.

Here, we adopt the intentional stance toward understanding the behavior of mothers and infants in face-to-face interactions. As we previously stated, in order to fully realize the power of the intentional stance for both understanding and predicting behavior we do not wish to ascribe arbitrary intentions to mothers and infants but instead ascribe those that optimally explain their behavior. In the case of the lion, it seem obvious that when a hungry lion is in close proximity to a zebra, ascribing her the intention of hunting the zebra is a highly successful application of the intentional stance. What intentions should we ascribe to mothers and infants? Here we employ a data-driven approach. We use new techniques from Bayesian inference and optimal control theory to search over a large space of possible intentions until we find the one that best explains the behavior of infants and their mothers.

Next, we instantiate a model of the observed infant smile-timing in an expressive infant-like robot. We perform a human-robot interaction study in which we recreate the setting of the mother-infant face-to-face interactions as faithfully as possible. We seek to determine if the smile behavior, learned from infants interacting with their mothers, will achieve the inferred intention on undergraduates interacting with the robot.

In the discussion, we examine how the identification of a particular intention that successfully predicts infant behavior suggests new research directions, including both behavioral and neuro-imaging studies, for exploring the mechanisms behind the realization of this intention. Another strength of adopting the intentional stance is the ability to make predictions of mother-infant behavior when the behavior of one partner is abnormal. To this end, we discuss possible applications of our model to understanding both infant social development with depressed

mothers, as well as possible ways in which the intentional stance may help design early diagnoses and treatments for Autism.

3.2 Optimal Control Models of Behavior

The intentional stance explains an agent's behavior by ascribing it intentions that render the agent's actions optimal with respect to the constraints imposed by its environment. In this section, we formalize Dennett's intentional stance by formalizing what it means for a behavior to be optimal with respect to a particular intention and a particular set of environmental constraints. Here, we show that the mathematical theories of Optimal Control and Bayesian Inference are ideally suited for this task. *Please note that some of this material is also contained in Chapter 2, however, key bits are repeated here to allow this chapter to be more clearly understood.*

3.2.1 Mathematical Formalism of Optimal Control

The optimal control problem is to determine a controller for a system that achieves a particular performance objective on average as well as possible. Where the term "average" is required because we allow the system to be stochastic. In the most general case, an agent is faced with a stream of observations, and performs actions that probabilistically affect the agent's future observations. The controller (or policy), which is the output of an optimal control algorithm, specifies an action for the agent to execute in response to any stream of observations that maximizes the agent's average achievement of the given performance objective.

Here, we use a common formulation of the optimal control problem called a Markov Decision Process (MDP). In the MDP setting, at each discrete timestep the agent observes the current state of the system and in response specifies an action. The underlying system is assumed to be Markovian which implies that the probability distribution over the next state of the system is conditionally independent of all previous states and actions given the current state and the current action. The probability distribution over the next state given the current state and

action is called the transition dynamics and intuitively specifies the laws governing the behavior of the system (or equivalently the constraints that the environment places on possible solutions). The specification of the system states, agent actions, and transition dynamics are collectively known as the plant of the control system, and intuitively specify the probabilistic laws governing the agent's interaction with its environment.

We assume that the agent seeks to maximize, over the long-term, the average value of a performance function, $r(\cdot)$, which specifies how desirable an agent finds each encountered state-action pair. In our formalization of Dennett's intentional stance, we equate the notion of an agent's intention and this performance function, $r(\cdot)$. The optimal behavior that maximizes a given performance function is called the optimal policy, denoted by the variable, $\pi^\star$, and specifies the optimal action for the agent to perform in any particular state. We require that $\pi^\star$ satisfy:

$$\pi^\star \quad = \quad \underset{\pi}{\operatorname{argmax}}\, E\left[\sum_{t=0}^{\infty} \gamma^t R_t \;\middle|\; \pi\right] \tag{3.1}$$

Where $R_{0:\infty}$ is an infinite sequence of random variables specifying the value of the performance function at each time step and $\gamma \in [0, 1)$ is a discount factor that specifies how desirable the agent finds achieving high performance now versus in the future. Richard Bellman provided a set of equations that provide necessary and sufficient conditions for a given policy (or controller) to be optimal:

$$\pi(x) \quad = \quad \underset{a}{\operatorname{argmax}} \left\{ r(s, a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x, a) v^\pi(x) \right\}, \forall x \in \mathcal{X} \tag{3.2}$$

Where v^π represents that value function for the policy π, x is a state, and $\mathcal{X}$ is the set of possible system states. Intuitively, the value function specifies the expected long-term achievement of the performance function when starting in the state x and employing the policy π. The expression inside the maximizer on the right-hand side of Equation 3.2 is the state-action value function, q^π, and is defined as:

$$q^\pi(x, a) \quad = \quad r(s, a) + \gamma \sum_{x' \in \mathcal{X}} p(x'|x, a) v^\pi(x) \tag{3.3}$$

The function q represents the expected value of the performance function of the agent over the long term if it executes a particular action in the state x, and

then behaves according to the policy π thereafter. Intuitively, we can think of the relative values of $q^{\pi}(x, a_1)$ and $q^{\pi}(x, a_2)$ as determining which action, a_1 or a_2, is more optimal in the state x for achieving the agent's intentions. To find the optimal policy, we use the policy iteration algorithm [8].

In order to complete the formalization of Dennett's intentional stance, we next define a function that specifies how likely an agent is to execute a particular action in a particular state given the optimal policy. Later, we will use these likelihood functions to determine which of a set of intentional models best explains a database of observed behavior. How should these probabilities be determined? A simple choice would be to assign probability 1 to the action given by $\pi^{\star}$ and 0 to all others. However, this choice is likely to be overly optimistic about the ability of the agent to consistently execute exactly the action prescribed by the optimal policy. Here, we assign probabilities to an agent's actions by assuming it employs a softmax action selection rule:

$$p(a|x, q^{\pi^{\star}}) \quad = \quad \frac{e^{\tau q^{\pi^{\star}}(x,a)}}{\sum_{a'} e^{\tau q^{\pi^{\star}}(x,a')}} \tag{3.4}$$

Where $\tau \in \mathbb{R}^{+}$ is a scalar parameter that when close to 0 makes the action probabilities uniform and when close to infinity reverts to predicting the agent will always choose the optimal action.

All of the machinery we have presented thus far has assumed that we know the agent's intention. What if we don't know *a priori* the agent's intention, but must infer it from observing the agent's behavior? Here, we employ a Bayesian approach, originally proposed in [47], to solve the problem. Our strategy is to use Bayes' rule to compute the posterior probability of a potential intention given observations of the agent's actions. With a bit of algebra and the assumption that each action is independent given the agent's intention and the current system state we can show that:

$$p(r|x_{1\ldots N}, a_{1\ldots N}) \quad \propto \quad p(r)\Pi_{i=1}^{N} p(a_i|x_i, r) \tag{3.5}$$

Intuitively, this equation specifies that our belief about how likely an agent is behaving optimally according to a particular intention is proportional to the product

of two terms. The first term is the prior probability of how likely we thought the agent was to pursue the intention r before we observed the agent's behavior. The second term encodes how probable the actions of the agent are assuming that the agent has the intention r. This completes our formalism of Dennett's intentional stance. Note, that just as the intentional stance dictates that we should ascribe intentions that best explain observed behavior, here we ascribe the intentions that assign the highest likelihood to the observed behaviors. In other words, we seek the intentional explanation that results in us being the least "surprised" by the agent's actions.

3.3 Mother-Infant Interaction Study

We employ Dennett's intentional stance toward understanding the early patterns of smiling between mothers and their infants. Our aim is to ascribe intentions to both mothers and infants that best explain their behavior as optimal. The technique described in the previous section gives us a principled method for asking the following question couched in intentional language "in these early interactions, what are they each trying to do?". It is possible that our method will uncover that all intentions are equally good at explaining the observed patterns of interaction (i.e. the posterior distribution over performance functions is flat). However, if after analyzing the patterns of interaction most of the posterior probability is on a particular performance function then we can confidently adopt these intentions as suitable explanations for understanding early mother-infant face-to-face interactions.

3.3.1 Dataset

As part of a study carried out by the developmental psychologists Alan Fogel and Daniel Messinger [38], thirteen mother-infant dyads were seen weekly between the ages of four and twenty-four weeks. The mother was instructed to play with her infant for a period of 5 minutes in a similar manner as she would at home. The infants were positioned on the mother's lap, facing towards her.

The following behavioral channels were coded at 30Hz: mother smile (yes / no), infant smile (yes / no), infant gaze (at mother / away from mother). Since gazing away has been identified as a mechanism for infants to regulate arousal [36], and thus may signal a switching of intention, we only analyze segments when the infant is gazing at mother. Additionally, between the ages of 18 and 24 weeks infants begin to spend a significant portion of their time gazing at non-social objects in the environment; therefore, an analysis of infant intentions for this age range that does not take this shift into account is likely to be misleading. We choose to focus on the period of life between 4 and 17 weeks when the majority of an infant's time in the face-to-face interaction is spent gazing at mother.

3.3.2 Model

Here we specify the mapping between the behaviors in the longitudinal dataset and language of control theory defined in the previous section.

Plant Models

We begin by defining the plant models that describe the control problem faced by both mother and infant. That is, in order to determine the intentions of mothers we specify the relevant states, actions, and transition probabilities governing the control problem she faces. Similarly, to model the intentions of infants, we specify the relevant states, actions, and transition probabilities governing the control problem he faces.

State space: the state of the interaction is encoded with two binary channels, one to encode whether or not each partner is smiling, as well as two continuous valued channels that encode how long each partner has been in its current configuration of smiling. For instance, one state of the system is that mother is smiling and she has been smiling for 5 seconds while the infant is not smiling and he has been not smiling for 2 seconds. Each of these dimensions of the state have been previously identified [37] as being important for predicting both mother and infant nonverbal behavior. In order to apply the solution techniques for the control problem described in the previous section, we convert the continuous dimensions of the

state space by discretizing them into fifty 400ms segments representing the time intervals $[0s, 0.4s), [0.4s, 0.8s), [0.8s, 1.2s), \ldots, [19.6s, \infty]$. To construct the full state space, we take the Cartesian product for each of the individual state components (i.e. the 2 binary smile variables and the 2 temporal variables) yielding a total of 10,000 states ($2 \times 2 \times 50 \times 50 = 10,000$).

Action space: when modeling infant behavior, the action space encodes whether or not the infant will be smiling at the next time step. When modeling mother behavior, the action space encodes whether or not mother will be smiling at the next time step.

State transition dynamics: When modeling infant behavior the transition dynamics encode mother's probabilistic responses to the infant's smiling, whereas, when modeling mother behavior the transition dynamics encode the infant's probabilistic responses to mother's smiling. One can think of the transition dynamics as playing the role of the "Social Physics" described in the previous section. In order to estimate the transition probabilities for a given state, we require an estimate of how likely the other agent is to change smile configurations during the next 400ms. For instance, if we are modeling the control problem from the infant's point of view, then the transition probabilities from the state of both smiling when it has been 1.2 seconds since infant started smiling and 2 seconds since mother started smiling are completely determined by how likely mother is to stop smiling within the next 400ms in that particular state. The probability of a particular agent changing her smile given the current state is computed using maximum likelihood estimation with temporal pooling. Pooling is required due to data sparsity, which did not allow us to estimate the probability of an agent switching smile for each of the $10,000$ states independently. Specifically, we fit the smile change probabilities independently for each joint smile configuration, but pooled data over similar temporal contexts. We estimate the smile change probabilities over the next 400ms by pooling data over the 16 regions of the two-dimensional space of time since each partner last changed smile (shown in Figure 3.1). The particular choice of the regions is motivated by allocating exponentially less temporal resolution as the time since each agent changed smile configurations gets larger. As a result, the

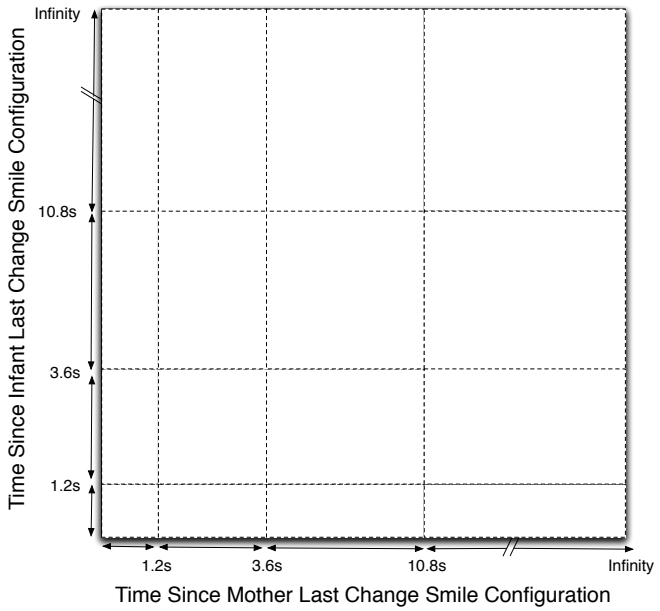

Figure 3.1: Regions used for temporal smoothing for maximum likelihood estimation of the state transition probabilities.

transition dynamics models for mother and infant each have 64 model parameters.

Prior Distribution over Intentions

We assume that each agent, mother and infant, intends to maximize a particular performance function from some finite set. We construct this set of intentions to encode our prior beliefs about intentions that are likely to accurately explain both mother and infant smiling. Each performance function considered assigns a fixed performance value per unit time spent in each of the four possible joint smile configurations. Specifically, without loss of generality we set the performance value for one of the configurations to always be 0, and then perform a uniform grid search over the performance values (ranging from -.5 to .5, with 11 total values considered) for the other 3 configurations. This construction gives us a total of 1,331 possible intentions (or performance functions) to consider. We assume a uniform prior probability over each of these 1,331 intentions. In order to compute the posterior probability of a particular characteristic of the agent's

intention (e.g. that infants prefer mother to smile at them), we use the law of total probability which tells us to sum the posterior probability mass over all intentions that are consistent with the particular characteristic.

Controller Models

Our model produces a template for each considered intention which assigns a likelihood to each observed smile onset and offset. We use a variant of the softmax action selection rule defined in Equation 3.4 to specify the likelihood of an agent's action given a particular intention.

$$p(smile|x, r, w_1, w_2) = \frac{e^{w_1 q^{\pi^\star}(x,smile,r)+w_2(issmiling(state))}}{e^{w_1 q^{\pi^\star}(x,smile,r)+w_2(issmiling(x))} + e^{w_1 q^{\pi^\star}(x,no-smile,r)}}$$

Where $q^{\pi^\star}$ is the optimal state-action value function for a given intention, x is the state of the interaction, r describes the agent's intention, w_1 and w_2 model the agent's ability to choose the most efficient action as well as an inertia term that enforces continuity of smile behavior over time, and $issmiling$ is a binary function that returns 1 if the agent is smiling in state x and -1 otherwise. The values of w_1 and w_2 are chosen independently for each considered intention to maximize the likelihood of the agent's actions. Recall from the previous section that the higher the likelihood of the dataset under a particular intentional model, the better the agent's actions can be predicted by ascribing the agent that particular intention.

Clarifying Example

Next, we present a hypothetical example of how observing different patterns of mother smiling towards her infant might help us provide evidence either for or against a particular intention. Suppose the state transition dynamics that we learn from data are such that: (1) infants will never smile unless their mother is smiling, (2) if the infant is to respond to a particular mother smile, she will only respond between 1 and 2 seconds following its initiation. Assuming mother's intention is to get her infant to smile with her as much as possible, what behavioral policy would be optimal for achieving this intention? Without providing too much unnecessary detail, a rough intuition is that the optimal strategy would be for her to smile for

two seconds, and if the infant does not smile back, then she would stop smiling and then immediately smile again. The softmax action selection rule specifies how likely mother is to smile in any particular state. To continue our hypothetical example, we would find it very unlikely if a mother who has the intention of making her infant smile ceases to smile after only smiling for 1 second. However, we would be far less surprised if that same mother took a short break in between attempts to get her infant to smile. The softmax rule formalizes the intuitive logic that pausing for a second or two in between smiles is not significantly worse than smiling immediately after ending a smile for getting her infant to smile. However, if mother always stops smiling after 1 second then she will never cause her infant to smile.

Figure 3.2 shows the graphical model (originally proposed in [47]) specifying the generative process for intentional behavior. The circles correspond to random variables and the arrows incident to a particular circle can be interpreted as the random variable in the circle being probabilistically generated based on the values of the random variables in the incident circles. For instance, the figure encodes that the variable X_2 is generated based on U_1 and X_1. Shading is used to indicate which variables are able to be observed directly from the data. The unshaded variables are those that we infer using Bayesian inference. In the case of modeling infant's behavior, the action nodes $A_{1:n}$ correspond to infant actions and the plant model encodes the infant's knowledge of mother's probabilistic responses to her actions. In the case of modeling mother's behavior, the action nodes $A_{1:n}$ correspond to mother actions and the plant model encodes the mother's knowledge of infant's probabilistic responses to her actions.

3.3.3 Results

Mothers interacting with their infants have the intention to maximize periods of mutual smiling ($p < .001$ non-parametric test). That is, if our goal is to explain mother's behavior as effectively as possible we should ascribe to her the intention of maximizing time spent engaging in mutual smiling with her infant. In order to compute the given non-parametric significance value, first new sessions

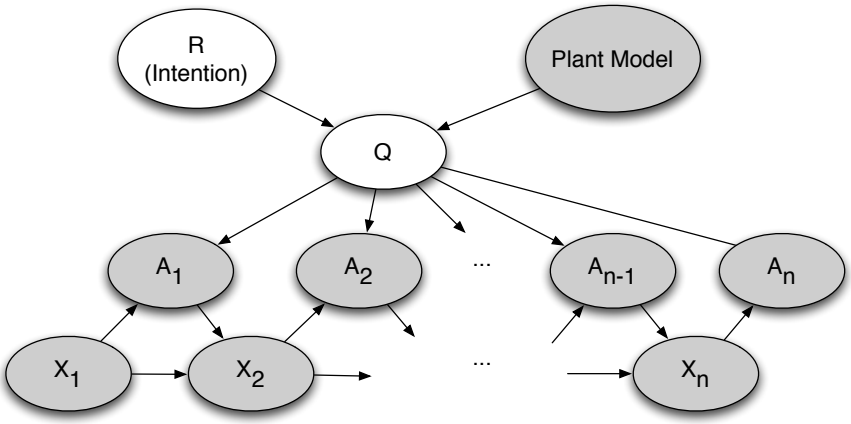

Figure 3.2: The graphical model specifying our model of the generation of intentional behavior of either mother or infant. The bubbles labeled X are states, those labeled A are actions, and Q is the optimal action-value function given a particular performance function and a plant model. Subscripts on variables indicate the temporal sequence in which the variables are generated.

were synthesized using randomly paired mothers and infants which were then analyzed in an identical manner to how we analyzed the true pairings. Second, we observed how many times the random session pairings showed a particular intention to be a better characterization of the data than the other possible intentions by the same or greater margin as we found in the true pairings.

Next, we found that the smiling of infants reveals an intention to maximize time spent with mother smiling at them while they are not smiling themselves ($p = .02$ non-parametric test). We refer to this configuration as "mother-only smiling". That is, out of all the possible intentions that infants might have, the one that is most predictive of infant smiling is the intention of maximizing the duration of mother-only smiling. On the x-axis of Figure 3.3 are the smile initiation and termination probabilities predicted by different infant intentions, and on the y-axis are the empirical probabilities of those same events. The size of the points is proportional to the total amount of time spent in each context. The intentions shown in the scatter plots were selected as the most predictive intentions from each of several sets of intentions that have easily interpretable high-level meanings (from left to right across each line): any intention, maximize time spent in both not

smiling, maximize time spent in infant-only smiling, maximize mutual smiling, and indifference to mother smiling. The best fitting intention (top plot) corresponds to maximizing time spent in mother-only smiling. If a model were a perfect fit to the data, then all of the points would lie on the line $y = x$. To compute how well each model fits the data, we computed a weighted correlation for each scatter plot (where the weighting is determined by the proportion of time spent in each context). Table 3.1 shows weighted correlations across all contexts as well as weighted correlations across only those contexts corresponding to smile initiations. The data in the table show us that several intentions capture the general trend that smile initiations are less probable than smile terminations. However, only the model of the infant's intention as maximizing time spent in mother-only smiling captures this coarse trend as well as the trend of the contexts in which infant is more or less likely to initiate a smile (see Figure 3.4). Table 3.1 shows that the best fitting intentional model achieves a weighted correlation between model predictions and empirical data for contexts corresponding to smile initiations of .5 (much higher than the other models). The fourth column of Table 3.1 shows how the information viewed in the scatter plots translates into a ratio of how likely the data is under one intentional model versus the intention to maximize mother-only smiling. All other intentions have very small likelihood ratios, indicating the presence of strong evidence for the intentional model of maximizing mother-only smiling. Therefore, even though some of the weighted correlations are relatively close among some of the intentional models, we have enough evidence to be able to confidently hone in on one particular intentional explanation for infant's behavior.

Figure 3.5 shows descriptive statistics of the posterior distribution over both mother and infant intentions. The figure elaborates our principal findings concerning each agent's preferred smile configuration by specifying the quantitative performance value that each agent appears to assign to being in each of the four joint smile configurations. In addition to these group-level characterizations of intentions, we also computed distributions over the intentions of individual mothers and infants (i.e. using only one individual's observed smiling behavior). Due to data sparsity, the plant model was not modified to reflect the particular re-

Table 3.1: Weighted Pearson correlation coefficients between the smile initiation and termination probabilities for different intentional models and the empirical probabilities. The weight for each point is proportional to the amount of time spent in each context. The second column is the weighted correlation across all contexts, whereas the third column is the weighted correlation across only contexts corresponding to smile initiations. The fourth column is the likelihood ratio between a particular intention and the intention to maximize mother-only smiling.

Intention	**All**	**Infant Not Smiling**	**Lik. Ratio**
Mother Smile, Infant No Smile	0.73	0.50	1
Both No Smile	0.66	-0.33	1.0×10^{-19}
Both Smile	0.13	0.22	1.4×10^{-67}
Infant Smile, Mother No Smile	0.47	-0.11	1.3×10^{-43}
Indifferent to Mother Smile	0.68	0.00	3.4×10^{-14}

sponse patterns of an individual's partner. Using the same method as we used for characterizing group-level intentions, for each agent (i.e. a mother or an infant) we computed the posterior probability that he or she maximally prefers each of the four joint smile configurations. The bottom row of Figure 3.5 shows the mean of these posterior probabilities across the 13 mothers and 13 infants. These bar graphs indicate that while there is individual variability in intentions across the dataset, the majority of infants and the majority of mothers have intentions consistent with the inferred group-level intentions.

Next, we performed two analyses to further illuminate specific infant strategies that were responsible for the model's inference of the infant's intention as maximizing mother-only smiling. We call these analyses "When to Smile" and "When to Stop Smiling" for reasons that will become clear shortly.

