THE UNIVERSITY *of* EDINBURGH

Decision shaping and strategy learning in multi-robot interactions

Aris Valtazanos

Doctor of Philosophy

Institute of Perception, Action and Behaviour

School of Informatics

University of Edinburgh

2013

Abstract

Recent developments in robot technology have contributed to the advancement of autonomous behaviours in human-robot systems; for example, in following instructions received from an interacting human partner. Nevertheless, increasingly many systems are moving towards more seamless forms of interaction, where factors such as implicit trust and persuasion between humans and robots are brought to the fore. In this context, the problem of attaining, through suitable computational models and algorithms, more complex *strategic* behaviours that can *influence* human decisions and actions during an interaction, remains largely open. To address this issue, this thesis introduces the problem of *decision shaping* in strategic interactions between humans and robots, where a robot seeks to lead, without however forcing, an interacting human partner to a particular state. Our approach to this problem is based on a combination of statistical modeling and synthesis of demonstrated behaviours, which enables robots to efficiently adapt to novel interacting agents. We primarily focus on interactions between autonomous and teleoperated (i.e. human-controlled) NAO humanoid robots, using the adversarial soccer penalty shooting game as an illustrative example. We begin by describing the various challenges that a robot operating in such complex interactive environments is likely to face. Then, we introduce a procedure through which composable strategy templates can be learned from provided human demonstrations of *interactive* behaviours. We subsequently present our primary contribution to the shaping problem, a Bayesian learning framework that empirically models and predicts the responses of an interacting agent, and computes action strategies that are likely to influence that agent towards a desired goal. We then address the related issue of factors affecting *human decisions* in these interactive strategic environments, such as the availability of perceptual information for the human operator. Finally, we describe an information processing algorithm, based on the Orient motion capture platform, which serves to facilitate direct (as opposed to teleoperation-mediated) strategic interactions between humans and robots. Our experiments introduce and evaluate a wide range of novel autonomous behaviours, where robots are shown to (learn to) influence a variety of interacting agents, ranging from other simple autonomous agents, to robots controlled by experienced human subjects. These results demonstrate the benefits of strategic reasoning in human-robot interaction, and constitute an important step towards realistic, practical applications, where robots are expected to be not just passive agents, but active, influencing participants.

Acknowledgements

I would first of all like to thank my second supervisor, Prof. D.K. Arvind, thanks to whom I was able to do my PhD in the first place and be in a position to write this document now. Four years ago, while I was on the verge of leaving Edinburgh after my undergraduate degree, he persuaded me to stay on and undertake a challenging yet highly interesting project. If I was able to start my PhD at the unusually young age of 21, I undoubtedly owe it to him.

I would also like to thank my first supervisor, Dr. Subramanian Ramamoorthy, for guiding me through this process and for all the very interesting discussions we have had over these four years. I am particularly grateful for his support and encouragement during the first months of my PhD, when I was desperately trying to conceal my lack of experience in robotics (which at the time amounted to a few hours of interaction with Lego Mindstorm kits). My overall PhD experience has definitely been a smooth one, and I credit him for this.

Furthermore, I would like to thank all my friends and colleagues who contributed to making these four years a stimulating and fun experience. I would particularly like to thank my first office mate, Yiannis, who helped me get acclimatised to the office environment in my early PhD days. I would also like to acknowledge my other Greek office mate, Stathis, for the numerous hours we spent together in and out of the lab, and to express my respect to the rest of the office for having to deal with so many Greeks on a daily basis. A special thanks goes to Zee, not only for gracing my papers' supporting videos with her presence, but also for putting up with me, inspiring me, and helping me become a more (ir)rational individual.

Last, and most certainly not least, I feel the need to thank my family and especially my parents for everything they have done for me over the past 25 years. Without any doubt, I would not be anywhere near where I am now, were it not for their continuous support, encouragement, and sacrifices.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Aris Valtazanos*)

Table of Contents

Chapter 1

Introduction

As the physical capabilities of autonomous robots improve, there is also a growing demand for multi-agent applications where robots can interact more seamlessly with other agents. In many such domains, and particularly in those featuring humans or human-controlled agents, robots are currently restricted to a passive role, which typically requires them to follow instructions or execute some pre-specified behaviour. By contrast, ***strategic*** interactions requiring active influence over interacting, potentially adversarial agents, remain a challenging problem.

One reason behind this strategic challenge are the limitations in component technologies, e.g. the perceptual differences between people and robots, or the limited motion capabilities of many robots. Historically, influence and persuasion in human-robot interaction have been primarily associated with how humans perceive robots (e.g. Siegel et al. (2009), Bainbridge et al. (2008)). Furthermore, it is more commonplace to see people give explicit and detailed instructions that are subsequently translated to robot motion (e.g. Nyga and Beetz (2012), Klingspor et al. (1997)). Even this task is incredibly challenging, forcing system designers to push the frontiers of object or human pose recognition or language processing. However, these issues are precursors rather than the primary content of interactive decision making. From this behavioural perspective, one important open question is the related problem of how autonomous robots can *actively* impact human decisions during an interaction.

Increasingly, we are seeing applications where the problem of strategy in interaction is being brought to the fore. For instance, consider the enormously successful automated driving applications. Here, robotics technology has reached a point of maturity where issues such as "failure to anticipate vehicle intent" and "an over-emphasis on lane constraints versus vehicle proximity in motion planning" are becoming prob-

Figure 1.1: The MIT-Cornell collision during the 2007 DARPA Urban Challenge. *Left:* Cornell's robot Skynet. *Right:* MIT's robot Talos. The collision was partly attributed to the failure of the involved vehicles to anticipate each other's intent. Photo: Fletcher et al. (2008).

lems of immediate interest (Fletcher et al. (2008), Flemisch et al. (2003)). Oversight of these issues has been shown to lead to erroneous interactive behaviours, as in Figure 1.1, where the colliding vehicles failed to anticipate and account for each other's motion.

Figure 1.2: A robot guide in a hospital environment. Robots operating in these domains must cooperate effectively with interacting humans and gain their trust, in order to lead them to the desired destination. Photo: AFP (`http://www.theage.com.au/news/technology/the-future-is-here/2006/11/05/1162661536462.html`).

In the domain of personal robotics, the development of robust platforms over the past decade has led to similar trends where the problem of strategically non-trivial interaction becomes very relevant, such as motion synthesis that must account for intentions of humans who co-exist in the work space (e.g. Dragan and Srinivasa (2012), Goodrich and Schultz (2007)). The development of such influencing behaviours would

constitute an important step towards several practical human-robot interaction applications, e.g. robot guides in public spaces (Figure 1.2), where trust and implicit persuasion are relevant issues that need to be incorporated into task specifications. The additional challenge for robots operating in these domains is that they must interact with humans (and not simply other autonomous agents), whose reactions to different situations may vary and are not known exactly a priori. Furthermore, humans can be adaptive and exhibit behaviours that change over time, so characterising their actions with respect to a well-defined library of behaviours may be hard. These constraints highlight the need for algorithms that unify existing approaches from the robotics, decision-making, and statistical modeling communities, in order to address the problem of behavioural influence in multi-robot and human-robot interaction.

1.1 Problem formulation and domain

Motivated by the above interactive decision problems, this thesis focuses on learning mechanisms for strategic influencing behaviours. We study this problem primarily in the setting of interactions between teleoperated (i.e. human-controlled) and autonomous robots. This allows us to emphasise learning of strategies, our core focus, while still operating in a realistic, physical setting, which makes robot learning non-trivial and challenging. This is also a setting which captures an aspect of robotics applications where many different robotic systems and people operate, simultaneously and without explicit coordination, in constrained environments, such as a construction site. The open questions for us would be:

- How robustly could autonomous robots interact with humans if they were not hindered by the current inferiority in physical abilities?
- How can robots effectively make strategic decisions that can influence and affect human behaviour during an interaction, without the benefit of analytically predefined models for the participant?
- How can such strategies be learned from human demonstration, and further synthesised through *repeated interaction* with a given human-controlled robot?