When to Smile

We first examined a context in which the infant would have to determine an optimal tradeoff between maximizing performance now and executing behavior that leads to better performance over the long term. Specifically, we looked at when the infant should smile again if he has just broken a mutual smile with mother. Our analysis shows that infants have the intention of maximizing mother-

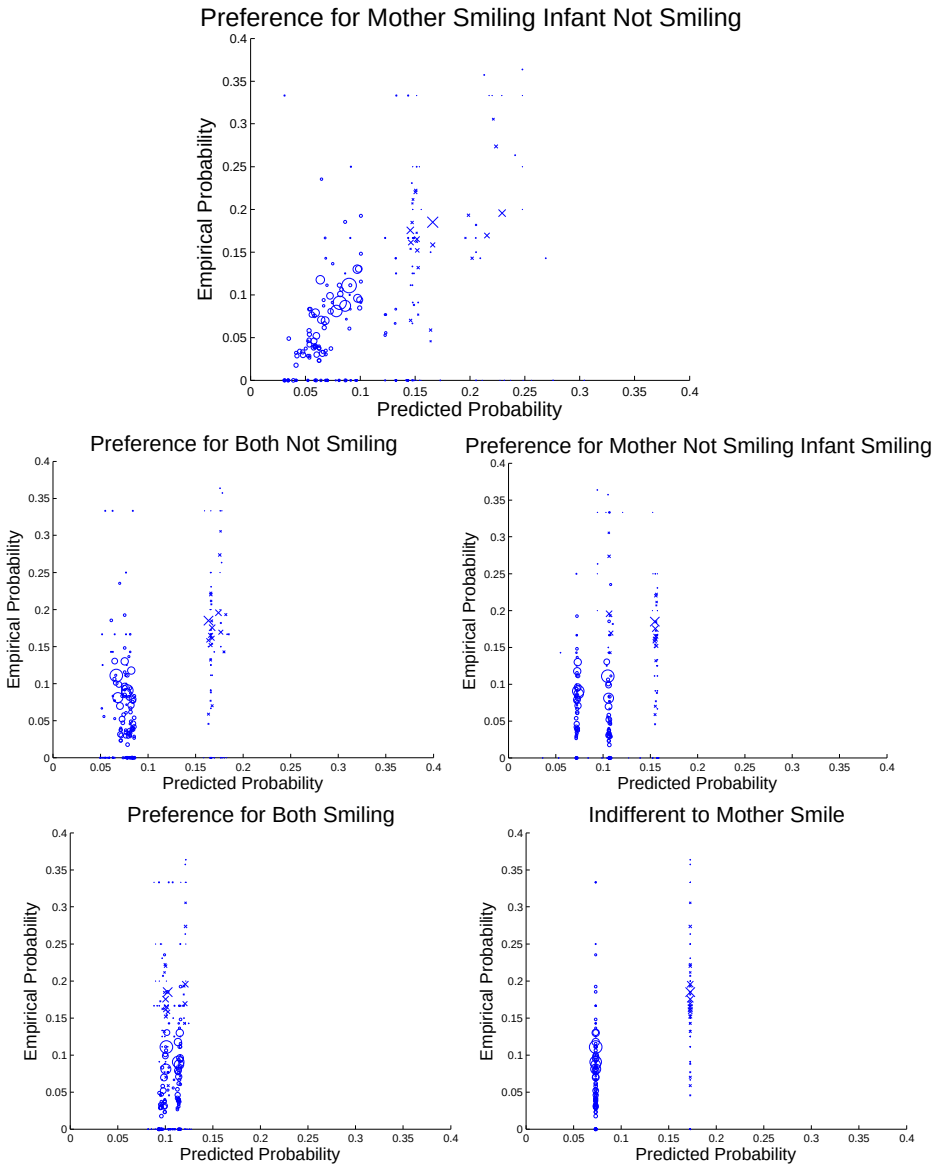

Figure 3.3: Scatter plots showing empirical infant smile initiation and termination probabilities versus model predictions. The size of each point is proportional to the amount of time spent in that particular context. Smile initiations are shown with a circle and terminations are shown with an "x".

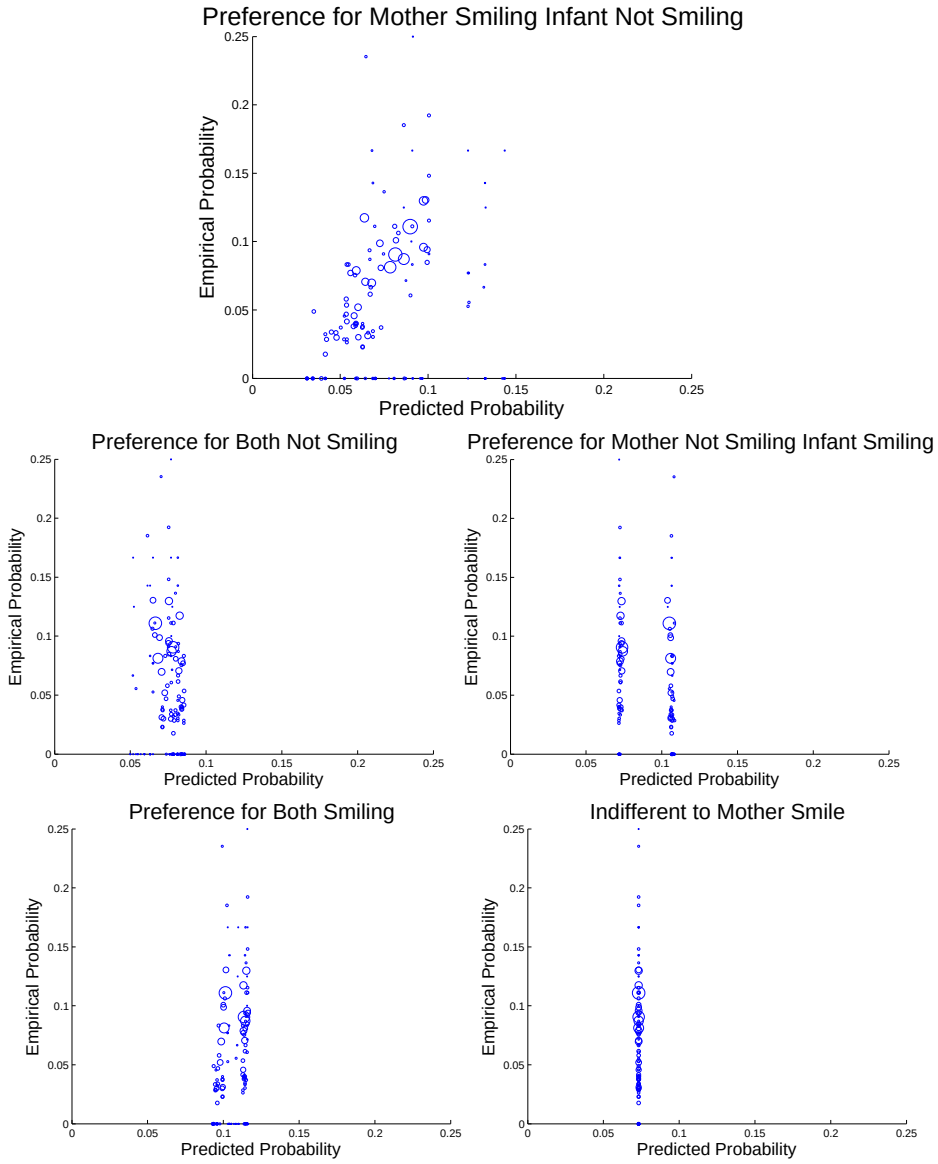

Figure 3.4: Scatter plots showing empirical infant smile initiation probabilities versus model predictions. The size of each point is proportional to the amount of time spent in that particular context.

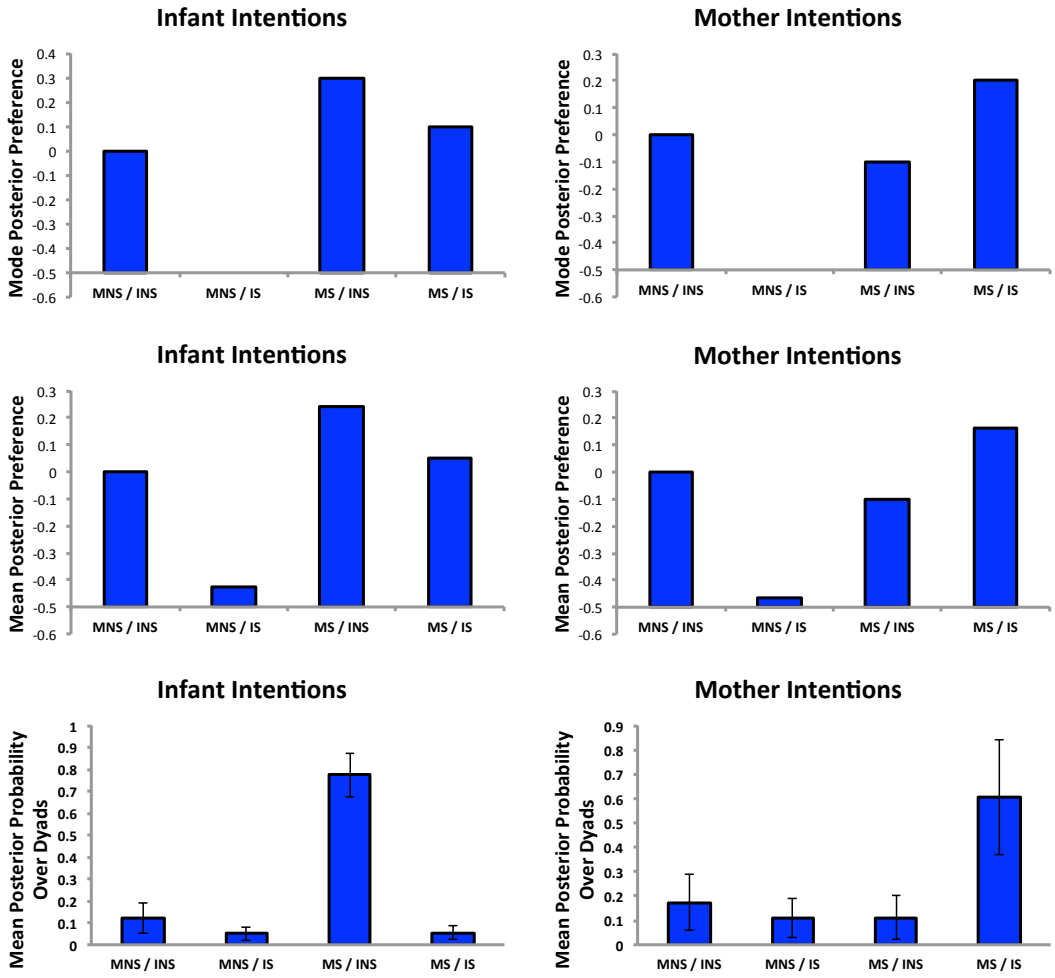

Figure 3.5: Descriptive statistics of the posterior distribution over mother and infant intentions. The bar graphs in the left column correspond to the distribution of infant intentions, and those in the right column refer to mother intentions. From top to bottom the rows indicate: the posterior mode, the posterior mean, and the mean posterior probability (over the 13 dyads) that a given infant or a given mother will maximally prefer a given state. MS and IS are short for Mother and Infant Smile. MNS and INS are short for Mother Not Smiling and Infant Not Smiling.

only smiling, which is precisely the configuration in which the infant now finds itself. Naively, one would expect that the infant would be very unlikely to ever smile in this situation. However, empirically this is not the case. In fact the infant is much more likely than average to smile after having recently broken a mutual smile. To understand how smiling in this situation might be an optimal behavior, we considered two potential plans of action that the infant might follow in this situation. First, the infant might decide not to smile again and simply enjoy the current state of mother-only smiling. Alternatively, the infant might decide to rejoin mother in a mutual smile some number of seconds in the future. The first strategy can be viewed as myopic since it maximizes the time spent in mother-only smile right now. However, the second strategy might be better in the long run. Suppose that the infant smiles 2 seconds after breaking the mutual smile. The infant will have experienced 2 seconds of mother-only smiling before rejoining the mutual smile. However, when the infant ceases smiling again, she has the chance to enjoy another 2 seconds of mother-only smiling, and so on. Of course, it is possible that if the infant waits 2 seconds to smile, then mother might stop smiling before the infant gets a chance to smile again. Therefore, the infant is in effect facing the problem of determining the optimal tradeoff between risking that mother will stop smiling before the infant rejoins mother in a mutual smile and enjoying the mother-only smile right now. Figure 3.6 shows the expected amount of mother-only smiling the infant would get before mother stops smiling for various infant smile wait times. The empirical data show that the smile wait times with higher expected mother-only smiling are also those that are more likely to be selected by the infant. The bottom four plots in Figure 3.6 show the predictions for alternative infant intentions. The alternative intentions do not fit the empirical trend.

When to Stop Smiling

Next, we examined the relationship between different durations of infant-initiated smiles (smiles where mother is currently not smiling) and their relative efficiency for achieving mother-only smiling. We examined each episode where the following occurred: (1) infant smiled at mother and mother was not smiling, (2)

after some period mother begins to smile and infant terminates her smile (either ordering is acceptable), and (3) mother eventually terminates her smile. We computed the total amount of mother-only smiling during one of these episodes as a function of the duration of the infant's smile. Shown in the left plot of Figure 3.7 are the efficiencies of different infant smile durations for creating mother-only smiling. The right plot shows that smile durations that are less efficient are less likely to be selected by the infant. The bottom four plots in Figure 3.7 show the predictions for alternative infant intentions. While several other intentions match the empirical probability for different duration smiles, no alternate intention is a good fit to the data in Figure 3.7 and Figure 3.6.

Our analysis presents a different picture of infant intentional communication than the classical view. We show that early infant social behavior can successfully be described as intentional without any need to look for morphological characteristics of adult intentional communication (e.g. eye contact or persistent gestures). In this view, the first forms of infant intentional communication do not arise at the end of the first year, but during the first four months of life. In order to enrich our understanding of these principles we perform a human-robot interaction study with a highly expressive infant-like robot. The usage of a humanoid robot allows us to manipulate timing and contingency patterns in a way that helps us better understand the intentional nature of smiling in early infancy.

3.4 Human-Robot Interaction Study

The purpose of this study is to determine if the models of infant smiling distilled from the mother-infant interaction data would have a similar effect on participants interacting with a highly life-like infant robot as they did on mothers interacting with their infants. Of particular interest is determining whether the temporal patterns of infant smiling, which we have shown can be understood as intentional behavior for maximizing mother-only smiling, will maximize participant-only smiling when instantiated on a robot. The robot used for this study, Diego San, has a realistic infant-like face that is capable of displaying a large repertoire

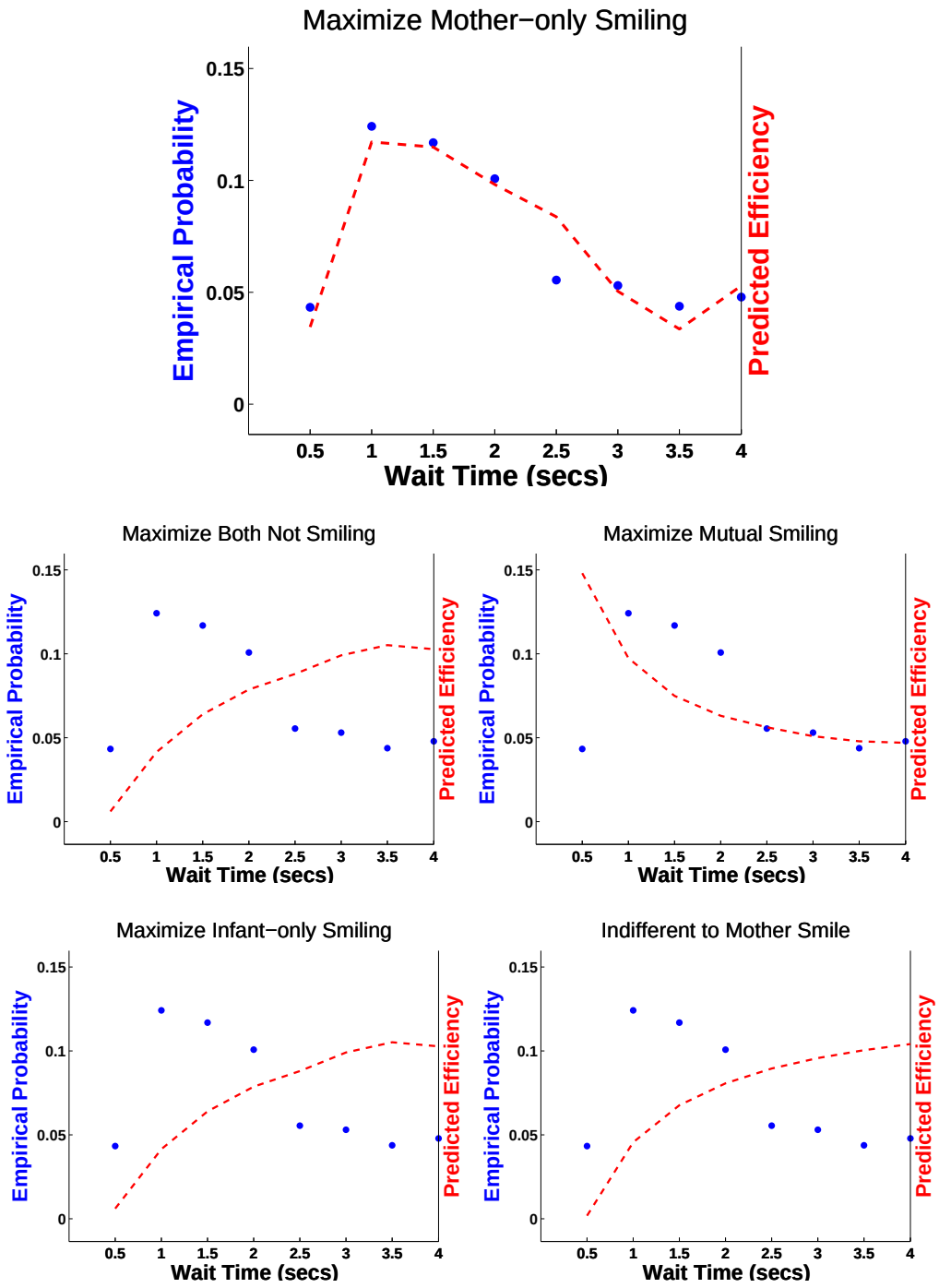

Figure 3.6: The average performance (dashed lines) of various infant wait times before rejoining mother in a mutual smile when the infant has just terminated a mutual smile vs. the empirical probability (dots) that the infant selects a particular wait time. Each plot corresponds to a different possible infant preference over joint smile configurations.

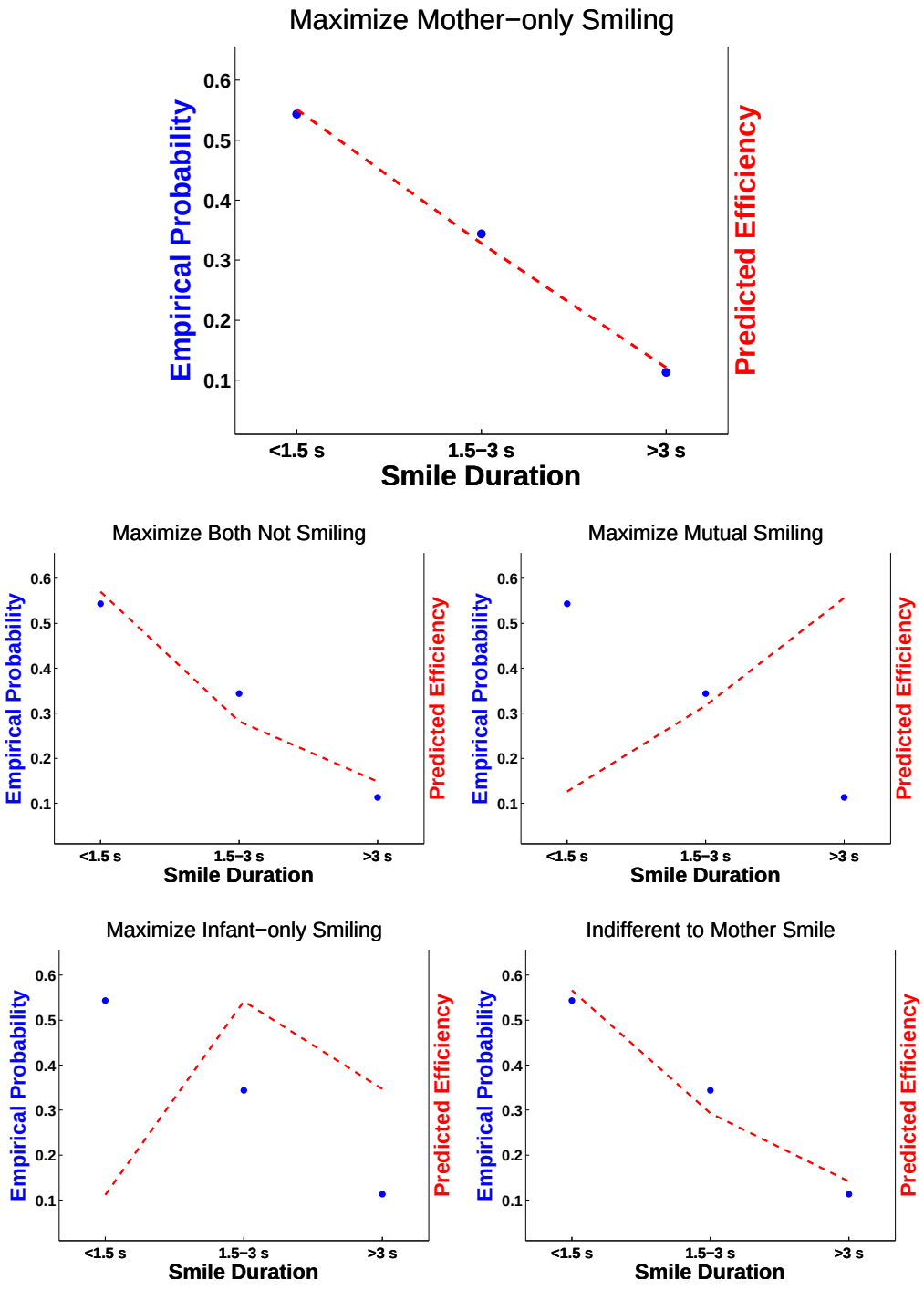

Figure 3.7: The average performance (dashed lines) of various durations of infant-initiated smiles vs. the empirical probability (dots) that the infant selects a smile duration. Each plot corresponds to a different possible infant preference over joint smile configurations.

of facial behaviors. Additionally, the robot's body is pneumatically actuated allowing for the production of smooth human-like movements. In this study, we programmed the robot to simulate the interactions that occurred in the mother-infant data.

3.4.1 Robot Sensorimotor Behavior

The robot tracked the participant's face using a combination of saccadic eye movements and head turns. While tracking the face, the robot was programmed to perform vergence eye movements so that the angle between the two eyes was appropriate for the viewing distance. Periodically, Diego would flap his arms, which subjectively gave the appearance of an expression of delight. The form of these flapping motions was derived from a motion-capture database of mother-infant interactions [64]. We programmed Diego to perform an open-mouthed smile known in the infancy literature as a "play smile" [46] (see Figure 3.8). Diego would randomly blink at an average frequency of 0.5 Hz. Since infants do not spend the entire face-to-face interaction gazing at their mothers, but occasionally look away, we programmed the robot to periodically rotate about its torso to either the far right or the far left. In coordination with the body movement, the robot shifted its gaze to give the appearance of being interested in an object to the side of the participant. A selection of the robot's behavior is shown in Figure 3.8.

We programmed four robot controllers that specified different patterns of body and facial movements in response to the location and expression of the participant's face. Ultimately, we sought to determine how the choice of controller affected participants' opinions of the robot as well as their behavior when interacting with the robot. Each controller tested had identical face-tracking, blinking, and vergence as described above. The following four controllers were tested:

Infant: Look away behavior was generated at an average frequency of 1/20 Hz. When looking away, the robot looked back at the subject at an average frequency of 1/3 Hz. Additionally, when the robot smiled, with 50% probability the robot flapped its arms. The probability of the robot smiling was based on the statistics of infant smiling from the longitudinal database.

Infant Plus: Looking away and arm flapping were identical to the Infant controller. The smile timing of this controller was identical to *Infant* with the modification that the robot was more likely to modulate its expression to be the same as the participant (elevated probability of matching of 50% per second). This controller allowed us to test the effect of a more contingent smile policy on the participants.

Replay: Diego's look away behavior, arm movements, smiles, and blinks were all matched to those recorded from the *Infant* controller interacting with the previous participant. Thus, while the statistics of each behavior were identical to the "Infant" condition, there was no contingency between the participant's smiling and the robot's smiling. However, face-tracking remained contingent.

Mirror: We were interested in how a controller that was minimally random and easily controllable would be perceived by the participants. The mirror controller did not look away, and matched its smile to that of the participant. Each time it smiled, the robot would also flap its arms.

In order to determine whether or not the participant was currently smiling we employed CERT (the Computer Expression Recognition Toolbox) [33] which provides automated realtime face detection and facial expression analysis from video. The video signal used to extract this information came from two cameras, one located in each of the robot's eyes. In order to determine when a participant changed from smile to not smile or not smile to smile, we detected when the output of the smile detector had crossed a threshold (set to 0 for all participants) for at least half a second (the half second threshold was used to make the robot's perceptions of the participant's smile less sensitive to transient noise).

3.4.2 Procedure

Participants were recruited through the UC San Diego Psychology department's subject pool. All participants were undergraduates, and each received course credit for participating in the experiment. Upon arriving at the lab, participants were given the following written instructions:

Researchers at the Machine Perception Laboratory are design-

> ing a robot named "Diego San". Diego San is just beginning to learn how to interact with people. Diego San has the ability to sense and respond to some of the same social cues that humans use to communicate with each other. Currently, Diego San can see where people's faces are and whether or not they are smiling. He does not have any other perceptual abilities (such as the ability to detect gestures or sounds). In this experiment you will interact with Diego San for four 3-minute sessions. During each session, Diego San will run a different social interaction program. Each program specifies a different pattern for how Diego responds to your actions. Following each interaction, we will administer a questionnaire that asks you to evaluate Diego San's behavior during the previous 3-minute interaction.

Diego was enclosed in a four-sided curtained enclosure to increase the participant's sense of privacy when interacting with the robot. The participant was required to remain seated during the experiment and to remain at least 18 inches from the robot at all times. The participant was seated in a chair with rolling wheels that allowed them to move within the enclosure while remaining seated. The participant interacted with each of the four robot controllers: infant, infant-plus, replay, and mirror. The order of presentation of each controller was counterbalanced to avoid order effects. Following each interaction, a questionnaire was administered (see Dependent Measures).

3.4.3 Dependent Measures

Smiling of the participant was recorded for later analysis using CERT [33]. For simplicity, we defined smile as all times when the output of CERT's smile detector exceeded the threshold of 0. Specifically, we computed the total amount of time spent in each of the four possible joint smile configurations of robot and participant. Between sessions, we administered the Godspeed questionnaire, a standard in the field of Human-Robot Interaction, to probe the participants' opinions of their preceding interaction with Diego. The questionnaire consisted of 21 5-point Likert scale items (e.g. is the robot *1 - apathetic* or *5 - responsive*). In order to assign a single number to how much a participant liked a particular controller, we

Figure 3.8: Diego-San the robot used in this study. The robot can generate life-like facial expressions as well as generate compliant and human-like motions with its body. Diego's perceptual capabilities were provided via computer vision software operating on images delivered from cameras in its eyes. In the top row of the figure is an example of the robot verging his eyes on a close face. In the second row of pictures are two key frames from Diego's arm-flapping behavior.

summed their responses over the 21 5-point Likert scales.

3.4.4 Results

Thirty-two participants took part in the experiment. An ANOVA revealed a significant effect of condition ($p = .0007$, $F(2, 94) = 4.45$) on overall rating after controlling for a participant's average rating across all conditions. Subsequent paired t-tests revealed that *mirror* was rated higher than either *infant* ($p = .036$, $t(31) = 2.19$) or *replay* ($p = .006$, $t(31) = 2.96$). Additionally, *infant* was rated higher than *replay* ($p = .027$, $t(31) = 2.32$). We developed an ANCOVA model to assess whether the time spent in each of the four smile configurations during an episode was related to the participant's rating of that episode. To this end we performed separate correlations between rating and time spent in each of the four possible smile configurations after partialing out the effect of participant. The only significant correlation was between amount of time spent in mutual smile with Diego and rating ($R = .57$, Pearson correlation, $p = 1.4 \times 10^{-9}$, $t(94) = 6.72$). Thus, based on their responses, participants had the same intention that mothers had when interacting with their infants (to draw their partner into a mutual smile). This result helps explain why subject's liked the mirror condition the most; in the mirror condition the robot mechanistically responded to a participant's smile with a smile of its own, making achieving the intention of mutual smiling quite easy.

Next, we examined which of the four controllers elicited the most participant smiling (see Figure 3.10). We report the results from the second half of the 3-minute interaction because the first half often contained periods where the subject was testing the affordances of the robot controller, however, the overall pattern of results when using both halves is very similar. An ANOVA revealed a significant effect of condition after controlling for a participant's mean smile time across all conditions ($p = .000002$, $F(3, 126) = 11.045$). Subsequent paired t-tests revealed that participants smiled more to *mirror* than either *infant plus* ($p = .0069$, $t(31) = 2.89$, two-tailed) or *replay* ($p = .00004$, $t(31) = 4.79$, two-tailed). Importantly, participants smiled more to *infant* than *replay* ($p = .0041$, $t(31) = 3.10$, two-tailed). Thus, the contingency of the infant controller to the participant's smiling

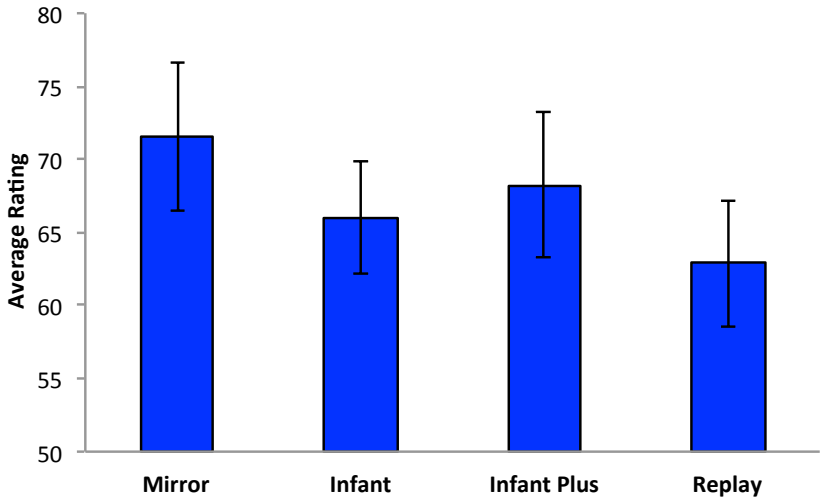

Figure 3.9: The average participant rating of each of the four smile controllers. Error bars represent standard errors. Significance between conditions was assessed using pair-wise t-tests.

increased the total amount of participant smiling.

In the mother-infant interaction study we found that the timing of infant smiling can be accurately predicted by ascribing infants the intention of maximizing mother-only smiling. To investigate whether the human-robot interaction results exhibited a similar trend, we analyzed the time spent in participant-only smiling for each controller. The bottom plot of Figure 3.10 reveals that the duration of participant-only smilaxing was longest for the infant controller. An ANOVA revealed a significant effect of condition after controlling for the mean amount of participant-only smiling across all conditions ($p = .0000001$, $F(3, 126) = 13.57$). Subsequent paired t-tests revealed that the duration of participant-only smiling was significantly longer for the infant controller than either mirror ($p = .00004$, $t(31) = 4.78$), infant-plus ($p = .0037$, $t(31) = 3.14$), or replay ($p = .0107$, $t(31) = 2.72$). The temporal patterns of infant smiling are effective at achieving the intention of partner-only smile, even when these timings are translated from the context of mother-infant interaction to interaction between undergraduates and a robot.

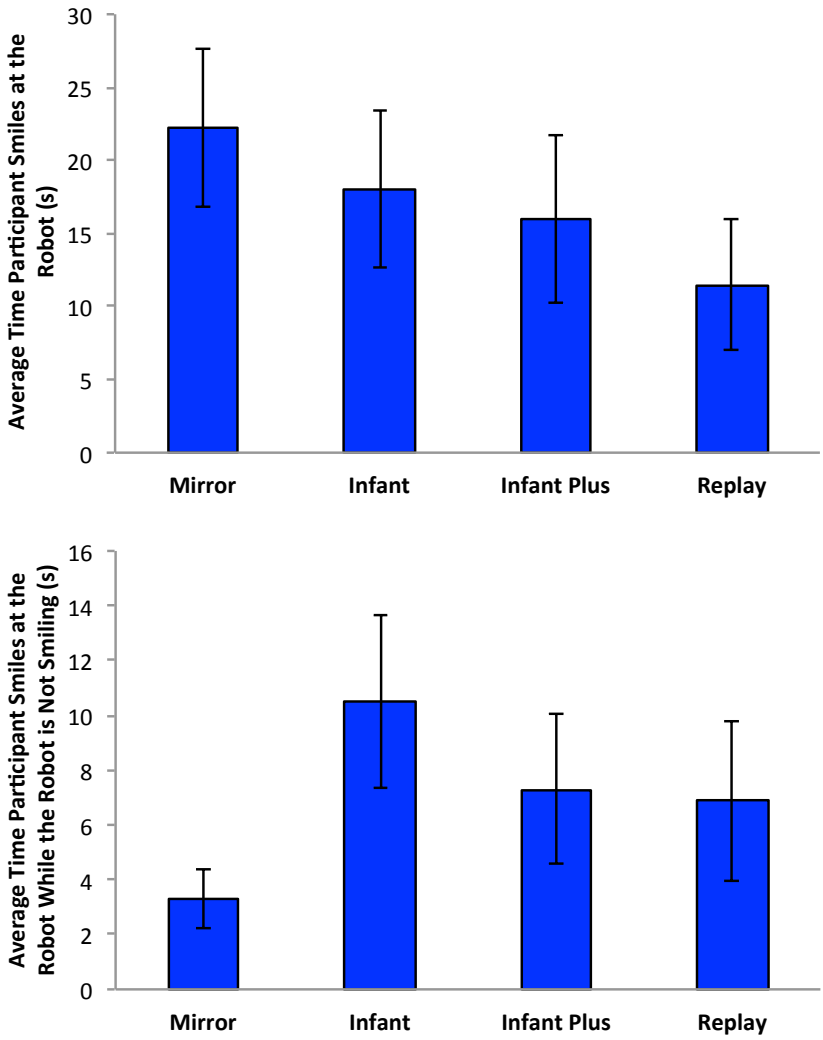

Figure 3.10: **Top:** the average duration of participant smiling for each of the four controllers. **Bottom:** the average duration of participant-only smiling for each of the four controllers. Error bars represent standard errors. Significance between conditions was assessed using pair-wise t-tests.

3.5 Discussion

Our results provide a principled computational account of the intentions of early social interactions between mothers and their infants. The young age at which infants are able to pursue intentional communication with the intent to engender a particular social response (in this case a facial expression) from another human is quite surprising. Moreover, the particular strategies that are implicated in the pursuit of these intentions are not simple, but require subtle sensitivity to the statistics of the smiling responses of mothers. Our findings suggest two follow-up experiments to illuminate the causal processes behind the realization of this intentional behavior. Firstly, if infants indeed have the intention of creating mother-only smiling, perhaps we might find increased activation of the reward centers in the infant's brain when viewing their mother's smiling face. Secondly, if the infants are shaping their smiling behavior in response to increasingly sophisticated models of the statistics of mother-smile responses in various contexts, we may be able to directly probe these expectations using a looking time paradigm [57].

The view of the infant as an intentional system can be applied to understanding mother-infant social interaction in atypical contexts. For instance, depressed mothers exhibit less smiles and less contingency to their infants than non-depressed mothers [19]. In response, their infants exhibit less smiling and generally less positive affect [19]. This pattern of infant response to a less contingent mother is predicted by our intentional model. Since infant smiles are not as effective a tool for achieving the intention of mother-only smiling (due to lower contingency on mother's part) smiling will be exhibited less frequently by the infant. Another setting in which the intentional stance may prove helpful is in examining the smiling patterns of infants who are eventually diagnosed with Autism Spectrum Disorder (ASD). If, as some have theorized [14], toddlers with ASD do not have the desire (or intention) for reciprocal social interaction, perhaps our model will reveal that infants who eventually develop ASD do not have the intention of maximizing mother-only smile, thus providing an early diagnostic for ASD. Additionally, if our model finds that the intention of ASD infants is to gain some other form of stimulation, perhaps this knowledge could be used to design early interventions to

help these infants in their social development.

3.6 Acknowledgment

The text of Chapter 3 is unpublished work to be submitted with authors P. Ruvolo, T. Wu, D. Messinger, A. Fogel, and J.R. Movellan. I was the primary author and researcher on this project, constructing the models, analyzing the data, and drafting the manuscript. Fogel and Messinger collected the dataset used for the analysis presented in this chapter. Wu created the software and hardware infrastructure necessary for running the human robot interaction experiment. Movellan supervised the research presented in this chapter.

Chapter 4

Inverse Optimal Control and Goal-based Imitation in Continuous State, Action, and Time

Abstract: We present an algorithm for inferring the intentions (or goals) of a demonstrator from observations of its behavior. In contrast to existing approaches that are tailored for the discrete-state Markov Decision Process, our approach handles a rich class of continuous time, state, and action control problems. Our approach to intention inference is computationally efficient and naturally allows for online updating of the estimate of the demonstrator's intention in response to new observations. Our method handles uncertainty in a principled fashion by both inferring distributions over possible intentions, rather than just point estimates, and naturally handling the case when the system dynamics of the demonstrator are not known exactly. We approach the problem of goal-imitation by showing that even when the intention of the demonstrator cannot be uniquely identified (i.e. the goal can only be constrained to lie on a manifold) that an imitator can synthesize behavior that is optimal with respect to the demonstrator's goal. We conclude by providing extensions of our model to partially observable control problems,

stochastic differential games, and control problems in discrete time with discrete state and action spaces.

4.1 Introduction

Learning from demonstration (or *Imitation Learning*) is a powerful method of skill acquisition in humans [41]. Due to the power of learning by imitation, roboticists have sought [3] to develop computational methods that allow robots to learn by observing humans performing a task. Loosely, one can categorize existing approaches as either: seeking to imitate the surface characteristics of a demonstrated behavior (mimicry) or seeking to imitate the goal of a demonstrated behavior [61] (see [49] for a survey of this work). In order to better contrast the two approaches, consider a robot watching a human walking across a narrow bridge. The robot in this case uses wheels to locomote, thus determining how to mimic the observed movements of the human's legs is non-trivial. However, if we can infer from the human's movements that his goal is to translate his body from one side of the bridge to the other without falling, then we can apply engineering methods to synthesize behavior for the robot that is optimal for achieving this task. In this case, the particular solution we construct will be tailored to the characteristics of the robot's sensors and actuators. We call this form of imitation *goal imitation* or *deep imitation*. Here, we provide a method of deep imitation and goal inference for continuous state, continuous action, and continuous time systems.

One approach to solving the problem of deep imitation is to perform the following two steps:

1. Infer the performance function (or intention) of the demonstrator from its behavior.

2. Compute an optimal controller for the imitator with respect to the performance function inferred in step 1.

To perform the second step, one can apply one of the numerous algorithms from the field of stochastic optimal control (e.g. dynamic programming [7] or reinforcement

learning [54]) for computing optimal controllers for a given performance function (this is known as the forward control problem, see Chapter 2). The problem posed in the first step is known as the Inverse Optimal Control problem and was originally formulated by Kalman [28]. However, a complication arises. It is known (see [44] for example) that for many control problems, a given policy of the demonstrator is not optimal with respect to only a single performance function. Therefore, the problem of computing the performance function of an agent from behavior may be underconstrained. Here, we provide two approaches for dealing with this issue. Firstly, in the case when we desire an explicit estimate of the demonstrator's performance function, we show that prior information about the agent's performance function can be leveraged to make the problem of goal-inference well-posed. Secondly, if an explicit estimate of the performance function is not required, but rather the synthesis of optimal imitative behavior, then by considering the two steps identified above jointly the lack of a unique performance function that is optimal for the demonstrator's behavior does not preclude deep imitation. We achieve this result by showing that for a large class of continuous systems, any performance function that renders the demonstrator's behavior optimal leads to the same prescribed optimal behavior for the imitator.

Algorithms for inverse optimal control are not limited in application to deep imitation. As we saw in Chapter 3, inverse optimal control methods can be used as an effective means for understanding natural behavior. Here, our method of inverse optimal control applies to a class of continuous systems of considerable interest in the study of biological and mechanical motor control and social interaction. Just as we used the framework of discrete optimal control to formalize Dennett's intentional stance in Chapter 3, the techniques presented here formalize the intentional stance for continuous systems. Despite the richness of the class of control problems that we consider, our solution to the inverse optimal control problem is quite straightforward. This simplicity allows us to consider several extensions not possible with previous methods. Firstly, our technique yields an estimate of the uncertainty of the inferred performance function rather than a point estimate. Secondly, we show how to use the uncertainty of the performance function to influ-

ence the generation of deep imitative behavior. Thirdly, we demonstrate that our method can handle uncertainty in the dynamical model of the demonstrator. We extend our method to partially observable control problems, the game-theoretic setting, and to a particular variant of discrete control. In order to help inspire future applications of our framework, we provide three examples of how the framework developed in this chapter could be used to answer questions of considerable interest in the study social interaction.

4.2 Related Work

In recent years, the inverse optimal control problem for discrete-time, discrete-state, and discrete-action Markov Decision Processes (MDPs) has received a great deal of attention from the machine learning community [1, 47, 55, 30, 42, 44]. In this setting, algorithms must make inferences about a demonstrator's performance function from state-action trajectories sampled from the demonstrator's optimal policy. While these approaches have achieved impressive results [30, 43], they scale poorly to problems that are more naturally formulated in continuous state, continuous action, and continuous time (such as the ones considered later in the chapter).

Approaches to inverse optimal control typically maximize one of two objectives. The first objective is to produce a performance function that makes the action choices of the demonstrator optimal compared to alternative actions. Underlying these formulations is an objective function that computes a score (e.g. log-loss or $0-1$ loss) between the optimal policy prescribed by a performance function and the observed behavior of the agent. Next, this objective function is either maximized to yield a point estimate of the performance function [44, 42, 15], or used as a likelihood function for a Markov Chain Monte Carlo (MCMC) based sampling procedure [47]. These approaches are sometimes called *policy matching* approaches. The second objective, also known as apprenticeship learning, is to produce a policy that achieves similar performance as the policy of the demonstrator under the assumption that the performance function of the demonstrator

[1, 55] is linear in some known features. In this formulation there is no guarantee that the produced policy will behave in a similar fashion to the demonstrator, only that its expected performance over the longterm will be similar. Our formulation bears similarities to both of these approaches.

Our method for performance function inference is based on *policy matching* in that our inference procedure maximizes a matching score between the demonstrator's actions and the optimal policy dictated by a particular performance function. However, our approach is also similar to *apprenticeship learning* in that we show that our formulation can be used to produce a policy that is optimal with respect to the inferred demonstrator's performance function even when we cannot uniquely determine this performance function. An additional benefit of our approach is that it allows for *apprenticeship learning* for producing optimal imitative behavior for an agent with different motor characteristics than the demonstrator, something that is not handled by other approaches.

Only a limited number of techniques have been developed to handle the case of inverse optimal control for non-linear continuous systems. Our work is most similar to that of Li *et. al.* [32]. We each independently came up with the same maximum likelihood estimator for the value function of the demonstrator that allows for the computation of the performance function in closed-form. However, in contrast to [32] we provide a multitude of extensions for the inverse optimal control problem as well as treating the problem of synthesizing optimal imitative behavior. In addition, a second approach to inverse optimal control for continuous systems was independently developed by both [27] and [2]. While our algorithm is guaranteed to find a performance function that makes the behavior globally optimal, these other approaches [27, 2] only guarantee that the behavior is locally optimal for the performance function. Finally, Ziebart *et. al.* [66] provided an approach for inverse optimal control in Bayesian games, however, the technique did not handle the range of control problems considered in our treatment of the game-theoretic setting.

4.3 Problem Formulation and Basic Approach

We study the finite-horizon continuous time optimal control problem (see Chapter 2 for more details). We observe a series of trajectories sampled from an agent (referred to for the rest of this chapter as the "demonstrator") interacting with a dynamical system. Each trajectory consists of the sequence of states visited and actions performed during the time interval $[0, T]$. While these state-action trajectories are continuous functions through time, we observe samples from these trajectories at discrete timepoints. We define the list of time indices at which we sample the state-action trajectories as $\mathcal{T} = (0 = t_1 < t_2 < \ldots < t_{n_k} = T)$. For simplicity of exposition we assume that the difference between any two consecutive elements from $\mathcal{T}$ is δt, however, this is not required.

Let $\mathcal{D}$ be a set of tuples with each $d_i \in \mathcal{D}$ consisting of a state, x_i, the control signal executed by the demonstrator, u_i, the time at which the sample was taken, $t_i \in \mathcal{T}$, and a trajectory identifier, s_i, which specifies which observed trajectory the sample was taken from. Additionally, we assume that the dynamics of the observed system follow Equation 2.12. In addition to $\mathcal{D}$, we assume that we know the passive dynamics function, $a(\cdot)$, controlled dynamics gain matrix function, $b(\cdot)$, and the noise gain function, $c(\cdot)$, of the demonstrator. In subsequent sections we will relax the requirement that $a(\cdot)$ and $c(\cdot)$ be known exactly (we suggest possible methods to handle uncertainty in $b(\cdot)$, but do not solve the problem in this dissertation). Further, we assume that the performance rate, $r_t(x, u)$ consists of the sum of a quadratic control cost of the form $\frac{1}{2}u^\top qu$ (as described in Section 2.4) with known matrix q, and an arbitrary function of the state. Optionally, the matrix q can depend on time and/or the state. Adding dependence of the matrix q on the state allows for non-quadratic control costs by approximating the control cost locally using its second-order Taylor expansion.

The goal of the algorithm will be to convert the system dynamics model and the observed trajectories from the demonstrator, $\mathcal{D}$, into an estimate of the performance function under which the action choices in $\mathcal{D}$ are optimal. The core of the algorithm is based on the relationship between the optimal control signal and the optimal value function for dynamical systems that follow Equation 2.12.

Recall from the Chapter 2 that the optimal control for this class of systems is given by:

$$u_t^*(x) \quad = \quad q^{-1}b(x)^\top \nabla_x v_t(x) \tag{4.1}$$

We do not know the value function, v_t, *a priori*, however, if the action choices in $\mathcal{D}$ are optimal, then they should approximately satisfy the preceding equation. We can use this fact as an optimization criterion for inferring the demonstrator's value function (we will return to this step in more detail shortly, however, for now assume that v_t can be determined). Once the value function, v_t, has been inferred, the state-dependent component of the performance rate, ρ_t can be computed by rearranging terms in Equation 2.18:

$$\begin{aligned}\rho_t(x) \quad = \quad & \frac{1}{\tau}v_t(x) - \nabla_t v_t(x) - \frac{1}{2}\nabla_x v_t(x)^\top b(x) q^{-1} b(x)^\top \nabla_x v_t(x) - a(x)^\top \nabla_x v_t(x) \\ & -\frac{1}{2} trace\left(c(x)c(x)^\top \nabla_{xx}^2 v_t(x)\right)\end{aligned} \tag{4.2}$$

Since everything on the right-hand side is known, this equation yields a closed-form expression for the demonstrator's performance function. In order to estimate v_t from the demonstrator's behavior, we model the likelihood of an action choice given the value function as a noisy realization of the optimal action given by Equation 4.1:

$$u_t(x) \quad = \quad q^{-1}b(x)^\top \nabla_x v_t(x) + \epsilon, \epsilon \sim N(0, \Sigma_u) \tag{4.3}$$

Where ϵ is an m-dimensional vector of 0-mean gaussian noise with known covariance, Σ_u. The source of this noise is assumed to be due to the demonstrator occasionally deviating from the optimal action prescribed by a particular value function.