With these considerations in mind, we introduce the problem of ***strategic interaction shaping*** in adversarial mixed robotic environments. A ***mixed robotic environment***

features both autonomous and human-controlled robots, which have identical hardware but differ at the behavioural level. In this context,

Definition 1.1.1 *The **interaction shaping problem**[1] deals with the ability of an autonomous robot to affect the state of an adversarial agent in a strategic interactive task.*

The conceptual structure of the interaction shaping problem is illustrated in Figure 1.3.

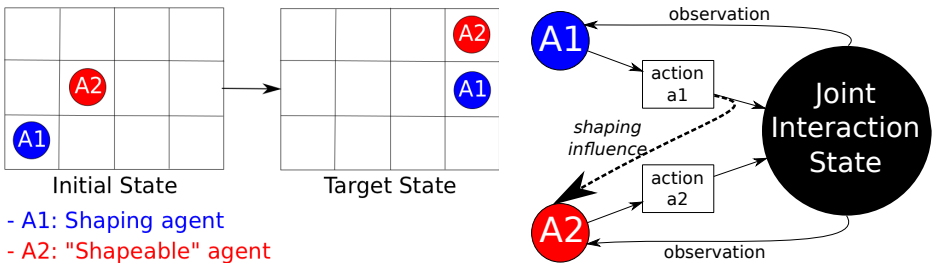


Figure 1.3: Conceptual structure of the interaction shaping problem. Agent A1 seeks to lead, *without directly forcing*, an *adversarial* agent A2 to a new *joint* target state. A1 must achieve this objective by *indirectly influencing*, through its own chosen actions, the actions selected by A2.

1.1.1 Problem assumptions and constraints

By considering primarily ***adversarial*** interactions, we seek to model situations where the interacting robot may be actively countering or refusing to cooperate with a shaping strategy. Furthermore, we consider interactions that are only ***partially controllable***, in that the autonomous robot cannot directly force the adversary to the desired joint state. Thus, the robot must shape the interaction *indirectly*, by identifying actions that can cause the adversary to behave in accordance with its own goal. Moreover, a robust shaping robot must learn to influence a given adversary from its own experience, without the provision of additional information on the characteristics (e.g. human-controlled vs. autonomous) of that agent. This adversarial setup presents different challenges to cooperative multi-robot settings, where robots are working towards a common objective.

[1] In various parts of this thesis, we use the term "*decision* shaping" instead of "*interaction* shaping", in order to avoid repetition of the word *interaction* and its derivatives. However, the two terms should be treated as equivalent.

1.2 Summary of main contribution

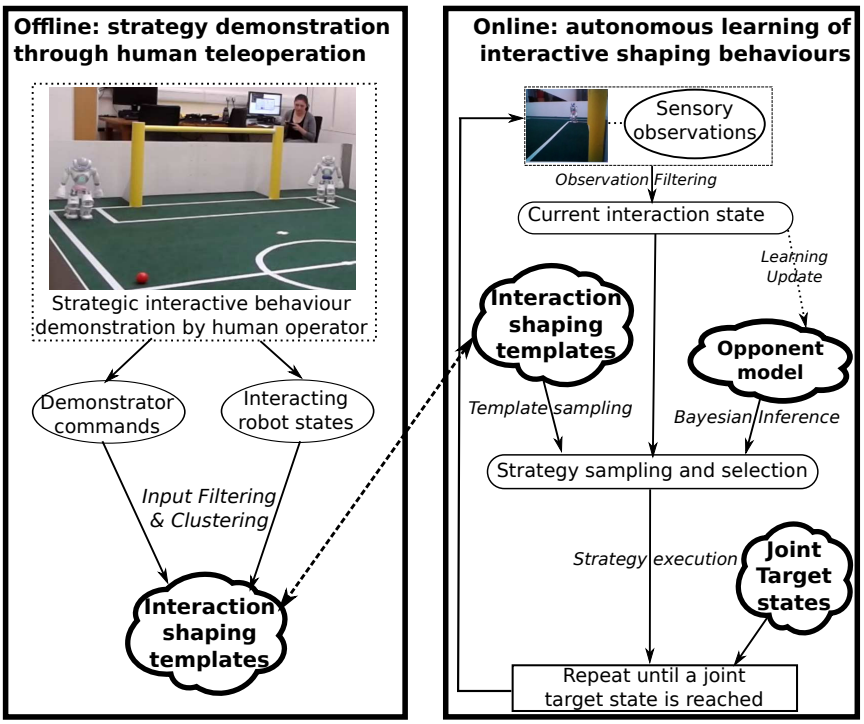


Figure 1.4: Conceptual structure of proposed approach to interaction shaping in multi-robot environments. In the offline learning phase, a human operator remote-controls a robot in a strategic interaction with another robot, which could also be teleoperated or be a heuristic autonomous adversary. The operator provides several traces of the desired behaviour, which are filtered and clustered into interaction templates, represented as state space *regions* and action space *tactics*. In the online phase, the learned templates form the basis of an autonomous shaping agent, who can strategically influence unknown adversarial robots. The shaping agent additionally maintains a model of its opponent's behaviour, which is empirically updated through *repeated interaction*, and a set of *joint* target states, representing desired terminal configurations for the two robots. By sampling and selecting, through Bayesian inference, different interaction strategies, the shaping agent progressively learns sequences of actions that can lead, with high probability, the adversary to the desired target states.

The primary contribution of this thesis is a learning framework for strategic interaction shaping in physical mixed robotic environments. The proposed approach is divided into two interrelated learning phases (Figure 1.4). In the *offline phase*, human demonstrators provide examples of interactive strategic behaviours, which are clustered into composable interaction templates, represented as state space *regions* and

action space *tactics*. In the *online* phase, the learned templates form the basis for a probabilistic Bayesian inference and sampling algorithm, which is used to learn behaviours that can influence a given strategic adversary. Thus, an agent learns interaction *strategies* – represented as sequences of actions – that are likely to influence the given adversary and lead the interaction to a desired state.

The proposed framework is designed to meet the needs of complex physical interactive environments. Here, robots must repeatedly make decisions from limited sensory information (e.g. noisy images coming from a perspective camera, as in the top-right part of Figure 1.4), in order to influence strategic adversaries with unknown behavioural profiles. Thus, our approach addresses different challenges than traditional established decision-making models, for example, Interactive Partially Observable Markov Decision Processes (Gmytrasiewicz and Doshi, 2005), which focus on offline optimisation over states and actions. By contrast, this thesis is concerned with approximate solutions to iterative reasoning problems, which are empirically obtained through *repeated interaction* with a given, a priori unknown, strategic adversary.

To the best of our knowledge, our work is the first to introduce a learning model for interactive, adversarial robotic environments with human-controlled agents. Moreover, our experiments innovate in demonstrating strategic autonomous behaviours that can influence interactive human decisions towards a desired outcome. Thus, we introduce a new modeling paradigm for robotic systems, unifying ideas and techniques from different heterogeneous domains, which leads to results with significant implications for many mixed-initiative domains.

1.3 Experimental domain

In order to demonstrate our approach, we require an experimental setup where the learning agent can interact with a wide range of strategic adversaries, including human-controlled opponents. To this end, our primary experimental domain is the robotic soccer ***penalty shooting*** problem between NAO humanoid robots (Figure 1.5), which can be either autonomous or human-controlled. In this game, a *striker* is tasked with scoring a goal against an adversarial *goalkeeper*, within a specified amount of time. The problem is inspired by the RoboCup Standard Platform League (`http://www.tzi.de/spl/`), an international robotic soccer competition featuring several

(a) (b)

Figure 1.5: Experimental setup. (a): The NAO humanoid robot. (b): The soccer field, with an orange ball on the penalty cross mark. The initial poses of the striker (near side, blue waistband) and the goalkeeper (far side, pink waistband) are also shown.

world-leading research groups in artificial intelligence and robotics[2]. A defining characteristic of this experimental setup is that the two robots are identical and thus have the same locomotion capabilities. Thus, even when one of the robots is controlled by a human subject, that subject cannot benefit by e.g. walking faster or kicking the ball harder (as would happen in a direct human-robot penalty shooting contest). Thus, we can compare and contrast autonomous and human-controlled robots directly at the *behavioural level*, without however removing the underlying physical uncertainty that is present in any realistic human-robot environment.