Since the elements of $\mathcal{D}$ are likely to contain data from the same trajectory sampled close together in time, it is unreasonable to assume that the noise vectors for each element of $\mathcal{D}$ are uncorrelated. For instance, consider recording the force exerted by a human bicep at a sampling rate of $1,000$Hz. If the bicep is mistakenly contracted in an unplanned way at $t = .401$s then it is more likely to contain a similar unplanned contraction at $t = .402$s. Therefore, we assume we are given a

kernel function, ω, that maps two elements $d_i, d_j \in \mathcal{D}$ to a matrix $\Sigma_{d_i,d_j} \in \mathbb{R}^{2m \times 2m}$ which specifies the covariance structure between the noise values ϵ_i and ϵ_j. For example, the kernel function ω might map d_i and d_j to $\begin{bmatrix} \Sigma_u & 0 \\ 0 & \Sigma_u \end{bmatrix}$ (indicating that the noise for these two observations is uncorrelated) except in the case that the elements were recorded during the same trajectory close together in time.

We now assume, as we did Chapter 2, that the value function at time $t \in \mathcal{T}$ is represented by a linear combination of known basis functions with unknown weights:

$$v_t(x, w_t) = \sum_{i=1}^{d} \phi_{t,i}(x) w_{t,i} \tag{4.4}$$

Plugging the parameterization in Equation 4.4 into Equation 4.5 yields the following relationship between the observed control signals and the basis functions weights:

$$u_t(x) = q^{-1} b(x)^\top \sum_{i=1}^{d} \nabla_x \phi_{t,i}(x) w_{t,i} + \epsilon \tag{4.5}$$

Given this likelihood model, we can infer a distribution over the unknown weights using Bayesian Linear Regression [9]. To accomplish this, we first define some additional notation to refer to the elements of $\mathcal{D}$ sampled at time t as $\mathcal{D}^t$. Next we form the design matrix at time t as:

$$X_t = \begin{pmatrix} q^{-1} b(x_1^t)^\top \nabla_x \phi_{t,1}(x_1^t) & \cdots & q^{-1} b(x_1^t)^\top \nabla_x \phi_{t,d}(x_1^t) \\ \vdots & \ddots & \vdots \\ q^{-1} b(x_{|\mathcal{D}^t|}^t)^\top \nabla_x \phi_{t,1}(x_{|\mathcal{D}^t|}^t) & \cdots & q^{-1} b(x_{|\mathcal{D}^t|}^t)^\top \nabla_x \phi_{t,d}(x_{|\mathcal{D}^t|}^t) \end{pmatrix}$$

The target vector for the linear regression at time t is defined as:

$$g_t = \begin{pmatrix} u_1^t \\ \vdots \\ u_{|\mathcal{D}^t|}^t \end{pmatrix}$$

Let g be a vector consisting of g_t stacked $\forall t \in \mathcal{T}$. Our goal will be to infer the vector w which consists of the w_t stacked $\forall t \in \mathcal{T}$. To this end, we define the

design matrix for the regression, X, as a block diagonal matrix formed from the matrices $X_t, \forall t \in \mathcal{T}$. We construct the noise covariance matrix Ω using the kernel function, ω. In order to apply Bayesian linear regression, we also specify a Gaussian prior over the vector of basis weights, w, consisting of a prior mean μ_0 and prior covariance Σ_0. The covariance matrix, Σ_0, can be used to specify the uncertainty in individual weights or to enforce smoothness of the value function weights (and therefore the value function itself) over time. Once the Gaussian prior, the design matrix, the target vector, and noise covariance matrix are specified, the posterior mean and covariance for the basis weights can be computed in closed form using the well-known Bayesian linear regression formula (see [9] for more details):

$$\mu = \left(X^\top \Omega^{-1} X + \Sigma_0^{-1}\right)^{-1} \left(\Sigma_0^{-1}\mu_0 + X^\top \Omega^{-1} g\right) \tag{4.6}$$

$$\Sigma = \left(X^\top \Omega^{-1} X + \Sigma_0^{-1}\right)^{-1} \tag{4.7}$$

A maximum *a posteriori* (MAP) estimate of the performance rate at time t with $t \neq T$ can be obtained by plugging the basis weights μ into Equation 4.4 to compute the value function and then substituting the resulting value function into Equation 4.2. Note that in general there will be no observations in $\mathcal{D}$ from the terminal time since in the finite-horizon control setting the agent does not perform an action at the terminal time. The only information that we can leverage to compute the terminal value function (and thus the terminal performance function) is that the value function tends to be smooth over time. Therefore, in order to get a sensible estimate of the value function at the terminal time, the covariance structure of the Gaussian prior over basis weights should encode that the basis weights for the terminal value function have significantly high-positive covariance with those from the immediately preceding time index.

Of particular note is that computing the performance function of the demonstrator requires solving a single linear regression problem. This is computationally easier than solving an instance of the forward optimal control problem using the collocation methods described in Chapter 2 (which required either solving a single quadratic regression problem or a series of linear regression problems). The relative efficiency of the inverse vs. the forward problem compares favorably with algorithms for the discrete-state, discrete-action, and discrete-time inverse opti-

mal control problem which typically require solving many instances of the forward control problem [1, 47, 55, 30, 42, 44]. In Section 4.13 we show how to bring the efficiency of our approach for the continuous control problem to the discrete problem.

4.3.1 Online Inference

Since the inference of the posterior distribution over the value function of the demonstrator is accomplished using linear regression, the estimate of this posterior distribution can be updated efficiently without requiring the full computation of the pseudo-inverse. This online update can be performed using the online linear regression method based on the Woodbury-Matrix inversion lemma [9].

4.3.2 The Role of Inverse Dynamical Models

In many cases the assumption that we observe both the states and the actions of the demonstrator may be unrealistic. For instance when analyzing biological or mechanical motion, the control signals will typically consist of generalized forces applied by either the muscles of a biological agent or motors in the case of a mechanical system such as a robot. If the observer is only able to directly observe the state of the demonstrator, rather than these forces, the forces must be inferred in order to apply our algorithm. In order to compute the control signal at time t we model the distribution of the next state as:

$$X_{t+\delta_t} \approx X_t + a(X_t)\delta t + b(X_t)U_t\delta_t + c(X_t)\sqrt{\delta t}Z_t \tag{4.8}$$

Where Z_t is a vector of values drawn from a standard normal distribution. Equation 4.8 is an approximation rather than an equality because $a(\cdot)$, $b(\cdot)$, U_t, and $c(\cdot)$ might not be constant over the time interval $[t, t+\delta t]$. However, given a sufficiently small δt this assumption will hold true. Given the values of two subsequent state measurements x_t and $x_{t+\delta t}$, the value of u_t can be computed from the preceding equation using a weighted linear regression where the target variables are given by $x_{t+\delta_t} - x_t - a(x_t)\delta t$, the design matrix is given by $\frac{1}{\delta t}b(x_t)$ and the covariance matrix

for the errors in the dependent variables is given by $\frac{1}{\delta t} c(x_t) c(x_t)^\top$. The mean and covariance for the resulting estimate of the control signal,u_t, is given by:

$$\begin{aligned} \mu_{u_t} &= \frac{1}{\delta t} \left(b(x_t)^\top \left(c(x_t) c(x_t)^\top \right)^{-1} b(x_t) \right)^{-1} b(x_t)^\top \left(c(x_t) c(x_t)^\top \right)^{-1} \\ &\qquad \times (x_{t+\delta_t} - x_t - a(x_t)\delta t) \qquad (4.9) \\ \Sigma_{u_t} &= \frac{1}{\delta t} \left(b(x_t)^\top \left(c(x_t) c(x_t)^\top \right)^{-1} b(x_t) \right)^{-1} \qquad (4.10) \end{aligned}$$

If we assume independence of noise in the demonstrator's selection of the optimal action and the noise in our estimate of the action, then we can consider both types of uncertainties. This is accomplished by augmenting the noise covariance matrix Ω by adding a block diagonal matrix containing the covariance matrix computed using Equation 4.10 for each estimated action.

4.4 Computing the Uncertainty of the Performance Function

It is difficult to obtain a measure of uncertainty of the performance function with other techniques for inverse optimal control. Previous approaches either produce a point estimate of the performance function [44] (which gives no sense of uncertainty), or else require an expensive Markov Chain Monte Carlo (MCMC) procedure to obtain the samples from the distribution from which the uncertainty can be estimated [47]. The simplicity of our formulation of inverse optimal control allows us to compute the variance (a measure of uncertainty) of the performance function in closed-form. In other words, our algorithm knows when it is certain about the value of the performance function at a particular state and time and knows when it is uncertain, and can determine its own uncertainty in an efficient manner.

Since our algorithm for inverse optimal control produces a Gaussian posterior distribution over the value function of the demonstrator, we can compute the mean and variance of the induced distribution over the performance function in closed-form using standard identities. In order to simplify our derivation, we

define:

$$\lambda_{t,i}(x) = trace\left(c(x)c(x)^\top \nabla^2_{xx}\phi_{t,i}(x)\right) \quad (4.11)$$

Which allows us to rewrite $trace\left(c(x)c(x)^\top \nabla^2_{xx} v_t(x)\right)$ as $\lambda_t(x)^\top w_t$. Next, we rewrite the expression for the performance by collecting terms linear and quadratic in w_t.

$$\begin{aligned}
\rho_t(x) &= \left(\frac{1}{\tau}\phi_t(x) - \frac{\phi_t(x)}{\delta_t} - J_{\phi_t}(x)^\top a(x) - \frac{1}{2}\lambda_t(x)\right)^\top w_t \\
&\quad + \left(\frac{1}{\delta t}\phi_{t-\delta_t}(x)\right)^\top w_{t-\delta_t} - \frac{1}{2} w_t J_{\phi_t}^\top b(x) q^{-1} b(x)^\top J_{\phi_t} w_t \quad (4.12)\\
&= \nu_t(x)^\top w_t + h_{t-\delta_t}(x)^\top w_{t-\delta t} + w_t^\top M(x) w_t \quad (4.13)\\
\nu_t(x) &= \frac{1}{\tau}\phi_t(x) - \frac{\phi_t(x)}{\delta_t} - J_{\phi_t}(x)^\top a(x) - \frac{1}{2}\lambda_t(x) \quad (4.14)\\
h_{t-\delta_t}(x) &= \frac{1}{\delta t}\phi_{t-\delta_t}(x) \quad (4.15)\\
M(x) &= -\frac{1}{2} J_{\phi_t}(x)^\top b(x) q^{-1} b(x)^\top J_{\phi_t}(x) \quad (4.16)
\end{aligned}$$

$$E[\rho_t(x)] = \nu_t(x)^\top \mu_t + h_{t-\delta t}(x)^\top \mu_{t-\delta_t} + \mu_t^\top M(x)\mu_t + trace(M(x)\Sigma_t) \quad (4.17)$$

Where we plugged in a finite-difference approximation of the temporal derivative of the value function, used linearity of expectations, and used the identity that the expected value of a quadratic form $x^\top Ax$ with $x \sim N(\mu, \Sigma)$ is $\mu^\top A\mu + trace(A\Sigma)$. Next, we start from Equation 4.13 to compute the variance of the performance

function.

$$
\begin{aligned}
Var[\rho_t(x)] &= Var\left[\nu_t(x)^\top w_t\right] + Var\left[h_{t-\delta t}(x)^\top w_{t-\delta_t}\right] + Var\left[w_t^\top M(x) w_t\right] \\
&\quad +2Cov\left[\nu_t(x)^\top w_t, h_{t-\delta_t}(x)^\top w_{t-\delta_t}\right] \\
&\quad +2Cov\left[\nu_t(x)^\top w_t, w_t^\top M(x) w_t\right] \\
&\quad +2Cov\left[h_{t-\delta t}(x)^\top w_{t-\delta_t}, w_t^\top M(x) w_t\right] \qquad (4.18) \\
&= \nu(x)^\top \Sigma_t \nu(x) + h_{t-\delta t}^\top(x) \Sigma_{t-\delta_t} h_{t-\delta t}(x) \\
&\quad +2trace\left[M(x)\Sigma_t M(x)\Sigma_t\right] + 4\mu_t^\top M(x)\Sigma_t M(x)\mu_t \\
&\quad +2\nu(x)^\top \Sigma_{t,t-\delta_t} h_{t-\delta t}(x) \\
&\quad +2\nu(x)^\top \Sigma_t M(x)\mu_t \\
&\quad +2h_{t-\delta t}^\top(x)\Sigma_{t-\delta t,t} M(x)\mu_t \qquad (4.19)
\end{aligned}
$$

Where $\Sigma_{t-\delta_t,t}$ is the covariance matrix between the basis weights at time $t-\delta t$ and time t. To achieve our result we applied the following identities for $x \sim N(\mu, \Sigma)$: $Var[x^\top A x] = 2trace\left[A\Sigma A\Sigma\right] + 4\mu^\top A\Sigma A\mu$, and $Cov\left[a^\top x, x^\top B x\right] = 2a^\top \Sigma B\mu$.

4.5 Incorporating Uncertainty in the Dynamics

The assumption that the system dynamics $a(\cdot)$, $b(\cdot)$, and $c(\cdot)$ are known to the observer may be unrealistic. For typical mechanical systems these functions can be quite complex and depend on moments of inertia and Coriollis forces that require substantial knowledge of the characteristics of the motor system of the demonstrator. Here we work out how to handle uncertainty in $a(\cdot)$ and $c(\cdot)$. We discuss the additional difficulties presented by allowing for uncertainty in $b(\cdot)$ and suggest possible solutions.

4.5.1 Uncertainty in the Passive Dynamics

In this case, we do not know exactly the passive dynamics of the demonstrator, but rather that we have a distribution over the function $a(x) \sim N(\mu_a(x), \Sigma_a(x))$ for all states x. The actions of the demonstrator are conditionally independent of

the passive dynamics given its value function. To see this we note that Equation 4.1 does not contain the function $a(\cdot)$. Therefore we proceed to estimate the weights of the basis functions as in the case where we have full certainty about the passive dynamics. The expression for the mean performance value for a particular state given in Equation 4.17 is identical except with $a(x)$ substituted for its mean value:

$$\begin{aligned}
E[\rho_t(x)] &= E\left[\nu_t(x)^\top w_t\right] + h_{t-\delta t}(x)^\top \mu_{t-\delta_t} + \mu_t^\top M(x)\mu_t + trace(M(x)\Sigma_t) \\
&= E\left[-a(x)^\top J_{\phi_t}(x) w_t\right] + \left(\frac{1}{\tau}\phi_t(x) - \frac{\phi_t(x)}{\delta_t} - \frac{1}{2}\lambda_t(x)\right)^\top \mu_t \\
&\quad + h_{t-\delta t}(x)^\top \mu_{t-\delta_t} + \mu_t^\top M(x)\mu_t + trace(M(x)\Sigma_t) \\
&= -\mu_a(x)^\top J_{\phi_t}(x)\mu_t + \left(\frac{1}{\tau}\phi_t(x) - \frac{\phi_t(x)}{\delta_t} - \frac{1}{2}\lambda_t(x)\right)^\top \mu_t \\
&\quad + h_{t-\delta t}(x)^\top \mu_{t-\delta_t} + \mu_t^\top M(x)\mu_t + trace(M(x)\Sigma_t) \\
&= \left(\frac{1}{\tau}\phi_t(x) - \frac{\phi_t(x)}{\delta_t} - J_{\phi_t}(x)^\top \mu_a(x) - \frac{1}{2}\lambda_t(x)\right)^\top \mu_t \\
&\quad + h_{t-\delta t}(x)^\top \mu_{t-\delta_t} + \mu_t^\top M(x)\mu_t + trace(M(x)\Sigma_t) \qquad (4.20)
\end{aligned}$$

This completes our proof since we have arrived at the original expression for the mean performance function except with the mean value of $a(x)$ substituted.

4.5.2 Uncertainty in the Controlled Dynamics

This case of handling uncertainty in the controlled dynamics matrix, $b(\cdot)$, is considerably more difficult and is currently unsolved. To see why, note that in contrast to the relative simplicity of handling uncertainty in the passive dynamics, the controlled dynamics matrix appears in Equation 4.1 causing considerable complications. The simple linear regression to compute the distribution over the basis function weights no longer applies. The reason for this is that ordinary least squares only allows for errors in the response variables and not in the inputs. A solution based on an error in variables model [22] or some sort of particle based sampling method are probably the most promising directions for achieving a solution.

4.5.3 Uncertainty in the Noise Gain Matrix

Handling uncertainty in $c(\cdot)$ follows much the same logic as uncertainty in $a(\cdot)$. Since $c(\cdot)$ does not appear in Equation 4.1 the posterior Gaussian distribution over the basis weights given the observed actions of the demonstrator is unchanged. We note that the expression for $\rho_t(x)$ only depends on $c(x)$ through the vector-valued function $\lambda_t(x)$. We assume in this case that we have a multivariate normal distribution over the elements of $c(x)$. Additionally, we use the notation $c_i(x)$ to refer to the ith column of $c(x)$ we use n_c to refer to the number of columns in $c(x)$. We use the notation μ_{c_i} to refer to the mean of $c_i(x)$ and Σ_{c_i,c_j} to refer to the covariance between $c_i(x)$ and $c_j(x)$.

$$\begin{aligned}
\lambda_{t,i}(x) &= trace\left(c(x)c(x)^\top \nabla^2_{xx}\phi_{t,i}(x)\right) \\
&= trace\left(c(x)^\top \nabla^2_{xx}\phi_{t,i}(x)c(x)\right) \\
&= \sum_{i=1}^{n_c} c_i(x)^\top \nabla^2_{xx}\phi_{t,i}(x)c_i(x)
\end{aligned}$$

Next, we compute how the distribution of λ_t will impact the mean estimate of the performance function:

$$\begin{aligned}
E[\rho_t(x)] &= E\left[\nu_t(x)^\top w_t\right] + h_{t-\delta t}(x)^\top \mu_{t-\delta_t} + \mu_t^\top M(x)\mu_t + trace(M(x)\Sigma_t) \\
&= \left(\frac{1}{\tau}\phi_t(x) - \frac{\phi_t(x)}{\delta_t} - J_{\phi_t}(x)^\top a(x)\right)^\top \mu_t + E\left[\frac{1}{2}\lambda_t(x)^\top w_t\right] \\
&\quad + h_{t-\delta t}(x)^\top \mu_{t-\delta_t} + \mu_t^\top M(x)\mu_t + trace(M(x)\Sigma_t) \\
&= \left(\frac{1}{\tau}\phi_t(x) - \frac{\phi_t(x)}{\delta_t} - J_{\phi_t}(x)^\top a(x)\right)^\top \mu_t + E\left[\frac{1}{2}\lambda_t(x)^\top\right] E\left[w_t\right] \\
&\quad + Cov\left[\frac{1}{2}\lambda_t(x), w_t\right] + h_{t-\delta t}(x)^\top \mu_{t-\delta_t} + \mu_t^\top M(x)\mu_t + trace(M(x)\Sigma_t) \\
&= \left(\frac{1}{\tau}\phi_t(x) - \frac{\phi_t(x)}{\delta_t} - J_{\phi_t}(x)^\top a(x)\right)^\top \mu_t + \frac{1}{2}E\left[\lambda_t(x)\right]^\top \mu_t \\
&\quad + h_{t-\delta t}(x)^\top \mu_{t-\delta_t} + \mu_t^\top M(x)\mu_t + trace(M(x)\Sigma_t) \qquad (4.21) \\
E\left[\lambda_{t,i}(x)\right] &= \sum_{i=1}^{n_c} \left(trace(\nabla_{xx}\phi_{t,i}(x)\Sigma_{c_i,c_i}(x)) + \mu_{c_i}(x)^\top \nabla_{xx}\phi_{t,i}(x)\mu_{c_i}(x)\right)
\end{aligned}$$

Note that we can handle uncertainty in both $a(\cdot)$ and $c(\cdot)$ simultaneously since there are no terms involving products of these two functions.

4.6 Extension to Partially Observable Problems

Next, we consider the problem of determining a performance function for a demonstrator controlling a dynamical system where the state at time t, x_t, is only partially observable. In this setting the demonstrator must base its decisions on sequences of noisy observations that indirectly reflect the state of the system.

Consider an agent that must control a system with the following system and observation dynamics:

$$dX_t = (a(X_t) + b(X_t)U_t)\,dt + C dB_t \tag{4.22}$$

$$dZ_t = o(X_t)dt + D dB'_t \tag{4.23}$$

Where C and D are known matrices that are constant with respect to the state (but can optionally depend on time) and specify the gains for the Brownian motion processes dB and dB'. Z_t is a random process specifying an observation at time t. In comparison with the fully observable case we do not allow the noise in the state dynamics to be state-dependent, however, the deterministic component of the state dynamics is not restricted. Here, we assume that the agent only has access to the observation process and not the state process.

Similarly to the fully-observable case, we assume that the agent is trying to maximize some state-dependent performance function over time with a quadratic cost on the control signal (if the control cost is not quadratic, then we can approximate it locally using its second-order Taylor approximation). Processes such as this are known as continuous time Partially Observable Markov Decision Processes (POMDPs). For any POMDP, it can be shown that the distribution of the underlying system state, x_t, conditioned on the sequence of observations up to time t is sufficient for selecting the optimal control signal at time t. In general, for processes with system and observation dynamics described by Equations 4.22 and 4.23 respectively, it is not easy to maintain in a compact form the exact distribution of states conditioned on the previous observations. Here, we assume that the demonstrator represents the distribution over the underlying system state using a single Gaussian that is updated according to the Continuous Time Extended Kalman Filter (also known as the Extended Kalman-Bucy filter). For a detailed treatment

of the Extended Kalman-Bucy filter see [24]. The basic idea of this filter is to linearize the observation and system dynamics about the current mean estimate of the state, and then apply the standard Kalman-Bucy update equations.

The assumption that the demonstrator updates its beliefs according to the Extended Kalman-Bucy filter allows us to reformulate the partially observable control problem in Equation 4.23 as a fully observable one. The state of the system at time t, x_t, is modified to contain the mean and covariance matrix of a Gaussian estimate of the underlying system state. We use the notation X'_t to refer to a random vector specifying the demonstrator's belief about the true underlying state of the system consisting of the mean, μ^s_t, and covariance, Σ^s_t, of a Gaussian distribution over the underlying state:

$$x'_t = \begin{bmatrix} \mu^s_t \\ vec(\Sigma^s_t) \end{bmatrix}$$

Where *vec* is a function that vectorizes the input matrix column-wise. The evolution of the augmented state can now be described by the following fully observable stochastic process:

$$\begin{aligned} d\begin{bmatrix} \mu^s_t \\ vec(\Sigma^s_t) \end{bmatrix} &= \begin{bmatrix} a(\mu^s_t) \\ vec\left(sym(J_a(\mu^s_t)^\top \Sigma^s_t) + CC^\top - k_t(x'_t)k_t(x'_t)^\top\right) \end{bmatrix} dt \\ &\quad + \begin{bmatrix} b(\mu^s_t) \\ vec\left(sym(J_{b_1}(\mu^s_t)\Sigma^s_t)\right) \quad \dots \quad vec\left(sym(J_{b_m}(\mu^s_t)\Sigma^s_t)\right) \end{bmatrix} U_t dt \\ &\quad + \begin{bmatrix} k_t(x'_t)o(\mu^s_t)^{-1}D \\ 0 \end{bmatrix} dB_t \qquad (4.24) \\ k_t(x'_t) &= \Sigma^s_t J_a(\mu^s_t)^\top \left(o(\mu^s_t)o(\mu^s_t)^\top\right)^{-1} o(\mu^s_t) \qquad (4.25) \\ sym(x) &= x + x^\top \qquad (4.26) \end{aligned}$$

Where J_a is the Jacobian of $a(\cdot)$ with respect to the state and J_{b_i} is the Jacobian of the ith column of the control gain matrix function $b(\cdot)$ with respect to the state. Equation 4.24 specifies a fully observable system with dynamics following the same form as we have treated earlier in the chapter. Thus, the same methods

for computing the perfrormance function and the uncertainty of the performance function for fully observable processes can be applied to this case.