Penalty shooting is a challenging problem in interactive decision making because it is an adversarial interaction between competing agents, where decisions must be made repeatedly in a limited period of time[3]. Thus, autonomous robots must not only make robust decisions in the presence of other, potentially human-controlled agents, but they must also *outperform* them at the end of each trial. The interactive nature of the task indicates that simple strategies (e.g. pick a side of the goal at random, align and kick) are likely to be unsuccessful, as the adversary may be able to recognise and defend them more easily. Instead, policies that incorporate human behavioural traits, such as *deceiving* the goalkeeper into moving towards the other side of the goal, are more likely

[2]The author of this thesis has also been an active participant in the Standard Platform League. In 2012, the author's team, *Edinferno*, reached the quarter-finals of this competition (out of 28 teams). The author was a key contributor to this effort, having developed most of the team's technical software and behaviour algorithms.

[3]In our version of the penalty shooting game, as later discussed in Section 4.2.5, the striker is only allowed to kick the ball once (no dribbling allowed). Thus, a key part of the interaction is the interval preceding the kick, where the players observe each other's moves and plan their actions accordingly.

to succeed. A robust autonomous agent must also be able to interact with a wide range of adversaries, whose behaviour and responses may vary. Thus, penalty shooting is an excellent fit to the interaction shaping problem considered in this thesis, supporting the evaluation of different decision-making algorithms in a physical experimental setup.

In the experiments presented in the remainder of this thesis, we look at several different versions of the penalty shooting problem (e.g. autonomous learning striker vs. human-controlled goalkeeper, human-controlled striker vs. simple fixed-heuristic autonomous goalkeeper, etc.). Our results are based on experiments with a wide range of human subjects, and are aimed at testing our algorithms under a diverse set of conditions and possible opponent strategies.

1.4 Thesis outline

1.4.1 Overview

In this section, we provide a brief summary of the contents of the remaining chapters of the thesis. We highlight the novel contributions presented in each chapter, as well as their connections to the overall interaction shaping problem. For each chapter, we also indicate the type of robotic environment on which analysis and evaluation is based.

To summarise, we first review selected literature from related domains (Chapter 2), and we then describe the problem of decision making in physically constrained multi-robot environments, highlighting the challenges presented by the sensing and the strategic uncertainty in these domains (Chapter 3). Then, we introduce and evaluate an algorithm for learning interactive strategic behaviours from human demonstrations (Chapter 4). We subsequently present and experimentally evaluate our main contribution, an empirical learning framework for interaction shaping in mixed environments (Chapter 5). This evaluation is followed by a further user study, which assesses the effects of perceptual constraints on the decisions humans make when interacting strategically with robots (Chapter 6). Motivated by these results, we subsequently describe a motion tracking algorithm towards the realisation of direct – as opposed to teleoperation-mediated – strategic human-robot interactions (Chapter 7). Finally, we discuss possible future extensions to our work (Chapter 8).

1.4.2 Chapter 2: Related work

In Chapter 2, we review related work (primarily) from the robotics and autonomous decision making literature. We discuss how our approach extends and combines existing ideas to address a novel class of robotic interaction problems.

1.4.2.1 Summary of chapter contributions

- A comprehensive survey of research methods related to interaction shaping.
- A critical evaluation of related work in the robotics and decision making literature, and a situation of the proposed approach with respect to these studies.

1.4.3 Chapter 3: Sensing and strategic uncertainty in multi-robot interactions

Chapter 3 introduces the problem of robust strategic decision making in adversarial interactions with physical limitations in action and perception. We first describe an inference mechanism for predicting and filtering the state of an interacting agent. Then, we propose an approach to devising strategic interactive behaviours for autonomous robots, illustrated through the robotic soccer domain. Our method constitutes a principled approach towards addressing the *sensory* and *strategic* uncertainty arising in these environments. We demonstrate how the exploitation of these constraints can lead to interesting forms of motion strategies against a variety of adversaries.

- **Interaction domain:** interactions between multiple simulated autonomous robots.

1.4.3.1 Summary of chapter contributions

- Formulation of the *Reachable Set Particle Filter*, a novel adversary state estimation algorithm combining data-driven approximation with dynamical constraints.
- A learning framework for interactive decision in multi-robot adversarial interactions, based on a combination of probabilistic and game-theoretic tools.

1.4.3.2 Related publications

- A. Valtazanos and S. Ramamoorthy, Intent inference and strategic escape in multi-robot games with physical limitations and uncertainty. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

1.4.4 Chapter 4: Learning to interact with strategic agents from human demonstrations

In Chapter 4, we bring humans in the interaction loop as controllers of physical robots. We address the problem of learning, from human demonstration, strategies that can indirectly influence adversaries by appropriately chosen actions. We present an algorithm that learns strategy templates from human demonstrations, and synthesises them based on the observed behaviour of an interacting robot. The resulting algorithm lays the foundations for our interaction shaping framework, by producing a set of reusable interaction templates that can be adapted to a wide range of strategic adversaries.

- **Interaction domain:** multi-robot interactions between NAO humanoid robots.
  - Offline phase: human-controlled robot vs. heuristic autonomous robot.
  - Online phase: autonomous robot programmed by demonstration vs. autonomous and human-controlled robots.

1.4.4.1 Summary of chapter contributions

- A learning procedure for clustering demonstrated behaviours into composable *interaction templates*, combining elements of learning by demonstration and opponent modeling.
- An algorithm for hierarchical synthesis of the learned templates, which is based on a *dynamically weighted Gaussian Mixture Model*.

1.4.4.2 Related publications

- A. Valtazanos and S. Ramamoorthy, Decision shaping and strategy learning in multi-robot interactions, *Journal article in preparation*.

1.4.5 Chapter 5: Learning to shape and influence strategic interactions

In Chapter 5, we present our main contribution to the interaction shaping problem. The proposed framework builds on a modification of the procedure of Chapter 4, which provides templates for *shaping behaviours*. These templates form the basis of an adaptive

learning algorithm, which progressively updates distributions on the adversary's responses to actions and on the reachability of different state space regions. Through a combination of Bayesian inference and iterated prediction of the adversary's actions, the algorithm learns strategies that are likely to ***shape*** the interaction and attain a desirable target state (as in Figure 1.3).

- **Interaction domain:** multi-robot interactions between NAO humanoid robots.
  - Offline phase: human-controlled robot vs. heuristic autonomous robot.
  - Online phase: ***learning*** autonomous robot programmed by demonstration vs. human-controlled and autonomous robots.

1.4.5.1 Summary of chapter contributions

- A principled method for using human demonstrations in interactive *learning* of strategic decisions.
- A procedure for sampling and selecting demonstrated actions, with the intention of reaching one of several possible target interaction states. This procedure yields temporally extended strategies with which a given adversary is expected to comply, thus maximising the expectation of reaching a target state.
- A Bayesian framework for strategic interaction shaping in mixed robotic environments, through which an autonomous agent can learn to interact with a wide range of adversaries.

1.4.5.2 Related publications

- A. Valtazanos and S. Ramamoorthy, Bayesian interaction shaping: learning to influence strategic interactions in mixed robotic domains, *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2013.
- A. Valtazanos and S. Ramamoorthy, Decision shaping and strategy learning in multi-robot interactions, *Journal article in preparation*.