4.7 Issues of Identifiability of the Performance Function

Depending on the particular characteristics of the observed dynamical system, independently of the number of trajectories that we observe from the demonstrator, it may not be possible to uniquely determine the demonstrator's performance function. To see why this is the case, consider a minimalistic example of a demonstrator controlling a point mass in 1-dimension. The state-space of the system is $x \in \mathbb{R}^2$ where the first element of x specifies the position, θ, of the point mass and the second specifies its velocity, $\dot{\theta}$. The control signal, $u \in \mathbb{R}$, specifies a force exerted on the point mass. Additionally, we make our standard assumption that the agent pays a cost proportional to the square of the force exerted on the mass. For simplicity of exposition, we assume that the problem is deterministic (i.e. $c(x) = 0, \forall x \in \mathbb{R}^2$). The dynamics of the system can be written as follows:

$$dX_t = (a(X_t) + b(X_t)U_t)\,dt \tag{4.27}$$

$$d\begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\dot{\theta} \\ 0\end{bmatrix} dt + \begin{bmatrix}0 \\ \frac{1}{m}\end{bmatrix} U_t dt \tag{4.28}$$

We can determine the optimal action of the agent given its value function by plugging in the system dynamics to Equation 4.1.

$$\begin{aligned} u_t^\star(x) &= q^{-1} \begin{bmatrix}0 \\ \frac{1}{m}\end{bmatrix}^\top \begin{bmatrix}\nabla_\theta v_t(x) \\ \nabla_{\dot{\theta}} v_t(x)\end{bmatrix} \\ &= \frac{1}{qm} \nabla_{\dot{\theta}} v_t(x) \end{aligned} \tag{4.29}$$

Of particular importance is that optimal control signal $u_t^\star(x)$ is unaffected by $\nabla_\theta v_t(x)$. Therefore, we are free to choose the value of $\nabla_\theta v_t(x)$ arbitrarily without affecting the goodness of fit of the value function to the demonstrator's action choices. In particular, given a value function that maximizes the goodness of fit to

the demonstrator actions, $v_t^\star$, we can create a new value function $v_t^{\star'}(x) = v_t^\star(x) + s_t(\theta)$, where s_t is an arbitrary time-dependent function of θ. Since $\nabla_{\dot{\theta}} v_t^{\star'}(x) = \nabla_{\dot{\theta}} v_t^\star(x) + 0 = \nabla_{\dot{\theta}} v_t^\star(x)$, we can guarantee that $v_t^{\star'}$ will be as good a fit to the demonstrator's action choices as $v_t^\star$. The performance function estimate for the agent can be determined from Equation 4.2.

$$\begin{aligned}
\rho_t(x) &= \frac{1}{\tau} v_t(x) - \nabla_t v_t(x) - \frac{1}{2} \nabla_x v_t(x)^\top b(x) q^{-1} b(x)^\top \nabla_x v_t(x) - a(x)^\top \nabla_x v_t(x) \\
&= \frac{1}{\tau} \left(v_t^\star(x) + s_t(\theta) \right) - \nabla_t \left(v_t^\star(x) + s_t(\theta) \right) - \frac{1}{2} \left(\frac{1}{qm} \frac{\partial v_t^\star(x)}{\partial \dot{\theta}} \right)^2 \\
&\quad -\dot{\theta} \left(\nabla_\theta v_t^\star(x) + \nabla_\theta s_t(\theta) \right) \qquad (4.30)
\end{aligned}$$

Therefore, different choices for s_t will lead to different performance functions for the demonstrator even though each will be an equally valid explanation of the demonstrator's behavior. In the next section, we will discuss how to leverage prior knowledge about the structure of the performance function of the demonstrator to overcome this problem.

4.8 Incorporating Prior Knowledge About the Performance Function

Next, we propose two methods for overcoming the performance function identifiability issues described in the previous section. Both methods rely on leveraging prior knowledge concerning features that are likely to be implicated in the demonstrator's performance function. We use these features as a regularizer for inferring the demonstrator's value function. There are two approaches to achieving this. The first method is to fit the demonstrator's value function in two steps. In step 1 we choose a value function that maximizes the goodness of fit to the demonstrator's action choices. In step 2 we construct an additional function that when added to the value function from step 1 does not change the demonstrator's optimal actions, but instead maximizes the closeness of the demonstrator's performance function to a linear combination of the known performance features. The second method involves performing steps 1 and 2 jointly. This formulation allows

for a tradeoff between goodness of fit to the observed action choices and parsimony with the performance features.

We now assume that in addition to the basis functions for approximating the value function of the agent, $\phi_{t,i}$, we are also given a set of performance features, $\alpha_{t,i}$. Additionally, we will use the notation $\alpha_t(x)$ to refer to the vector of performance features evaluated at the state x. We assume that performance functions that can be well-represented by a linear combination of these performance features are more likely goals for the demonstrator.

4.8.1 Method 1

Here, we assume that we are given features, $\zeta_{t,i}$, to approximate a secondary value function for the demonstrator. In contrast to the features for approximating the primary value function, we require that:

$$q^{-1}b(x)^\top \nabla_x \zeta_{t,i}(x) = 0, \forall t \in [0,T], \forall x \in \mathbb{R}^n \tag{4.31}$$

If the preceding equation is satisfied, we can construct a new value function v'_t by adding an arbitrary function v_t and any linear combination of the features, $\zeta_{t,i}$, without changing the prescribed optimal actions of the value function v_t.

$$\begin{aligned} u^\star_t(x) &= q^{-1}b(x)^\top \nabla_x \left(v_t(x) + \sum_j w_{\zeta,t,j}\zeta_{t,j}(x) \right) \\ &= q^{-1}b(x)^\top \nabla_x v_t(x) + \sum_j w_{\zeta,t,j} q^{-1}b(x)^\top \nabla_x \zeta_{t,j}(x) \\ &= q^{-1}b(x)^\top v_t(x) \end{aligned}$$

Let v_t be a value function that optimally predicts the demonstrator's actions computed using the procedure in Section 4.3. Our goal is now to compute weights, $w_{\zeta,t,i}$, for each basis function $\zeta_{t,i}$. The final value function will be given by $v'_t(x) = v_t(x) + \zeta_t(x)^\top w_{\zeta,t}$ where we use $\zeta_t(x)$ to refer to the vector valued function of the basis functions given by $\zeta_{t,i}$. The final estimate of the performance function will be determined by plugging $v'_t(x)$ into Equation 4.2. We assume that we are given a set of states $\mathbf{x_t}$ for each time index $t \in \mathcal{T}$ to use to fit the weights w_ζ. These states could be the same as the states in which the demonstrator was observed, or

could be selected using another criterion. We determine the optimal basis weights, w_ζ according to the following equation:

$$w_\zeta^\star = \arg\min_{w_\zeta} \kappa(v_t, w_\zeta) \tag{4.32}$$

$$\begin{aligned}
\kappa(v, w_\zeta) = \min_{w_\alpha} \Bigg\{ & \sum_{t \in \mathcal{T} \setminus T} \sum_{x \in \mathbf{x_t}} \Bigg(\alpha_t(x)^\top w_{\alpha,t} \\
& - \Bigg(\frac{1}{\tau} v_t(x) + \frac{1}{\tau} \zeta_t(x)^\top w_{\zeta,t} - \nabla_t v_t(x) \\
& \quad - \frac{\zeta_{t+\delta t}(x)^\top w_{\zeta,t+\delta t} - \zeta_t^\top(x) w_{\zeta,t}}{\delta t} \\
& \quad - \frac{1}{2} \nabla_x v_t(x)^\top b(x) q^{-1} b(x)^\top \nabla_x v_t(x) \\
& \quad - a(x)^\top \nabla_x v_t(x) - a(x)^\top J_{\zeta,t}(x) w_{\zeta,t} \\
& \quad - \frac{1}{2} trace\big(c(x) c(x)^\top \nabla_{xx}^2 v_t(x) \big) \\
& \quad - \frac{1}{2} trace\big(c(x) c(x)^\top \sum_j \nabla_{xx}^2 \zeta_{t,j}(x) w_{\zeta,t,j} \big) \Bigg)^2 \\
& + \sum_{x \in \mathbf{x_T}} \big(\alpha_T(x)^\top w_{\alpha,T} - \big(v_T(x) + \zeta_T(x)^\top w_{\zeta,T} \big) \big)^2 \Bigg\}
\end{aligned} \tag{4.33}$$

In the objective function we separated the terminal time T from other time indices due to the differing structure of the HJB equation for this time index. Also, note that since v is considered fixed for this optimization, the only unknowns in the optimization problem are the weights w_α and w_ζ which only appear linearly within the squared objective function. Thus, the optimization in Equation 4.33 is an Ordinary Least-Squares (OLS) problem and can be solved with standard methods.

In general the posterior distribution over the value function v and the basis weights w_α will not be jointly Gaussian. In order to deal with this problem one can compute the Hessian of the objective function given in Equation 4.34 and apply LaPlace's approximation [9] to obtain a Gaussian estimate of the posterior distribution over the basis weights. If one only wishes to determine the variance of the performance function, this likely can be done in closed form. To see this, note that the solution to the linear regression problem will involve multiplying a fixed pseudo-inverse with a target vector that changes depending on the demonstrator's

value function either quadratically or linearly. Since the posterior distribution over the demonstrator's value function, v_t, is Gaussian, the estimate of w_ζ will be a linear combination of linear and quadratic forms in v_t and the variance of this linear combination can be computed using techniques similar to Section 4.4. We leave the computation of the variance of the performance function in this setting as future work.

4.8.2 Method 2

The first method for incorporating prior knowledge of the performance structure allows us to modify the performance function estimate provided that we don't worsen the fit to the demonstrator's action choices. However, sometimes it may be preferable to allow for a tradeoff between making the demonstrator's action choices fit less well and making the estimated performance function closer to one from the linearly parameterized family. There are two principal reasons for this. First, if the demonstrator's actions are somewhat noisy, then incorporating prior information about the performance structure may increase robustness. Second, if the features used to represent the value function are local, as is the case with the features described later in Section 4.11 (meaning their influence is confined to a local region of the state space), then we will have no way to estimate the performance function of the agent in parts of the state space that we have yet to observe demonstrator behavior.

In order to balance fitting the demonstrator's actions well and having a parsimonious performance function, we simply add the objective function from Method 1 multiplied by a weight term with the original objective function (i.e. when we do not enforce prior information).

$$w^\star \quad = \quad \arg\min_w \left\{ \frac{1}{\varphi}\kappa(v(w), \mathbf{0}) + \sum_{(x,u,t,s)\in\mathcal{D}} \left(u - q^{-1} b(x)^\top J_{\phi_t}(x) w_t\right)^2 \right\} \quad (4.34)$$

Where φ is a positive constant indicating the desired tradeoff for fitting a parsimonious performance function and fitting the decision-making agent's actions well, $\mathbf{0}$ is a vector of all zeroes of the appropriate dimensionality, and we use $v(w)$ as a

shorthand notation to indicate the value function given by the basis weights w. In contrast to Method 1 which requires solving two sequential linear regression problems and therefore always converges to a unique solution, the objective function in Equation 4.34 involves solving a single quadratic regression problem. To see this note that the function $\kappa(\cdot,\cdot)$ contains the term $\frac{1}{2}\nabla_x v_t(x)^\top b(x) q^{-1} b(x)^\top \nabla_x v_t(x) = \frac{1}{2} w_t^\top J_{\phi,t}(x) b(x) q^{-1} b(x)^\top J_{\phi,t}(x)^\top w_t$ which is quadratic in w_t.

Since the determination of the value function weights is no longer a linear least-squares problem, we no longer have the nice property that the variance of the basis weights can be computed in closed form. A possible workaround is to compute the Hessian of the objective function given in Equation 4.34 and apply LaPlace's approximation [9] to obtain a Gaussian estimate of the posterior distribution over the basis weights.

4.9 Goal-based imitation for Mechanical and Motor Systems

For a large class of biological and mechanical systems known as "natural systems" (defined as ones where the kinetic energy of the system is $\frac{1}{2}\dot{\theta}^\top m(\theta)\dot{\theta}$) the state of the system can be represented as $x = \begin{bmatrix}\theta \\ \dot{\theta}\end{bmatrix}$ where θ and $\dot{\theta}$ denote the position and velocity respectively of the system in some generalized coordinate system. For instance, θ might represent the joint angles of a robot and $\dot{\theta}$ might represent the angular velocities of these joint angles. For the rest of this section we will refer to θ and $\dot{\theta}$ as "position" and "velocity" for concreteness. For such systems, the dynamics take the following form [23, 35]:

$$f(\theta,\dot{\theta})U_t + c(\theta_t,\dot{\theta}_t)dB_t \;=\; m(\theta_t)\ddot{\theta}_t + \mathcal{C}(\theta_t,\dot{\theta}_t)\dot{\theta}_t + \tau(\theta) \tag{4.35}$$

Where $\mathcal{C}$ is the position and velocity-dependent coriolis-matrix, m is the position-dependent inertial matrix, τ are generalized-forces exerted due to gravity, and f specifies a transformation from the control signals of the agent into generalized torques about the joint angles. Since the inertial matrix m is always positive

definite, we can rewrite the equations of motion given by Equation 4.35 in the standard form of the systems we have studied so far in this chapter:

$$d\begin{bmatrix}\theta_t \\ \dot{\theta}_t\end{bmatrix} = \left(\begin{bmatrix}\dot{\theta}_t \\ -m^{-1}(\theta_t)\left(\mathcal{C}(\theta_t,\dot{\theta}_t)\dot{\theta}_t + \tau(\theta)\right)\end{bmatrix} + \begin{bmatrix}0 \\ m^{-1}(\theta_t)f(\theta,\dot{\theta})\end{bmatrix} U_t\right) dt + \begin{bmatrix}0 \\ m^{-1}(\theta)c(\theta_t,\dot{\theta}_t)\end{bmatrix} dB_t \tag{4.36}$$

Of particular importance is that many of the components of the dynamics in the previous equation do not depend on the particulars of the natural system. For instance, for any natural system we require that the top-half of the passive dynamics is always $\dot{\theta}$. Also, we require that the elements of the top half of the noise and controlled dynamics matrices are all zero. This observation will become critical when we examine the problem of goal-based imitation for natural systems.

Since the system the demonstrator is assumed to be interacting with is a control-affine diffusion (with dynamics given by Equation 4.36), we can compute a closed-form expression for the demonstrator's performance function in terms of its value function. The performance rate will be a function of θ, $\dot{\theta}$, and time, and therefore we assume that the demonstrator's performance is dictated by the positions and velocities of its joint angles. Next, since we model the control problem faced by the imitator as a control-affine diffusion with an unknown performance function, we can now substitute the closed-form expression for the demonstrator's performance function into the imitator's HJB equation. This substitution encodes the assumption that the imitator and the demonstrator derive the same performance value for being in a particular state configuration. The result of making this substitution will be a new PDE, very similar to the original HJB, except that it will include the value function of the demonstrator and will not contain an explicit performance function. Next, we use collocation methods to compute an approximate solution to the resulting PDE. Recall that collocation methods (see Chapter 2) are an approach for generating approximate solutions to PDEs, and that solving the HJB for a control-affine diffusion yields the optimal control law given in Equation 4.1.

Before proceeding with the derivation we define some additional notation.

In order to differentiate between the functions in Equation 4.36 that specify the dynamics of the imitator and the demonstrator, we use a tick symbol, $'$, to identify those that correspond to the imitator. For instance, we use v_t to refer to the value function of the demonstrator at time t. Similarly, we use v'_t to refer to the value function of the imitator at time t.

We begin by plugging $b(x) = \begin{bmatrix} 0 \\ m^{-1}(\theta) f(\theta_t, \dot{\theta}_t) \end{bmatrix}$ into Equation 4.5 to determine a likelihood function for the demonstrator's value function. Next, we apply the procedures in this chapter to compute a MAP estimate of the demonstrator's value function given the observed behavior. Next, we substitute the expression for the demonstrator's performance function, given by Equation 4.2, into Equation 4.2 to generate a new PDE.

$$\begin{aligned} -\nabla_t v'_t(x) &= \rho_t(x) - \frac{1}{\tau} v'_t(x) + \frac{1}{2} \nabla_x v'_t(x)^\top b'(x) q'^{-1} b'(x)^\top \nabla_x v'_t(x) \\ &\quad + a'(x)^\top \nabla_x v'_t(x) + \frac{1}{2} trace(c'(x) c'(x)^\top \nabla^2_{x,x} v'_t(x)) \\ \rho_t(x) &= \frac{1}{\tau} v_t(x) - \nabla_t v_t(x) - \frac{1}{2} \nabla_x v_t(x)^\top b(x) q^{-1} b(x)^\top \nabla_x v_t(x) \\ &\quad - a(x)^\top \nabla_x v_t(x) - \frac{1}{2} trace\left(c(x) c(x)^\top \nabla^2_{x,x} v_t(x)\right) \end{aligned}$$

Unfortunately, the new PDE contains the term $\nabla_\theta v_t(x)$ (through the dependence on the performance rate $\rho_t(x)$) which cannot be recovered using the MAP estimate for fitting the demonstrator's value function (see Section 4.7 for an example of why this gradient direction cannot be determined). Next, we show how to simplify the PDE to remove all of these unidentifiable terms.

Instead of identifying an optimal value function for the imitator from scratch, we can define without loss of generality the imitator's value function as: $v'_t(x) = v_t(x) + g_t(x)$ where v_t is the estimate of the demonstrator's value function given by Equation 4.5 and g_t is a supplementary value function that we will fit in order to solve the PDE. Next, we write the PDE in terms of our new construction for

the imitator's value function:

$$\begin{aligned}
-\nabla_t v_t(x) &- \nabla_t g(x,t) = \rho_t(x) - \frac{1}{\tau}\left(v_t(x) + g_t(x)\right) \\
&+\frac{1}{2}\left(\nabla_x v_t(x) + \nabla_x g_t(t)\right)^\top b'(x) q'^{-1} b'(x)^\top \left(\nabla_x v_t(x) + \nabla_x g_t(x)\right) \\
&+a'(x)^\top \left(\nabla_x v_t(x) + \nabla_x g_t(x)\right) \\
&+\frac{1}{2} trace(c'(x)c'(x)^\top \left(\nabla^2_{x,x} v_t(x) + \nabla^2_{x,x} g_t(x)\right)) \qquad (4.37) \\
\rho_t(x) &= \frac{1}{\tau} v_t(x) - \nabla_t v_t(x) - \frac{1}{2}\nabla_x v_t(x)^\top b(x) q^{-1} b(x)^\top \nabla_x v_t(x) \\
&\quad -a(x)^\top \nabla_x v_t(x) \\
&\quad -\frac{1}{2} trace\left(c(x)c(x)^\top \nabla^2_{x,x} v_t(x)\right) \qquad (4.38)
\end{aligned}$$

Now, we make two algebraic simplifications. Note, that these simplifications do not imply that we are making any additional assumptions. First, we can cancel out the temporal derivative and $\frac{1}{\tau}$ term of the value function of the demonstrator, v_t, from both sides of Equation 4.37 (after substituting in the form of $\rho_t(x)$). Second, we notice that:

$$\begin{aligned}
a(x)^\top \nabla_x v_t(x) &= \dot{\theta}^\top \nabla_\theta v_t(x) - \left(m^{-1}(\theta_t)\left(\mathcal{C}(\theta_t, \dot{\theta}_t)\dot{\theta}_t + \tau(\theta)\right)\right)^\top \nabla_{\dot{\theta}} v_t(x) \\
a'(x)^\top \nabla_x v_t(x) &= \dot{\theta}^\top \nabla_\theta v_t(x) - \left(m'^{-1}(\theta_t)\left(\mathcal{C}'(\theta_t, \dot{\theta}_t)\dot{\theta}_t + \tau'(\theta)\right)\right)^\top \nabla_{\dot{\theta}} v_t(x)
\end{aligned}$$

Therefore, the term $\dot{\theta}^\top \nabla_\theta v_t(x)$ appears on both sides of Equation 4.37 and can be cancelled. After these modifications we arrive at the simplified form of Equation 4.37:

$$
\begin{aligned}
-\nabla_t g_t(x) \;=\; & -\frac{1}{2}\nabla_{\dot\theta} v_t(x)^\top m^{-1}(\theta_t) f(\theta,\dot\theta) q^{-1} \left(m^{-1}(\theta_t) f(\theta,\dot\theta)\right)^\top \nabla_{\dot\theta} v_t(x) \\
& + \left(m^{-1}(\theta_t)\left(\mathcal{C}(\theta_t,\dot\theta_t)\dot\theta_t + \tau(\theta)\right)\right)^\top \nabla_{\dot\theta} v_t(x) \\
& - \frac{1}{2} trace\left(m^{-1}(\theta) c(\theta_t,\dot\theta_t) c(\theta_t,\dot\theta_t)^\top m^{-1}(\theta)^\top \nabla^2_{\dot\theta,\dot\theta} v_t(x)\right) + \frac{1}{\tau} g_t(x) \\
& + \frac{1}{2}\left(\nabla_{\dot\theta} v_t(x) + \nabla_{\dot\theta} g_t(t)\right)^\top m'^{-1}(\theta_t) f'(\theta,\dot\theta) q'^{-1} \\
& \qquad \times \left(m'^{-1}(\theta_t) f'(\theta,\dot\theta)\right)^\top \left(\nabla_{\dot\theta} v_t(x) + \nabla_{\dot\theta} g_t(x)\right) \\
& - \left(m'^{-1}(\theta_t)\left(\mathcal{C}'(\theta_t,\dot\theta_t)\dot\theta_t + \tau'(\theta)\right)\right)^\top \nabla_{\dot\theta} v_t(x) + a'(x)^\top \nabla_x g_t(x) \\
& + \frac{1}{2} trace\Big(m'^{-1}(\theta) c'(\theta_t,\dot\theta_t) c'(\theta_t,\dot\theta_t)^\top m'^{-1}(\theta)^\top \\
& \qquad \times \nabla^2_{\dot\theta,\dot\theta}\left(v_t(x) + g_t(x)\right)\Big) \qquad (4.39)
\end{aligned}
$$

The key property of this PDE is that nowhere in the equation does the gradient of the demonstrator's value function with respect to θ appear. To complete our model of goal-based imitation, we use the collocation methods described in Chapter 2 to find a solution to the PDE in Equation 4.39. By construction, the value function $v'_t(x)$ will yield a policy that will be approximately optimal for the same goal as the demonstrator. This result holds even though we can't say exactly what that goal is.

4.9.1 Accounting for Uncertainty in Goal-Based Imitation

In general the imitator will not be able to determine the gradient of the demonstrator's value function with respect to $\dot\theta$ exactly. However, the procedure for fitting the value function to the demonstrator's behavior outlined in this chapter provides a Gaussian posterior distribution over these components of the gradient. One can modify the collocation approach for solving PDEs based on solving a series of linear regressions backwards in time (see Chapter 2) from minimizing the squared difference between the left- and right-hand sides of the PDE to instead minimizing the expected squared difference between the left- and right-hand sides. The result of this is a linear regression problem at each step where the dependent variables

have different variances depending on the variance of $\nabla_{\hat{\theta}} v_t(x)$. The specific values of the variance induced by these components of the gradient could be computed using the same techniques as in Section 4.4, however, we leave the derivation of a formula for the variance of these values as future work. Regression with different degrees of noise in the response variables is known as a heteroscedastic regression. The standard method for solving this problem is to perform a weighted least-squares fit to the expected value of the response variables where the weights of each instance are proportional to the reciprocal of the variance of the response variable.

4.10 Inverse Optimal Control of Stochastic Differential Games

The work presented so far can easily be extended to a game-theoretic setting by considering a class of games known as stochastic differential games. A stochastic differential game is a generalization of the continuous time stochastic control problem considered in this chapter. The crucial difference is that in stochastic differential games, multiple decision-making agents specify a control signal at each point in time that has an effect on a shared system state vector. Here, we consider 2-player stochastic differential games, however, the extension to more players is trivial. Consider the following dynamical system:

$$dX_t = \left(a(X_t) + b_1(X_t)U_t^1 + b_2(X_t)U_t^2\right)dt + c(X_t)dB_t \tag{4.40}$$

In comparison to Equation 2.12, there are now two control signals specified at each point in time, U_t^1 and U_t^2, corresponding to the action choices of agent 1 and agent 2 respectively. We assume that agent 1 is attempting to maximize its expected performance which is of the form: $r_t^1(x,u) = \rho_t^1(x) - \frac{1}{2}u^\top q_1 u$. Similarly, we assume that agent 2 is attempting to maximize its expected performance which is of the form: $r_t^2(x,u) = \rho_t^2(x) - \frac{1}{2}u^\top q_2 u$. Where, as in the single-agent case, q_1 and q_2 are known real symmetric positive-definite-matrices.