1.4.6 Chapter 6: Perceptual constraints in interactive teleoperation

Chapter 6 assesses the ability of humans to respond to strategic behaviours by autonomous robots, and the factors impacting their performance in related mixed robotic environments. We report on an experimental study assessing the effects of *limited perception* on human decision making. We assess user performance on a cooperative and an adversarial task, each requiring different forms of interaction with an autonomous robot. In each case, the perceptual information available to users is progressively restricted, from full visibility of the interaction environment, to having access only to the teleoperated robot's noisy camera feed.

- **Interaction domain:** multi-robot interactions between teleoperated and autonomous NAO humanoid robots.

1.4.6.1 Summary of chapter contributions

- An experimental study examining the correlation between the *strategic difficulty* of a teleoperation task, the *perceptual information* available to human operators, and their ability to *infer the intent* of the interacting autonomous robots.
- Illustration of specific examples where subjects exhibit considerably different behaviours when perception is restricted, and scenarios where autonomous robots are more likely to influence the outcome of an interaction.

1.4.6.2 Related publications

- A. Valtazanos and S. Ramamoorthy, Evaluating the effects of limited perception on interactive decisions in mixed robotic domains, *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2013.

1.4.7 Chapter 7: Towards direct strategic human-robot interactions

The methods and experiments described in the previous chapters correspond to interactions where humans are *indirect* participants as robot operators. Chapter 7 presents a motion tracking algorithm which can be used for *direct* physical human-robot interaction. The proposed technique addresses the problem of continuous simultaneous tracking of human *posture* and *position*, in unconstrained interaction environments

where traditional motion capture systems are not directly applicable. Our method is based on a combination of *optical* and *inertial sensing* motion capture technologies, which are jointly used to learn a predictive model of human motion from demonstrated data.

- **Interaction domain:** human motion capture, with application to ***physical*** strategic human-robot interaction.

1.4.7.1 Summary of chapter contributions

- A novel motion tracking algorithm combining the relative strengths of optical (Microsoft Kinect) and inertial sensing (Speckled Computing) systems, with application to complex physical human-robot interactions.
- A novel application of wearable inertial sensor systems to position inference in unconstrained motion capture environments.

1.4.7.2 Related publications

- A. Valtazanos, D.K. Arvind, S. Ramamoorthy, Using wearable inertial sensors for posture and position tracking in unconstrained environments through learned translation manifolds, *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2013.

1.4.8 Chapter 8: Conclusions and future work

Chapter 8 reviews the main contributions of the thesis to interactive decision making in strategic human-robot interactions. This is followed by a discussion on potential extensions to challenging robotic applications, for example, field robotics systems requiring robust interaction interfaces between human operators and deployed robots.

1.5 List of all contributed publications

This thesis has led to the publication of the following peer-reviewed international journal and conference articles:

- A. Valtazanos, D.K. Arvind, and S. Ramamoorthy. Using wearable inertial sensors for posture and position tracking in unconstrained environments through

learned translation manifolds. *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2013.

- A. Valtazanos and S. Ramamoorthy. Bayesian interaction shaping: learning to influence strategic interactions in mixed robotic domains. *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2013.

- A. Valtazanos and S. Ramamoorthy, Evaluating the effects of limited perception on interactive decisions in mixed robotic domains, *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2013.

- A. Valtazanos, D.K. Arvind, and S. Ramamoorthy. Latent space segmentation for mobile gait analysis. *ACM Transactions on Embedded Computing Systems* 12(4), 2013.

- A. Valtazanos and S. Ramamoorthy. Intent inference and strategic escape in multi-robot games with physical limitations and uncertainty. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

- A. Valtazanos and S. Ramamoorthy. Online motion planning for multi-robot interaction using composable reachable sets. *RoboCup International Symposium – Springer Verlag Lecture Notes in Artificial Intelligence*, 2011.

- A. Valtazanos and S. Ramamoorthy. NaOISIS: A 3-D behavioural simulator for the NAO humanoid robot. *RoboCup International Symposium – Springer Verlag Lecture Notes in Artificial Intelligence*, 2011.