Suppose we observe trajectories of two agents interacting with this dynam-

ical system. The input $d_i \in \mathcal{D}$ consisting of tuples of state, action, time, and trajectory identifier will now be augmented to contain the action of the second decision-making agent. Our goal will be to recover the state performance rates $\rho_t^1(\cdot)$ and $\rho_t^2(\cdot)$ from $\mathcal{D}$. Our method for doing this in the single agent case was to assume that the demonstrator was selecting its action choices optimally with respect to its performance function. In the game-theoretic setting we assume that the agents are at a Nash Equilibrium [45].

A Nash-Equilibrium is achieved when each agent cannot improve its outcome by modifying its strategy given that it knows the other agent's strategy and that the other agent's strategy is fixed. In the case of Stochastic Differential Games, the strategy space consists of all time-varying polices which map states into actions. The Nash Equilibrium condition states that in order for two players' policies to be in a Nash equilibrium, each policy must be optimal assuming the other player's policy is known and held fixed.

Our approach follows the same logic as in the single-agent case, we attempt to infer two value functions, one for player 1 and one for player 2, that optimally match each of the player's actions. Fortunately, assuming that the other player's policy is known, Equation 4.40 can be rewritten in a particularly convenient form. For instance, from player 1's point of view:

$$dX_t = \left(a'_t(X_t) + b_1(X_t)U_t^1\right)dt + c(X_t)dB_t \tag{4.41}$$

$$a'_t(X_t) = a(X_t) + b_2(X_t)\pi_t^2(x) \tag{4.42}$$

Where $\pi_t^2(\cdot)$ is the policy of the second agent. Equation 4.41 is an instance of the single-agent continuous time stochastic optimal control problem we have considered earlier in this chapter. From Equation 4.1, we know that given a candidate value function, v_t^1, $\pi_t^1(x) = q_1^{-1}b_1(x)^\top \nabla_x v_t^1(x)$. By plugging the value function for player 1 into Equation 4.2 we can recover the performance function for which the observed

behavior is optimal.

$$\begin{aligned}\rho_t^1(x) &= \frac{1}{\tau}v_t^1(x) - \nabla_t v_t^1(x) - \frac{1}{2}\nabla_x v_t^1(x)^\top b_1(x) q_1^{-1} b_1(x)^\top \nabla_x v_t^1(x) \\ &\quad -a(x)^\top \nabla_x v_t^1(x) \\ &\quad +\nabla_x v_t^2(x)^\top b_2(x) q_2^{-1} b_2(x)^\top \nabla_x v_t^1(x) \\ &\quad -\frac{1}{2} trace\left(c(x)c(x)^\top \nabla_{xx}^2 v_t^1(x)\right) \end{aligned} \tag{4.43}$$

Using the same procedure as before we can determine a performance function for player 2 that makes the inferred value function an optimal value function:

$$\begin{aligned}\rho_t^2(x) &= \frac{1}{\tau}v_t^2(x) - \nabla_t v_t^2(x) - \frac{1}{2}\nabla_x v_t^2(x)^\top b_2(x) q_2^{-1} b_2(x)^\top \nabla_x v_t^2(x) \\ &\quad -a(x)^\top \nabla_x v_t^2(x) \\ &\quad +\nabla_x v_t^1(x)^\top b_1(x) q_1^{-1} b_1(x)^\top \nabla_x v_t^2(x) \\ &\quad -\frac{1}{2} trace\left(c(x)c(x)^\top \nabla_{xx}^2 v_t^2(x)\right) \end{aligned} \tag{4.44}$$

As in the single-agent case, our objective will be to tune these value functions, v^1 and v^2, to maximize the fit to each player's observed action choices. Crucially, the optimal action choice for a particular value function (e.g. for player 1) is $q_1^{-1} b_1(x)^\top \nabla v_t^1(x)$ which does not depend on the value function of the other player. Therefore, we can solve two separate optimization problems, which are each ordinary least-squares problems, in order to find the optimal basis weights w_1 and w_2 for the linearly parameterized value functions v_1 and v_2. However, while the particular value function inferred for player 1 will not depend on player 2's behavior, the estimated performance function will depend on player 2's behavior through the term $\nabla_x v_t^2(x)^\top b_2(x) q_2^{-1} b_2(x)^\top \nabla_x v_t^1(x)$ in Equation 4.43.

If we want to enforce prior knowledge about the performance functions of both agents, we can follow steps similar to those outlined in Section 4.8. The resulting algorithm involves solving for the value functions of each player in a joint optimization rather than two independent optimizations as is the case when we do not enforce prior information. As before, we will pay a computational price for this change by requiring the solution to a quadratic least-squares rather than a linear least-squares problem.

4.11 Features for Value Function Representation

In some rare cases, we will know in advance a suitable set of basis functions for representing the demonstrator's value function. For instance, if the dynamical system is linear (i.e. $a(\cdot)$ is a linear function and $b(\cdot)$ and $c(\cdot)$ are constant functions) and we assume the state-performance rate, ρ_t, is quadratic, then we can guarantee that the demonstrator's value function is quadratic in the state variables. In cases such as this, we can parameterize the value function using a vector containing all products and co-products of the state-variables. In other words we use basis functions $\phi_i(x) = x_j x_k, \forall j, k \leq n$. To allow for performance functions that are quadratic plus a linear term, we also add additional basis functions linear in the state-variables.

However, in most cases, we will not know ahead of time what an appropriate functional form of the value function will be. Next, we propose a very general parameterization that will allow for the representation of a rich set of value functions. Our approach is to use a collection of local value function approximators that are averaged using state-sensitive weightings defined by a kernel function:

$$v_t(x) \quad = \quad \alpha_t(x)^\top w_t \phi(x) \tag{4.45}$$

Where $\phi(x)$ is a vector of basis functions designed to accurately represent the value function in a local region. Here, w_t is a matrix rather than a vector and specifies the weighting of the basis features for each of the local approximators. Finally, $\alpha_t(x)$ specifies the relative weighting of each of the local approximators for determining the value function. Specifically, we define α_t using the locations of a set of vectors $\mu_{t,1} \ldots \mu_{t,n_k}$ and a kernel function, k. The i^{th} element of $\alpha_t(x)$ takes the following form:

$$(\alpha_t(x))_i \quad = \quad \frac{k(x, \mu_{t,i}, \sigma)}{\sum_{j=1}^{n_k} k(x, \mu_{t,j}, \sigma)} \tag{4.46}$$

$$k(x, \mu, \sigma) \quad = \quad \exp\{-(x-\mu)'\sigma(x-\mu)\} \tag{4.47}$$

Where σ is a fixed, symmetric positive-definite matrix. Since σ is fixed, the value function is fully specified by the vectors $\mu_{t,1:n_k}$ and by the weight vectors $w_{t,1:n_k}$. The positions of the vectors $\mu_{t,1:n_k}$ can be set before optimizing the value function

(preserving the linear nature of the optimization) using the k-means algorithm on the observed states from $\mathcal{D}^t$. In our experience, a particularly good choice for ϕ is to have it contain all products and co-products of the state dimensions plus the terms linear in each of the state dimensions. In this way, the value function is approximated by a mixture of many local quadratic value functions.

4.12 Potential Applications to the Study and Synthesis of Social Interactions

Later, we present an example of using our techniques for analyzing social interaction. However, while this application is promising, we have barely scratched the surface of the potential for applying these new techniques to the computational study of social interaction. In the preceding section, we discussed how the problem of two agents interacting could be conceived of as a stochastic differential game where the agents have a shared state vector (which might encode different physical characteristics of each agent as well as some shared features of the environment) and each agent specifies a control vector at each point in time that, in turn, influences the state differential. This formulation is sufficiently general to handle a wide variety of social interaction settings. Here are a few possibilities.

Socially Intelligent Machines: The ability to infer the intentions behind human movements (e.g. facial expressions, gaze shifts, or gestures) is likely to provide critical information for either a computer program or a robot to successfully interact with humans. The techniques provided here allow for the determination of a human's intention. For example, imagine two factory workers collaborating to assemble a product. In order for the workers to collaborate optimally, each has to have a good idea of the other's intentions. This knowledge is important both so that the workers can sequence relevant subtasks optimally, as well as allowing the workers to help their partner when he encounters particularly difficult parts of the assembly process. The ability to instantiate abilities such as these in a robot, is likely to be crucial for allowing these robots to flexibly interact with human workers.

Computational Analysis of Conversational Dynamics: There is a growing body of work [48] on using the dynamics of conversations or business meetings to detect high-level attributes about the participants (e.g. dominance or influence). By formulating the turn-taking behavior of these situations as a continuous stochastic differential equation where the state variables might include participant eye gaze, body posture, voice volume, etc. we could uncover the intentions of each participant using the techniques in this chapter. Perhaps, the language of intentions will provide a better framework for operationalizing concepts such as influence and dominance than relying on hand labeling of these high-level attributes as has been done in the past. Additionally, this analysis might suggest a computational model of how humans infer these attributes in others.

Computational Analysis of coregulation in mother-infant interaction: The developmental psychologist Alan Fogel put forth a theory that mother-infant interactions are marked by coregulation in which the actions of one partner (e.g. an infant smile) influence the responses of the other partner (e.g. a mother vocalization) [21]. These influences can either be within the same modality (e.g. facial expression to facial expression) or cross-modal (e.g. touch to vocalization). In order to understand interactions such as these at a computational-level we might formulate the unfolding of these behaviors over time as a continuous dynamical system (where the state variables could encode smile intensities, or volume of vocalizations, etc.). The techniques presented in this dissertation could then be used to uncover the intentions that best predict both mother and infant behavior.

Analysis of facial expressions during dialogues: determining at a computational-level what underlies various behaviors that we see in one-on-one conversations would be a fruitful application area for our techniques. For instance, one might build a model of how individual facial movements (e.g. eyebrow movements, gaze shifts, or mouth movements) of a listener affect a speaker and vice-versa. One might then infer whether a particular pattern of movements in either the listener or the speaker was optimal for eliciting particular facial responses in the other.

4.13 Connection to Discrete Inverse Optimal Control

The key feature that makes our approach to Inverse Optimal Control in the continuous case computationally tractable is that the minimization of the right-hand side of the HJB equation with respect to the control signal can be performed analytically. For traditional discrete-time, discrete-state, and discrete-action control problems, this is not the case. The analogous equation to the HJB equation for the discrete control problem is the Bellman optimality equation. Suppose we have a state space $\mathcal{X}$ and an action space $\mathcal{A}$, then the Bellman optimality equation gives us a set of sufficient conditions for a value function to be the optimal value function $v^\star : \mathcal{X} \to \mathbb{R}$:

$$v^\star(x) = \min_{a \in \mathcal{A}} \left(r(x,a) + \gamma \sum_{x \in \mathcal{X}} p(x'|x,a) v^\star(x') \right), \forall x \in \mathcal{X} \tag{4.48}$$

Where, as defined in Chapter 2, γ is a discount factor specifying the preference for long-term vs. short-term performance. Here, we provide a model for inferring the performance function of an agent for a discrete-state, discrete-action, discrete-time MDP. In contrast to the continuous case, the maximization in Equation 4.48 cannot be carried out analytically thus we cannot obtain a maximization-free expression similar to Equation 4.2 that relates the value function and system dynamics to the performance function. More importantly, we cannot obtain a maximization-free objective function for fitting the value function to the demonstrator's behavior. Next, we show that if we assume a particular form for the cost on the action, then we can overcome both of these limitations.

4.13.1 Problem Formulation

We are given a state space, $\mathcal{X}$, an action space, $\mathcal{A}$, a discount factor $\gamma \in [0, 1)$, and a transition dynamics model $T_{x,x',a} = p(x'|x,a)$. We will also write $T_{x,a} \in \mathbb{R}^n$ to denote the vector of transition probabilities from state x after executing action a. Additionally we are given access to a set of demonstration state-action

pairs, $\mathcal{D}$, that we assume were selected optimally with respect to an unknown performance function.

Our first goal is to determine the performance function that explains the demonstrator's behavior as optimal. We assume an infinite-horizon discounted optimal control formulation with known discount factor $\gamma \in [0, 1)$. That is we assume that the goal of the demonstrator is to compute a controller, π^* such that:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t r(X_t, U_t) \;\middle|\; \pi\right] \tag{4.49}$$

However, instead of allowing the policy π to be a mapping from states to actions as is typical for discrete-state MDPs (see Chapter 2), we assume that the agent's policy specifies a mapping from each state to a discrete probability distribution over the action space $\mathcal{A}$. That is we write the action U_t as $U_{t,1} \ldots U_{t,m}$ with the constraint that $\sum_{i=1}^{m} U_{t,i} = 1$ and $0 \leq U_{t,i} \leq 1 \; \forall \; i \in \{1 \ldots m\}$. We can think of this action as an internal plan that the demonstrator formulates, however, only one of these actions is actually carried out in the environment with the probabilities of each possible action given by the demonstrator's internal action plan. As the observer, we only get access to the particular action in $\mathcal{A}$ that was actually executed. Additionally we make the assumption that the performance function of the demonstrator has the form:

$$r(X_t, U_t) = \rho(X_t) + \frac{1}{\lambda} H[U_t] \tag{4.50}$$

Where $H[U_t]$ is the entropy of the discrete probability distribution defined by U_t, and λ is a constant that controls the relative cost/benefit tradeoff for the agent choosing action plans that are entropic vs. choosing ones that will steer it toward desirable states. One can think of this action cost as an assumption that being deterministic requires effort. Optionally, a cost for executing a particular action in $\mathcal{A}$ can be added to the previous performance function without complicating the derivations that follow. However, for brevity we do not treat the case with action costs on the elements of $\mathcal{A}$ (choosing only to enforce costs on the probability distributions over the elements of $\mathcal{A}$ given by U_t).

$$H[U_t] = -\sum_{i=1}^{m} U_{t,i} \log U_{t,i} \tag{4.51}$$

The function ρ is an arbitrary state desirability function that we will attempt to infer from examples of the demonstrator's behavior. As in the continuous case, our strategy for accomplishing this task will be to infer the demonstrator's value function. Next, we will develop an expression relating the demonstrator's value function to the demonstrator's state desirability function.

4.13.2 Approach

From the Bellman optimality equation we know that for a candidate value function to be an optimal value function it has to satisfy the following equation for all states, x. We use $v(x)$ to refer to the value given to state x, and we write v to refer to the value function of all states expressed as a vector in $\mathbb{R}^{|\mathcal{X}|}$.

$$v(x) = \max_u \left\{ r(x,u) + \gamma \sum_{i=1}^{|\mathcal{A}|} u_i T_{x,a_i}^\top v \right\} \tag{4.52}$$

$$= \rho(x) + \max_u \left\{ \frac{1}{\lambda} H[u] + \gamma \sum_{i=1}^{|\mathcal{A}|} u_i T_{x,a_i}^\top v \right\} \tag{4.53}$$

Where the summation is over all possible actions in $\mathcal{A}$ where we weight their expected future performance by the probability of selection under the discrete probability distribution u. The optimal u is given by the following expression (see proof in Appendix A.1):

$$u_i^*(x) = \frac{e^{\lambda\gamma T_{x,a_i}^\top v}}{\sum_{j=1}^{|\mathcal{A}|} e^{\lambda\gamma T_{x,a_j}^\top v}} \tag{4.54}$$

Now we can substitute $u^\star$ into Equation 4.53 to obtain an expression for

the performance function from the value function, v.

$$\begin{aligned} v(x) &= \rho(x) - \frac{1}{\lambda}\sum_{i=1}^{|\mathcal{A}|}\frac{e^{\lambda\gamma T_{x,a_i}^{\top}v}}{\sum_{j=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_j}^{\top}v}}\log\frac{e^{\lambda\gamma T_{x,a_i}^{\top}v}}{\sum_{j=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_j}^{\top}v}} \\ &\quad +\gamma\sum_{i=1}^{|\mathcal{A}|}\frac{e^{\lambda\gamma T_{x,a_i}^{\top}v}}{\sum_{j=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_j}^{\top}v}}T_{x,a_i}^{\top}v \qquad (4.55) \\ &= \rho(x) + \frac{1}{\sum_{i=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_i}^{\top}v}} \\ &\quad \times\sum_{i=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_i}^{\top}v}\left(\gamma T_{x,a_i}^{\top}v - \frac{1}{\lambda}\log\frac{e^{\lambda\gamma T_{x,a_i}^{\top}v}}{\sum_{j=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_j}^{\top}v}}\right) \\ &= \rho(x) + \frac{1}{\sum_{i=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_i}^{\top}v}} \\ &\quad \times\sum_{i=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_i}^{\top}v}\left(-\frac{1}{\lambda}\log\frac{1}{\sum_{j=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_j}^{\top}v}}\right) \\ &= \rho(x) + \frac{1}{\sum_{i=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_i}^{\top}v}} \\ &\quad \times\sum_{i=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_i}^{\top}v}\left(\frac{1}{\lambda}\log\sum_{j=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_j}^{\top}v}\right) \\ &= \rho(x) + \frac{\frac{1}{\lambda}\log\sum_{j=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_j}^{\top}v}}{\sum_{i=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_i}^{\top}v}} \times\sum_{i=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_i}^{\top}v} \\ &= \rho(x) + \frac{1}{\lambda}\log\sum_{j=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_j}^{\top}v} \qquad (4.56) \end{aligned}$$

After rearranging terms we can write the performance function as:

$$\rho(x) = v(x) - \frac{1}{\lambda}\log\sum_{j=1}^{|\mathcal{A}|}e^{\lambda\gamma T_{x,a_j}^{\top}v} \qquad (4.57)$$

4.13.3 Determining the Performance Function from Behavior

Next, we develop a method for computing a maximum likelihood estimate of v from the state-action pairs, $\mathcal{D}$, collected from the demonstrator. Since we have a formula to compute the probability of an observed action choice, a, at a

particular state, x, given a candidate value function, v, if we assume that the probability of each action choice is conditionally independent of all others given the current state and value function we can write the log-likelihood of the given state-action pairs using the product rule as:

$$\begin{aligned}
\log p(x_{1:|\mathcal{D}|}, a_{1:|\mathcal{D}|}|v) &= \log \Bigg(p(x_1|v)p(a_1|x_1, v) \\
&\qquad \Pi_{i=2}^{|\mathcal{D}|} \Big(p(x_i|x_{1:i-1}, a_{1:i-1}, v) \\
&\qquad\quad p(a_i|x_{1:i}, a_{1:i-1}, v) \Big) \Bigg) \qquad (4.58) \\
&= \log \Bigg(p(x_1)p(a_1|x_1, v) \\
&\qquad \Pi_{i=2}^{|\mathcal{D}|} \Big(p(x_i|x_{i-1}, a_{i-1})p(a_i|x_i, v) \Big) \Bigg) \qquad (4.59)
\end{aligned}$$

Since we are interested in maximizing this log-likelihood as a function of v we can ignore any terms that do not depend on v. Thus:

$$\begin{aligned}
\arg\max_v \log p(x_{1:|\mathcal{D}|}, a_{1:|\mathcal{D}|}|v) &= \arg\max_v \sum_{i=1}^{|\mathcal{D}|} \log p(a_i|x_i, v) \qquad (4.60) \\
\log p(a_i|x_i, v) &= \log \frac{e^{\lambda\gamma T_{x_i,a_i}^\top v}}{\sum_{j=1}^{|\mathcal{A}|} e^{\lambda\gamma T_{x_i,a_j}^\top v}} \qquad (4.61) \\
&= \lambda\gamma T_{x_i,a_i}^\top v - \log \sum_{j=1}^{|\mathcal{A}|} e^{\lambda\gamma T_{x_i,a_j}^\top v} \qquad (4.62)
\end{aligned}$$

This objective function is concave in the optimization variables, v, and thus standard optimization methods will converge to a global optimum. The argument that the objective function is concave is as follows. Since the sum of many concave functions is also concave, if we show that $\log p(a|x, v)$ is concave $\forall a \in \mathcal{A}, x \in \mathcal{X}$ then we can conclude that the right side of Equation 4.60 is concave. $\log p(a|x, v)$ is a linear function minus log-sum-exp composed with a linear function. Log-sum-exp is a well-known convex function [10]. Any convex function composed with an affine function is also convex. Therefore $\log p(a|x, v)$ is a concave function (since

all linear functions are concave) minus a convex function which is guaranteed to be concave.

Now that we have shown that the log-likelihood function is concave, the procedure for inferring an estimate of the performance function given sequences of behavior from a demonstrator is to:

1. Infer a maximum likelihood estimate of the demonstrator's value function.

2. Plug the maximum likelihood estimate of the value function computed in step 1 into Equation 4.57 in order to obtain the performance value of each state.

Optionally, one can enforce prior information about the performance function by adding a term that penalizes performance functions that are improbable *a priori*, however, this modification may or may not preserve the concavity of the resulting optimization problem.

Future work includes considering the case where the dynamical model of the demonstrator is not known exactly. This modification will lead to an error-in-variables regression model that might be able to be approached using techniques similar to an error-in-variables model proposed for logistic regression [13] (given the similarity of the objective function for multinomial logistic regression and that proposed here). In a similar spirit, it may be possible to extend techniques from Bayesian Logistic Regression to determine an approximation of the posterior distribution over value functions [26].

4.13.4 Relation to Previous Work on Discrete Inverse Optimal Control

The work presented in this section is related to two threads in the inverse optimal control literature: Inverse Optimal Control for Linearly Solvable Markov Decision Processes (LMDPs) and Bayesian Inverse Reinforcement Learning. In [18], Dvijotham and Todorov propose a method for inverse reinforcement for LMDPs. In a similar spirit to what we do here, in the LMDP setting the action

is changed to be a probability distribution. However, in contrast to the probability distribution over actions used here, the action in an LMDP is over the next state. However, while this formulation is somewhat unnatural for many MDPs, the authors do provide a procedure for embedding a traditional Markov Decision process as an LMDP. The idea is that once this embedding is done, then the inverse optimal control algorithm for LMDPs can be applied to infer the performance function for the original problem. The principle drawback of this method is that the embedding requires that the number of actions available at each state to be equal to the number of reachable successor states. Our method does not require this assumption. Additionally, in [18] a system of linear equations must be solved to yield the embedded LMDP. Since the objective function for matching the demonstrator's actions is defined on the resulting LMDP the nature of how the objective function on the LMDP relates to the original control problem is obfuscated.

Bayesian Inverse Reinforcement Learning, originally proposed in [47] and extended in [15], uses a very similar likelihood function for the demonstrator's actions to the one we use in our work. In [47] they model the probability of an action given the current state, performance function, and the optimal value function as:

$$p(a|x_i, r, v_r^\star) = \frac{e^{\gamma\lambda T_{x,a}^\top v_r^\star}}{\sum_{j=1}^{|\mathcal{A}|} e^{\gamma\lambda T_{x,a_j}^\top v_r^\star}} \tag{4.63}$$

Where $v_r^\star \in \mathbb{R}^{|\mathcal{A}|}$ is the optimal value function over the performance function r expressed as a vector. Of crucial importance is that the optimal value function $v_r^\star$ is computed *without* taking into account the model of action noise dictated by Equation 4.63. Thus, the model states that the demonstrator computes an optimal policy for the control problem assuming that it can behave deterministically. Then when the agent actually executes its plan it sometimes makes mistakes where the rate of mistakes have to do with the relative value of choosing suboptimal actions compared to the optimal action. However, this begs the question if the agent knew that its actions were not perfectly reliable (i.e. subject to noise), could it possibly construct a better policy? Our approach makes explicit the connection between the likelihood of the observed action choices and the underlying control problem

without introducing a post-hoc likelihood function. Another side benefit of our approach compared to [47] and [15] is that, for these approaches as one searches through the space of performance functions periodically one needs to compute the optimal value function $v_r^\star$ via a potentially expensive dynamic programming step that our approach does not require.

4.14 Experiment: Application to Motion Capture Analysis of Mother-Infant Interaction

The purpose of this experiment is to apply our methods for intention inference to the study of a database of mother-infant motion capture that we have collected. Our technique for Inverse Optimal Control is perfectly suited to this task as it provides a natural framework for a computational-level analysis of infant motor movements. In general, almost any interaction between an infant and either social or nonsocial objects could be formulated within this framework, however, to get started with this endeavor we perform a computational-level analysis of infant head movements in response to movements of the toy by the infant's mother.

There are several important reasons for pursuing a computational analysis of infant head movements. The first is an engineering one. The ability to determine quickly which object a person is tracking could be used to provide crucial information to an assistive robot. The second is a scientific one. The methods developed in this chapter provide a formalization of Daniel Dennett's intentional stance for infant motor movements. Therefore, our techniques provide a principled and flexible method for ascribing intentional language such as "tracking", "attempting", or "reaching" to infants. Having a clear and rigorous notion of what we mean by these terms is likely to organize and expand our taxonomy of infant motor development. A particularly nice benefit of defining intentional terms within a rigorous framework is that we can automatically determine behavioral markers for the same intention in novel situations by simply modifying our description of the underlying control problem. We do not, as is done with conventional methods, need to determine by hand what a particular intentional behavior "ought" to look

Figure 4.1: An example of a typical interaction from our experiment. Here mother is told to get her infant to reach for the orange cube.

like in a novel situation.

Here, we provide a framework for answering quantitative questions regarding infant object tracking in naturalistic interactions. Specifically, we show that our technique segments out portions of the mother-infant motion capture sessions that correspond to infant object tracking. Secondly, we show how our model can be used to provide a way to look at the developmental trends of infant head movements. Here, we study the interaction between mother and infant as a system consisting of infant's head direction and the angle between the infant's head direction and a toy that mother is holding. We leave the study of the higher dimensional movements that can be determined using our motion capture dataset as future work (e.g. movement of infant arms and legs).

4.14.1 Methods

We describe the salient details of the dataset used for our analysis, as well as the particular details of how we applied our framework for inverse optimal control.

Experimental Dataset

Upon arriving at the lab, the infant was dressed in a custom-made motion capture suit consisting of 48 motion markers distributed around the infant's arms, legs, and torso. Mother and infant were brought into a playroom that had been instrumented with 10 PhaseSpace visible light motion capture camera pairs. The infant was strapped to a supportive seat that allowed unrestrained movement of her legs and arms while providing stability around the trunk (see Figure 4.1). Mother was fitted with a head mounted camera to record video from her point of view. In addition, mother was given gloves with motion capture markers as well as a head band instrumented with motion markers to allow for accurate tracking of her head position. The experiment alternated portions of interaction between mother and her infant either with or without the mediation of objects (which were also fitted with motion capture markers).

The sequence of interactions in the experiment was as follows:

1. Three rounds of interleaved 1-minute face-to-face interactions and 1 minute interaction with each of three toys (small cube, large cube, mobile) (total: 6 minutes).

2. Experimenter presents a toy at a set of prescribed locations in an attempt to elicit reaching.

3. Step 1 of the interaction is repeated (i.e. three rounds of interleaved 1-minute face-to-face interactions and 1 minute interaction with each of three toys) (total: 6 minutes).

The dataset consists of 4 subjects seen longitudinally for varying numbers of sessions. Specifically, the number of sessions we have for the four subjects are 9, 15, 5, and 9.

4.14.2 Intentional Model of Infant Head Movements

Here we describe how we formulate the movement of infant's head as well as the toy as a continuous state, continuous action, and continuous time control

problem. Once this formalization has been accomplished, then we can apply our methods for inverse optimal control described earlier in the chapter to infer the infant's intentions. Our motivation for the particular characterization of the state space is to answer the question of whether or not at any particular point in time during the session the infant has the intention of tracking a toy held by mother.

State Space

The state consists of the angle, α, between the infant's head direction and the vector from the infant's head to the toy in the plane of the floor, as well as the angle, θ, between the infant's head direction and a vector normal to the infant's chair (see Figure 4.2).

System Dynamics

We model the system as obeying a stochastic differential equation of the form of Equation 2.12. The system dynamics are:

$$d\begin{bmatrix}\alpha\\ \dot{\alpha}\\ \theta\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\dot{\alpha}\\ 0\\ \dot{\theta}\\ 0\end{bmatrix} dt + \begin{bmatrix}0\\ -1\\ 0\\ 1\end{bmatrix} U_t dt + \begin{bmatrix}0 & 0\\ c_\alpha & -c_\theta\\ 0 & 0\\ 0 & c_\theta\end{bmatrix} dB_t \tag{4.64}$$

Where c_α and c_θ are noise scaling factors associated with random movements of the toy and head respectively. As before, we use the dot notation above a variable to indicate its temporal derivative. The control signal at time t, U_t, represents the angular acceleration of the infant's head in the plane of the floor.

Determining the Infant's Intentions

In order to apply our framework for intention inference, we assume that the performance rate consists of a quadratic penalty on the control signal, U_t, added to a time-varying arbitrary function of the state. Additionally, we do not use temporal discounting of the performance rate, therefore we set $\tau = \infty$. Finally, we

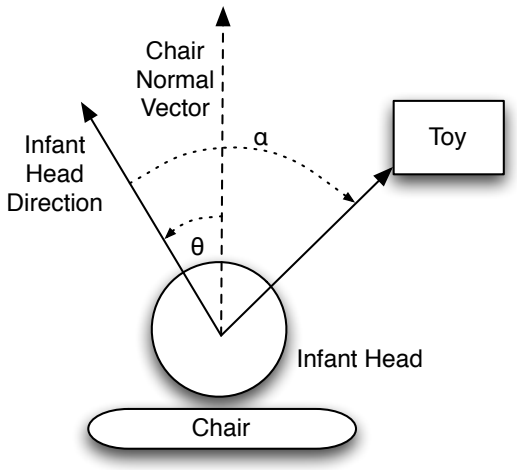

Figure 4.2: A schematic of the control problem faced by the infant. The diagram is drawn from the point of view of looking down at the interaction from the ceiling of the room. At each point in time the infant specifies an angular acceleration for his head direction. The angle of the toy and the x-axis is assumed to evolve according to a Brownian motion process.

set the control penalty matrix q to the value 1. Our goal will be to infer the time-varying state-based component of the performance rate from the head movements of the infant. Plugging our system dynamics into Equation 4.2 we arrive at the following expression for the infant's performance rate:

$$\begin{aligned}\rho_t(x) &= -\nabla_t v_t(x) - \dot{\alpha}\nabla_\alpha v_t(x) - \dot{\theta}\nabla_\theta v_t(x) \\ &\quad -\frac{1}{2}\left(\nabla_{\dot{\alpha}} v_t(x) - \nabla_{\dot{\theta}} v_t(x)\right)^2 \\ &\quad -\frac{1}{2}c_\theta^2\left(\nabla^2_{\dot{\theta},\dot{\theta}} v_t(x) - 2\nabla^2_{\dot{\alpha},\dot{\theta}} v_t(x) + \nabla^2_{\dot{\alpha},\dot{\alpha}} v_t(x)\right) - \frac{1}{2}c_\alpha^2\nabla^2_{\dot{\alpha},\dot{\alpha}} v_t(x)\end{aligned}$$

With optimal action given by:

$$u_t^\star(x) = \nabla_{\dot{\theta}} v(x) - \nabla_{\dot{\alpha}} v(x)$$

We parameterize the value function at time t as a quadratic function of the state variables. To allow for parameters of the quadratic function to vary through time (and thus for the state-dependent component of the performance rate to vary through time), we use multiple quadratic functions each spaced 4 seconds apart

over the time interval $[0, T]$. Each of these quadratic functions is allowed different basis weights. The value at a time point between two of the quadratics is given by linearly interpolating the value functions from the two temporally-closest quadratic functions.

We encode the prior knowledge that the parameter weights of the value function should vary smoothly over time. Also, we use *Method 1* described in Section 4.8 to enforce a prior that the infant's performance function is largely invariant to the products of two distinct state-variables (e.g. $\theta\dot{\theta}$), but rather is mostly dependent upon the square of a single state variable (e.g. θ^2).

By applying our method to a state-action trajectory of infant and object movements, we are able to determine a distribution over the infant's time-varying performance rate function. In order to convert this posterior distribution into the intentional language of "the infant has the intention of tracking the toy" we compute the probability that a performance function sampled from the posterior distribution over performance functions has the property that the infant assigns a positive value to tracking the toy. We estimate this probability by sampling from the Gaussian posterior distribution over value functions.

4.14.3 Results

First, we apply our Inverse Optimal Control algorithm to three sessions. The purpose of this experiment was to see whether our model segments meaningful portions of the session that correspond to our intuitions of what infant object tracking "should" look like. Specifically, we analyze the portion of the session where mother and infant are interacting with a large orange cube. This choice was made because this object was the largest and had the most robust tracking performance. To apply our model, we consider the start of the time horizon for the control problem as when the orange cube is handed to mother, and we consider the terminal time to be when mother hands the toy back to the experiment.

Shown in Figure 4.3 is a sequence of video frames taken from the point in the session where the model assigns the highest probability to the infant having the intention to track the toy. Figure 4.4 is the same as Figure 4.3 except that the

images correspond to the point in the session that the model assigns the lowest probability to the infant having the intention to track the toy. Interestingly, during this portion of the session the infant looks up at mother for a portion of the interval. Two additional pairs of image sequences (one with high evidence of tracking and one with low evidence) for two additional mother-infant sessions are shown in Figures 4.5-4.6 and Figures 4.7-4.8.

Next, we examine how the intentions of infant head movements change over developmental time. We analyze the head movements of one infant, *rob020*, across six motion-capture sessions spaced over a period of 11 weeks. As before, for each session we apply our inverse optimal control model to estimate the probability that the infant intends to track the toy as a function of time. Shown in Figure 4.9 are histograms (over 4 second intervals) showing the probability our model assigns to the infant having the intention of tracking the toy. As the infant gets older there are more periods where the model is certain that the infant intends to track the toy. In order to summarize the developmental trend of toy tracking, we computed the mean proportion of time during play with the toy that the infant intends to toy. The results of this analysis are shown in Figure 4.10. The proportion of time the infant intends to track the toy generally increases with the infant's age. We can interpret this increasing trend either as indicating that either the infant spends more time trying to track the toy as he gets older, or that the infant has become better at executing optimal tracking movements (thus providing clearer evidence of his intention to our model).

4.14.4 Discussion of Motion Capture Results

While existing work tends to focus on the development of tracking *performance* (rather than intention) and to do so in constrained conditions, here, the model is able to determine the developmental trajectory of how much the infant intends to track the toy. In addition, our model is sophisticated enough to be applied in naturalistic situations, such as the data collected here, where the mother is free to manipulate the object in an arbitrary fashion.

Our model gives us a rather simple recipe for determining if the infant's

Figure 4.3: Shown are five frames spaced at intervals of 1 second from the portion of the session where the model assigns the highest probability to the infant intending to track the toy.

Figure 4.4: Shown are five frames spaced at intervals of 1 second from the portion of the session where the model assigns the lowest probability to the infant intending to track the toy.

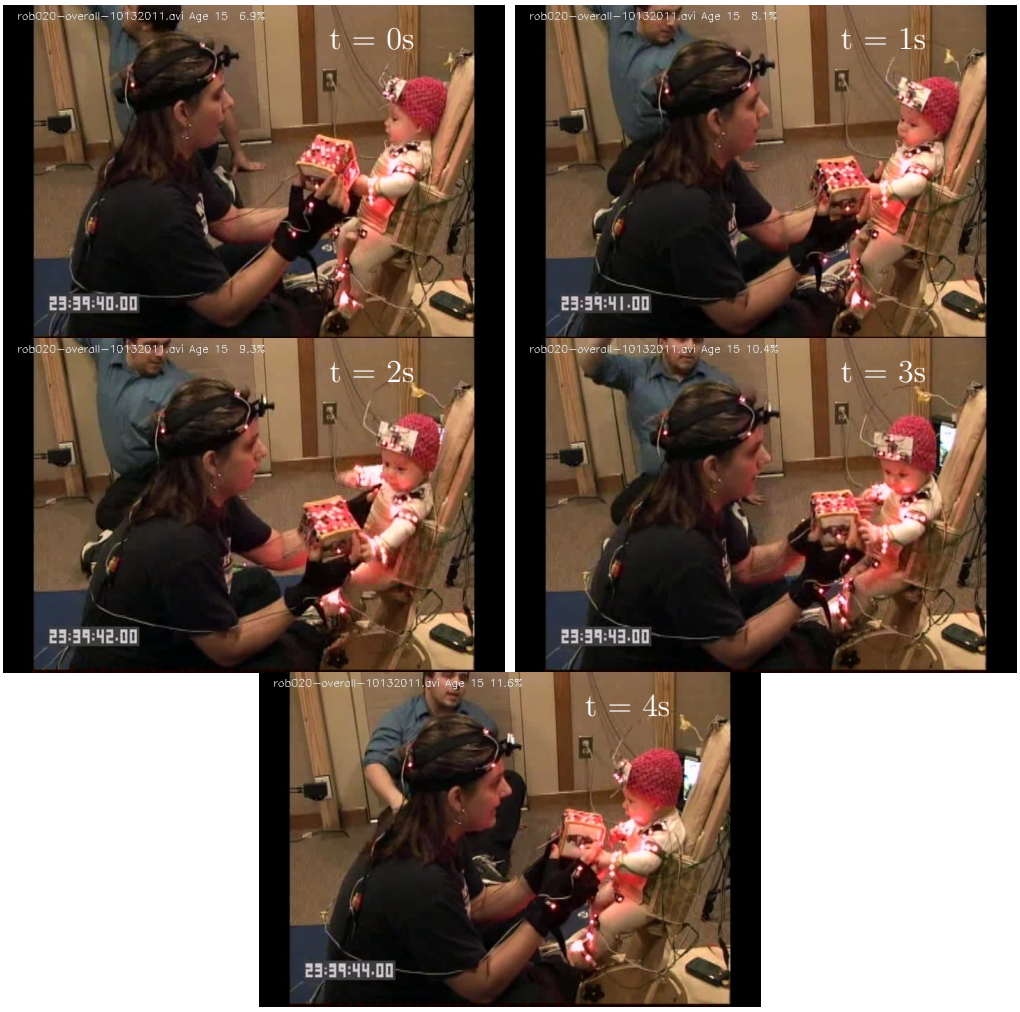

Figure 4.5: Shown are five frames spaced at intervals of 1 second from the portion of the session where the model assigns the highest probability to the infant intending to track the toy.

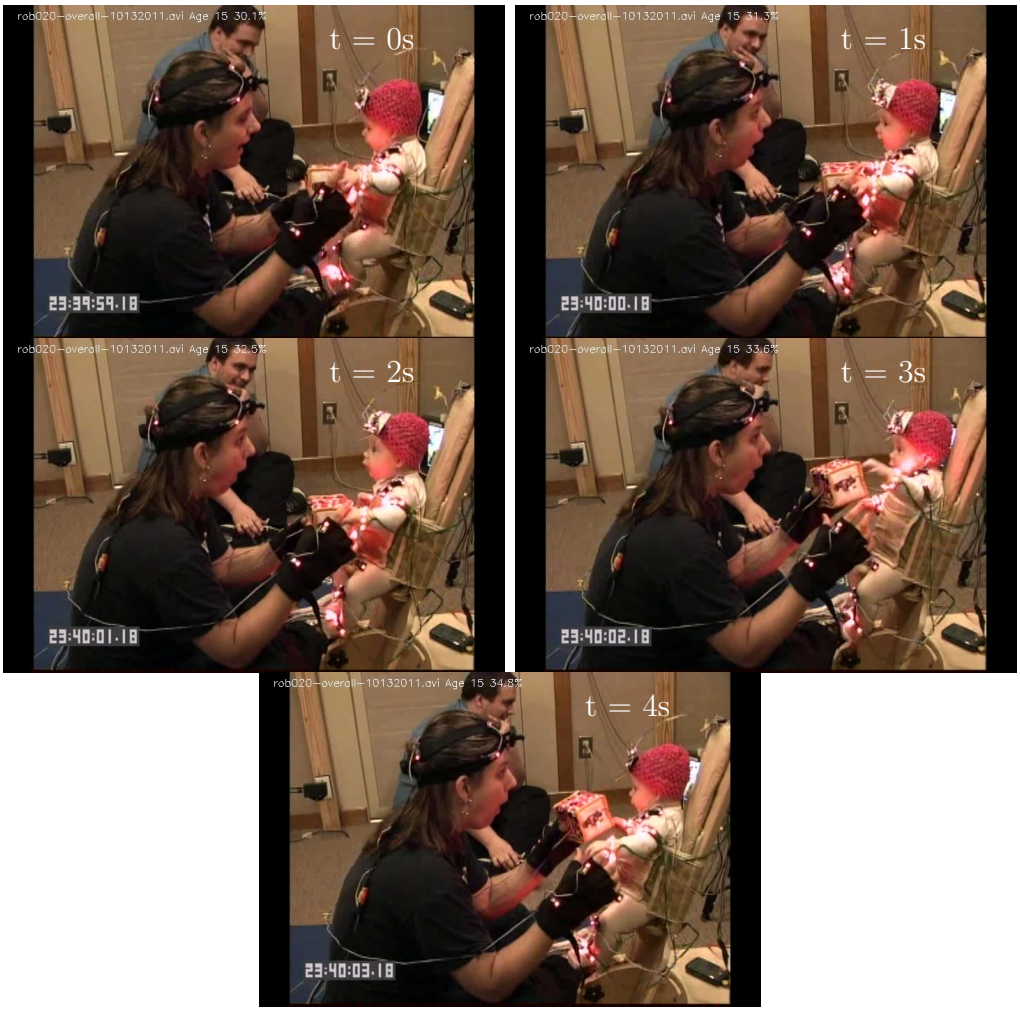

Figure 4.6: Shown are five frames spaced at intervals of 1 second from the portion of the session where the model assigns the lowest probability to the infant intending to track the toy.

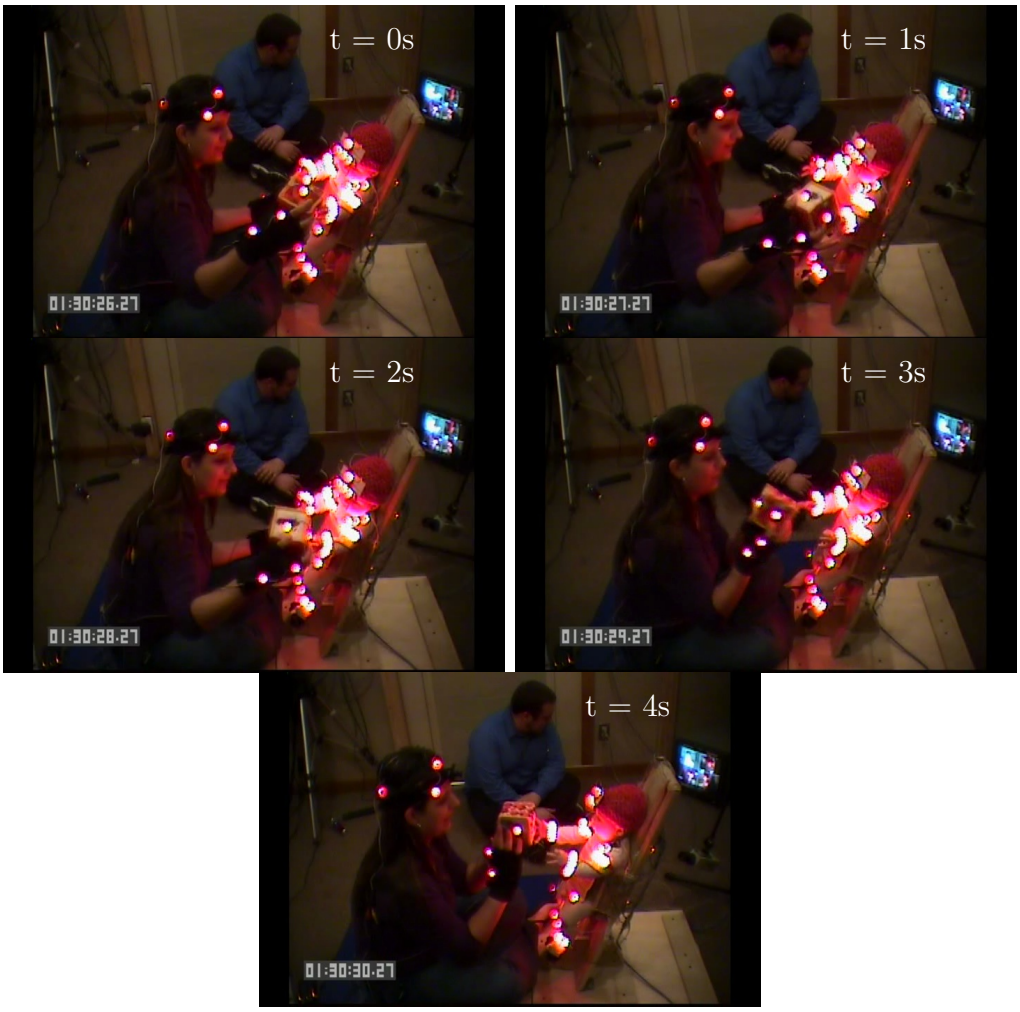

Figure 4.7: Shown are five frames spaced at intervals of 1 second from the portion of the session where the model assigns the highest probability to the infant intending to track the toy.

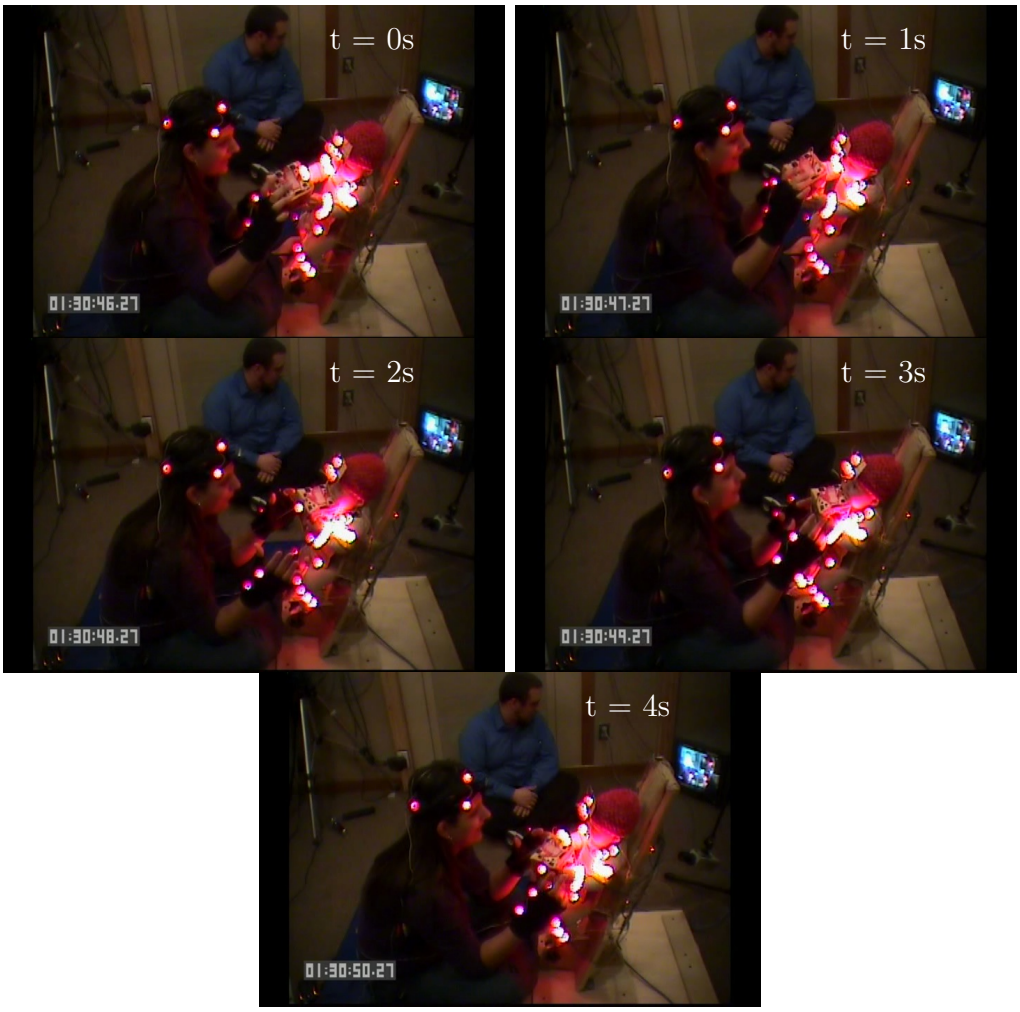

Figure 4.8: Shown are five frames spaced at intervals of 1 second from the portion of the session where the model assigns the lowest probability to the infant intending to track the toy.

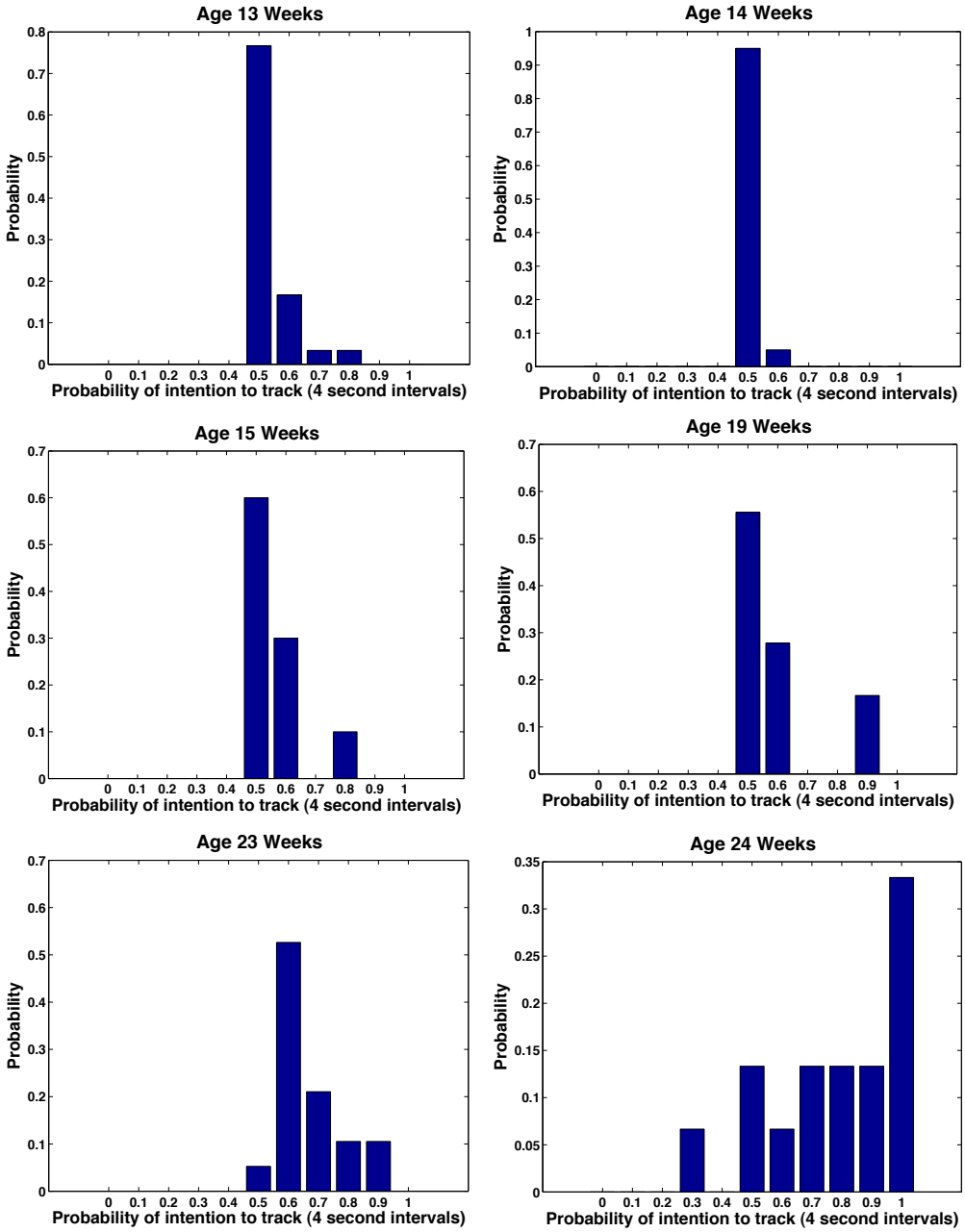

Figure 4.9: Histograms showing the proportion of time that the infant *rob020* intends to track the toy during 6 different motion capture sessions.

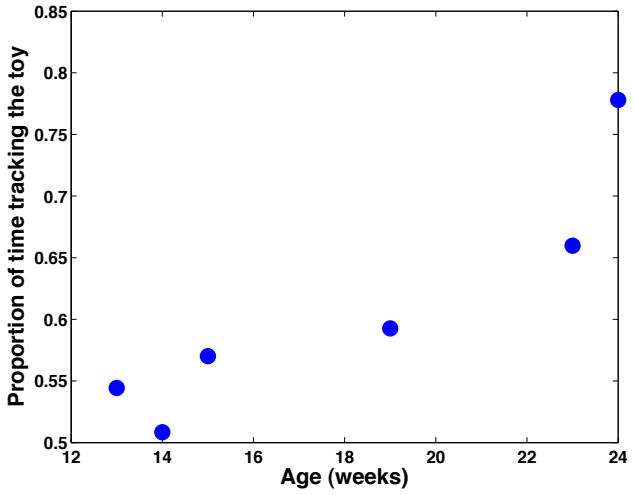

Figure 4.10: The expected proportion of time the infant *rob020* has the intention of tracking the toy over 6 different sessions. The plot shows an increasing proportion of time spent tracking the toy over developmental time.

intention is to track the toy. One might wonder whether we could have specified a heuristic model that would have been just as effective. For instance, a reasonable heuristic would be to look at the mean value of α over some window of time. The intuition being that if α remains low then the infant is probably tracking the toy. However, this heuristic approach will not select segments that correspond to our intuition of tracking. For instance, if the infant were staring straight ahead and the mother was holding the toy in front of the infant without moving it, then this heuristic would designate this is an interval where the infant is tracking the toy (since α would remain near 0 the entire time). However, this scenario does not agree with our intuition about what it means to track a toy. The model presented here is able to answer this question from first-principles by clearly articulating the assumptions behind our inference of the infant's intention. Additionally, there are several potential extensions (see Future Work) that can be easily handled within our framework, but would make designing a sensible heuristic even more difficult.

Applications to Assistive Robotics

In addition to application in the study of infant motor movements, our system can be applied to give robots new perceptual abilities by allowing them to ascribe intentional language to their human interactive partners. A key strength of our inverse optimal control algorithm is its computational performance. Since our technique is based on linear regression, with a sparse design matrix, a 90-second segment of motion capture can be analyzed in about 200 milliseconds. Due to the computational efficiency of this approach, we were able to create a real-time version of our model for inferring the intentions of head movements of a user interacting with a laptop. The approach uses a computer-vision based system called CERT [33] to automatically infer the user's head pose from each video frame recorded using the laptop's built-in webcam. As a proof of concept, we have our system show two agents moving back and forth across the laptop's screen. From the pattern of user head movements, as determined by CERT, we can very efficiently and very quickly hone in on which of the two agents the user is tracking. This software implementation shows that our system is computational efficient and effective even when given the noisy sensory data generated by computer-vision based analysis of human faces. These initial results suggest that our approach may be a promising technique for allowing an assistive robot to determine the intention behind a human's head movements in real-time.

Future Work

While the model presented detects head tracking successfully, there are many ways in which the model could be extended to make more precise inferences about the intentions of infant head movements. Firstly, the assumption that the motion of the toy is solely driven by Brownian motion is not realistic. In our data, the toy is typically held by mother (although sometimes by the infant herself). If mother is holding the toy, the movement of the toy may be far from random, in fact mother may have the intention to help the infant track the toy. If the infant infers that by rotating her head toward the toy, mother is likely to begin to move the toy toward her, then the optimal solution to the tracking problem changes.

This can be accommodated by our model through a modification of the passive dynamics of the control problem.

A shortcoming of the current model is that it assumes that the infant can sense the location of the toy at all points in time. In reality the infant's estimate of the position of the toy will be driven by the direction of the infant's gaze, which in turn provides the infant with noisy sensory information indicating the location of the toy. In this setting we can view the motion of the head as well as of the eyes as important clues to the infant's underlying intention. Reformulating the model presented in this chapter to take this into account would involve using the partially observable inverse optimal control model.

Finally, we will jointly analyze multiple types of infant movement (including limb movements as well as facial expressions). Anecdotally, it appears that in our data the physical and social contexts of motor development are tightly coupled. The result is that behavioral categories that are natural from an adult perspective may be artificial from an infant's perspective. For example, when a caregiver is present, making facial expressions, vocalizing, or moving the legs, may be as effective to make contact with or track an interesting object as simply moving the head or reaching with the arms. Having a principled computational method for cataloguing and understanding developmental milestones in terms of not only the morphology of the infant's behavior, but how different behavioral morphologies may represent new strategies or techniques to accomplish the same intention, may have a significant impact on our understanding of the computational processes driving infant social and motor development.

4.15 Conclusion

We have presented a method for inverse optimal control and imitation in continuous state, action, and time. The strength of our approach lies in the fact that a distribution of the value function of the demonstrator can be computed efficiently and in closed-form. The existence of the exact distribution over value functions allows the handling of many forms of uncertainty not considered in other

approaches. We also show that our system is applicable to many control settings including ones with partial observability, game-theoretic interactions, and discrete state spaces.

We concluded with an experiment that indicates that our approach is well-suited for rapidly uncovering the intentions behind human head movements. Our work with infant motion capture demonstrates that our technique is an effective formalization of Dennett's intentional stance. That is, for continuous systems, we can, in a principled and automatic fashion, ascribe intentions to natural agents. The generality of our formulation allows for both understanding and synthesizing behavior in many different domains.

4.16 Acknowledgment

The text of Chapter 4 is unpublished work to be submitted as two separate manuscripts. The first manuscript will include the research presented in Sections 4.1-4.13 with authors P. Ruvolo and J.R. Movellan. I was the primary author and researcher on the work contained in these sections, providing the mathematical derivations and drafting the manuscript. The second manuscript will include the research in Section 4.14 with authors P. Ruvolo, T. Wu, W. Mattson, D. Messinger, and J.R. Movellan. I was the primary author and researcher on this project, deriving the models, performing the analysis, helping to design the motion capture setup, and drafting the manuscript. Mattson collected the dataset and helped design the motion capture setup used in the experiment. Wu helped with design of the motion capture setup. Messinger and Movellan supervised this research.

Appendix A

A.1 Optimal Action With Entropy Penalty

$$
\begin{aligned}
& \underset{u,u\succeq 0,\|u\|_1=1}{\arg\max} \left\{ \frac{1}{\lambda} H[u] + \gamma \sum_i u_i T_{x,a_i}^\top v \right\} \\
&= \underset{u,u\succeq 0,\|u\|_1=1}{\arg\max} \left\{ \frac{1}{\lambda} H[u] + \gamma \sum_i u_i T_{x,a_i}^\top v - \frac{1}{\lambda} \sum_i u_i \log \sum_j e^{\lambda\gamma T_{x,a_j}^\top v} \right\} \\
&= \underset{u,u\succeq 0,\|u\|_1=1}{\arg\max} \left\{ -\sum_i u_i \log u_i + \sum_i u_i \left(\gamma\lambda T_{x,a_i}^\top v - \log \sum_j e^{\lambda\gamma T_{x,a_j}^\top v} \right) \right\} \\
&= \underset{u,u\succeq 0,\|u\|_1=1}{\arg\max} \left\{ -\sum_i u_i \log u_i + \sum_i u_i \log \lambda_i \right\} \\
&= \underset{u,u\succeq 0,\|u\|_1=1}{\arg\min} \left\{ \sum_i u_i \log \left(\frac{u_i}{\lambda_i} \right) \right\} \\
&= \underset{u,u\succeq 0,\|u\|_1=1}{\arg\min} \ \mathrm{KL}\left(u \| \lambda\right) \\
\lambda_i \quad &= \frac{e^{\gamma\lambda T_{x,a_i}^\top v}}{\sum_j e^{\gamma\lambda T_{x,a_j}^\top v}}
\end{aligned}
$$

Where to get from the first to the second line we subtracted a constant (it is a constant since u is constrained to add up to 1). The KL-divergence is minimized when the two distributions u and λ are equal. Therefore the maximizer of the original expression is $u_i^\star = \frac{e^{\gamma\lambda T_{x,a_i}^\top v}}{\sum_j e^{\gamma\lambda T_{x,a_j}^\top v}}$.

Table A.1: This table includes a list of commonly used symbols (i.e. that occur regularly throughout the thesis. In general, symbols are defined as when they are first referenced, however, this listing is helpful for symbols that are used much later in the text from where they were originally defined.

Symbol	Meaning
n	number of state dimensions
m	number of control dimensions
k	number of Brownian noise dimensions
d	number of basis functions for value the approximator
$\mathcal{X}$	state space for a control problem
$X_t \in \mathbb{R}^n$	system state at time t
$U_t \in \mathbb{R}^m$	control signal at time t
T	terminal time for a finite-horizon control problem
$\mathcal{A}$	action space for both the discrete and control problem
$v_t : \mathcal{X} \times [0, T] \to \mathbb{R}$	value function at time t
$v \in \mathbb{R}^{\lvert\mathcal{X}\rvert}$	discrete-state value function expressed as a vector
$Q \in \mathcal{X} \times \mathcal{A} \to \mathbb{R}$	discrete state-action value function
w	basis weights for the value approximators
w_t	basis weights for the value approximator at time t
$v : \mathcal{X} \to \mathbb{R}$	discrete-state value function
$a : \mathbb{R}^n \to \mathbb{R}^n$	passive-dynamics
$b : \mathbb{R}^n \to \mathbb{R}^{n \times m}$	controlled dynamics gain function
$c : \mathbb{R}^n \to \mathbb{R}^{n \times k}$	Brownian motion gain matrix
$\gamma \in [0, 1)$	discount factor for discrete-time MDP
$\tau \in \mathbb{Z}^+$	discount factor for continuous MDP
$\rho_t : \mathbb{R}^n \to \mathbb{R}$	state-reward rate at time t
$\rho : \mathcal{X} \to \mathbb{R}$	state reward (discrete case)
$\psi_T : \mathbb{R}^n \to \mathbb{R}$	terminal reward function
$q \in \mathbb{R}^{m \times m}$	quadratic control-cost matrix
$r : \mathcal{X} \to \mathbb{R}$	discrete-state reward function
$r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$	continuous MDP reward function

Bibliography

[1] P. Abbeel and A.Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

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

[3] B.D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[4] E. Bates et al. *Language and context: The acquisition of pragmatics*, volume 13. Academic Press New York, 1976.

[5] A.J. Bell and T.J. Sejnowski. The independent components of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.

[6] Bastien Berret, Enrico Chiovetto, Francesco Nori, and Thierry Pozzo. Evidence for composite cost functions in arm movement planning: An inverse optimal control approach. *PLoS Comput Biol*, 7(10):e1002183, 10 2011.

[7] D.P. Bertsekas. Dynamic programming and optimal control, vol. ii. 2007.

[8] D.P. Bertsekas, D.P. Bertsekas, D.P. Bertsekas, and D.P. Bertsekas. Dynamic programming and optimal control. 1995.

[9] C.M. Bishop. *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[10] S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004.

[11] L.M. Brossard and T.G. Dècarie. Comparative reinforcing effect of eight stimulations on the smiling response of infants. *Journal of Child Psychology and Psychiatry*, 9(1):51–59, 1968.

[12] N.J. Butko and J.R. Movellan. Infomax control of eye movements. *Autonomous Mental Development, IEEE Transactions on*, 2(2):91–107, 2010.

[13] R.J. Carroll, C.H. Spiegelman, K.K.G. Lan, K.T. Bailey, and R.D. Abbott. On errors-in-variables for binary regression models. *Biometrika*, 71(1):19–25, 1984.

[14] C. Chevallier, G. Kohls, V Troiani, E.S. Brodkin, and R.T. Schultz. The social motivation theory of autism. *Trends in Cognitive Sciences*, 2012.

[15] Jaedeug Choi and Kee-Eung Kim. Map inference for bayesian inverse reinforcement learning. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1989–1997. 2011.

[16] J.F. Cohn and E.Z. Tronick. Mother-infant face-to-face interaction: The sequence of dyadic states at 3, 6, and 9 months. *Developmental Psychology*, 23(1):68–77, 1987.

[17] D.C. Dennett. *The intentional stance*. The MIT press, 1989.

[18] K. Dvijotham and E. Todorov. Inverse optimal control with linearly-solvable mdps. In *ICML*, volume 10, pages 335–342, 2010.

[19] T. Field. Infants of depressed mothers. *Infant Behavior & Development*, 1995.

[20] T. Flash and N. Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *The journal of Neuroscience*, 5(7):1688–1703, 1985.

[21] A. Fogel. *Developing through relationships: Origins of communication, self, and culture*. University of Chicago Press, 1993.

[22] W.A. Fuller, S.M. Miller, D.J. Schnell, American Statistical Association, and American Statistical Association. Annual Meeting. Measurement error models. 1987.

[23] D.T. Greenwood and RM Rosenberg. Classical dynamics. *Journal of Applied Mechanics*, 44:517, 1977.

[24] M.S. Grewal and A.P. Andrews. Kalman filtering: theory and practice using matlab. 2001.

[25] C.M. Harris and D.M. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394(6695):780–784, 1998.

[26] TS Jaakkola and MI Jordan. Bayesian logistic regression: a variational approach. In *Proceedings of the 1997 Conference on Artificial Intelligence and Statistics, Ft. Lauderdale, FL*, 1997.

[27] M. Kalakrishnan, E. Theodorou, and S. Schaal. Inverse reinforcement learning with pi 2. In *The Snowbird Workshop, submitted to*, 2010.

[28] R.E. Kalman. When is a linear control system optimal? *Journal of Basic Engineering*, 86:51, 1964.

[29] M. Kawato, Y. Maeda, Y. Uno, and R. Suzuki. Trajectory formation of arm movement by cascade neural network model based on minimum torque-change criterion. *Biological Cybernetics*, 62(4):275–288, 1990.

[30] J.Z. Kolter, P. Abbeel, and A.Y. Ng. Hierarchical apprenticeship learning with application to quadruped locomotion. In *Neural information processing systems*, volume 20. Citeseer, 2008.

[31] B.M. Lester, J. Hoffman, and T.B. Brazelton. The rhythmic structure of mother-infant interaction in term and preterm infants. *Child Development*, 56(1):15–27, 1985.

[32] W. Li, E. Todorov, and D. Liu. Inverse optimality design for biological movement systems. In *World Congress*, volume 18, pages 9662–9667, 2011.

[33] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett. The computer expression recognition toolbox (cert). In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 298–305. IEEE, 2011.

[34] D. Marr. *Vision*. New York: WH Freeman & co, 1982.

[35] L. Meirovitch. *Methods of analytical dynamics*. Dover Publications, 1970.

[36] D. Messinger. Smiling. In *Encyclopedia of Infant and Early Childhood Development*, volume 3, pages 186–198. Oxford: Elsevier, 2008.

[37] D.M. Messinger, P. Ruvolo, N.V. Ekas, and A. Fogel. Applying machine learning to infant interaction: The development is in the details. *Neural Networks*, 23(8-9):1004–1016, 2010.

[38] D.S. Messinger, A. Fogel, and K.L. Dickson. What's in a smile? *Developmental Psychology*, 35(3):701, 1999.

[39] J. More. The levenberg-marquardt algorithm: implementation and theory. *Numerical analysis*, pages 105–116, 1978.

[40] J.R. Movellan. Tutorial on continuous time stochastic optimal control, 2012.

[41] C.L. Nehaniv and K. Dautenhahn. *Imitation and social learning in robots, humans and animals: behavioural, social and communicative dimensions*. Cambridge Univ Pr, 2007.

[42] G. Neu and C. Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proc. UAI*, pages 295–302. Citeseer, 2007.

[43] A. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang. Autonomous inverted helicopter flight via reinforcement learning. *Experimental Robotics IX*, pages 363–372, 2006.

[44] A.Y. Ng and S. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 663–670, 2000.

[45] M.J. Osborne and A. Rubinstein. *A course in game theory*. The MIT Press, 1994.

[46] H. Oster. Baby facs: Facial action coding system for infants and young children. *Unpublished monograph and coding manual, New York University*, 2001.

[47] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *Proceedings of the 20th international joint conference on Artifical intelligence*, IJCAI'07, pages 2586–2591, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[48] R. Rienks and D. Heylen. Dominance detection in meetings using easily obtainable features. *Machine Learning for Multimodal Interaction*, pages 76–86, 2006.

[49] S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):537, 2003.

[50] H. Shan, L. Zhang, and G.W. Cottrell. Recursive ica. *Advances in neural information processing systems*, 19:1273, 2007.

[51] A. Simpkins and E. Todorov. Practical numerical methods for stochastic optimal control of biological systems in continuous time and space. In *Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL'09. IEEE Symposium on*, pages 212–218. IEEE, 2009.

[52] E.C. Smith and M.S. Lewicki. Efficient auditory coding. *Nature*, 439(7079):978–982, 2006.

[53] G. Stechler and E. Latz. Some observations on attention and arousal in the human infant. *Journal of the American Academy of Child Psychiatry*, 1966.

[54] R.S. Sutton and A.G. Barto. *Reinforcement learning*, volume 9. MIT Press, 1998.

[55] U. Syed and R.E. Schapire. A game-theoretic approach to apprenticeship learning. *Advances in neural information processing systems*, 20:1449–1456, 2008.

[56] Y. Tassa and E. Todorov. High-order local dynamic programming. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2011 IEEE Symposium on*, pages 70–75. IEEE, 2011.

[57] E. Téglás, E. Vul, V. Girotto, M. Gonzalez, J.B. Tenenbaum, and L.L. Bonatti. Pure reasoning in 12-month-old infants as probabilistic inference. *science*, 332(6033):1054, 2011.

[58] E. Todorov, M.I. Jordan, et al. Optimal feedback control as a theory of motor coordination. *Nature neuroscience*, 5(11):1226–1235, 2002.

[59] E. Todorov and Y. Tassa. Iterative local dynamic programming. In *Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL'09. IEEE Symposium on*, pages 90–95. IEEE, 2009.

[60] E.D. Tronick, H. Als, and T.B. Brazelton. Mutuality in mother-infant interaction. *Journal of Communication*, 27(2):74–79, 1977.

[61] D. Verma and R. Rao. Goal-based imitation as probabilistic inference over graphical models. *Advances in neural information processing systems*, 18:1393, 2006.

[62] N. Wiener. *Cybernetics, or Communication and Control in the Animal and the Machine*. The MIT Press, 1948.

[63] D.M. Wolpert, K. Doya, and M. Kawato. A unifying computational framework for motor control and social interaction. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1431):593–602, 2003.

[64] T. Wu, J. Artigas, M. Mattson, P. Ruvolo, J.R. Movellan, and D. Messinger. Collecting a developmental dataset of reaching behaviors: First steps. *IROS2011 Workshop on Cognitive Neuroscience Robots*, 2011.

[65] M.E. Yale, D.S. Messinger, A.B. Cobo-Lewis, and C.F. Delgado. The temporal coordination of early infant communication. *Developmental Psychology*, 39(5):815, 2003.

[66] B.D. Ziebart, J.A. Bagnell, and A.K. Dey. Modeling interaction via the principle of maximum causal entropy. In *Proc. ICML*, pages 1255–1262, 2010.