- A. Valtazanos, D.K. Arvind, and S. Ramamoorthy. Comparative study of segmentation of periodic motion data for mobile gait analysis. *ACM International Conference on Wireless Health*, 2010.

The thesis author is the primary author of all the above works, having developed the described theoretical models and conducted the associated experimental evaluation.

Chapter 2

Related work

This chapter reviews related literature and background material to the methods presented in this thesis. Section 2.1 reviews models of autonomous decision making based on Markov Decision Processes. In Section 2.2, we present an overview of related ideas from the opponent modeling and behavioural influence literature, whereas in Section 2.3 we consider other works where shaping has been studied in contexts different to our own. Section 2.4 considers the use of human demonstrations in interaction strategy learning, while Section 2.5 reviews studies on interactive decision making in human-robot domains (which is the focus of our user study in Chapter 6). In Section 2.6, we present robotic soccer domains which are similar to our own. Section 2.7 describes techniques from the hybrid systems and particle filtering, which inspire the inference algorithm presented in Chapter 3. Furthermore, Section 2.8 reviews related motion capture platforms and algorithms, which form the basis of our approach to unconstrained human motion tracking in Chapter 7. Finally, Section 2.9 summarises the key findings of this literature review, and reiterates the structure of the remaining chapters of this thesis.

2.1 Models of autonomous decision making

2.1.1 Partially Observable Markov Decision Processes

Partially Observable Markov Decision Processes (POMDPs) (Kaelbling et al., 1998) have been the standard model for decision problems in partially observable domains. A POMDP is defined as a tuple $\langle S, A, T, \Omega, O, R \rangle$, where:

- S is a set of states

- A is a set of actions
- $T : S \times A \times S \mapsto [0,1]$ is the transition function returning the likelihood of a new state following the execution of an action
- Ω is a set of observations the agent can make on the environment
- $O : S \times \Omega \times A \mapsto [0,1]$ is a function returning the likelihood of an observation being made at a state following the execution of an action
- $R : S \times A \times S \mapsto \Re$ is the reward function returning the expected payoff for the agent following the execution of an action.

In POMDPs, optimal policies are computed through optimisation over the spaces and functions defined above. This optimisation is typically conducted offline, by iteratively computing the expected utility of different policies based on the dynamics of the modeled system. Due to their general formulation, POMDPs have been applied in a wide range of decision-making problems (see Cassandra (1998) for a comprehensive survey), with several of those applications being related to robot planning e.g. (Pineau and Gordon, 2005; Hsiao et al., 2007; Ong et al., 2009). POMDPs have also been used in human-robot interaction to model interaction with human agents, e.g. (Broz et al., 2008; Rosenthal and Veloso, 2011; Taha et al., 2011).

Despite their representational power, POMDPs do not directly account for adversarial or other interacting agents in multi-robot domains. In these environments, a robot must cope with two types of uncertainty: the ***sensing*** uncertainty which is due to the robot's noisy sensory readings, and the ***strategic*** uncertainty, which represents the unknown effects of the actions of an interacting robot; in Chapter 3, we discuss these constraints in more detail. With respect to the observation function Ω defined above, these two types of uncertainty are indistinguishable. Thus, explicit modeling of and reasoning over the behaviour of interacting agents becomes a challenging task.

To alleviate this constraint, Interactive POMDPs (IPOMDPs) (Gmytrasiewicz and Doshi, 2005) extend POMDPs by introducing an *interactive* definition of the state space. Interactive states are a product of world states (as defined in traditional POMDPs) and possible *models* of the other interacting agents' policies. In particular, an IPOMDP for agent i interacting with agent j is a tuple $\langle IS_i, A, T_i, \Omega_i, O_i, R_i \rangle$, where:

- $IS_i = S \times M_j$ is a set of interactive states, where S is defined as above, and M_j is the set of possible models for j. A model $m_j \in M_j$ is a tuple $\langle f_j, h_j, O_j \rangle$,

where h_j is a history of observations on j, f_j is a mapping from a history to a distribution $\Delta(A_j)$ over j's actions, and O_j is a function specifying how the environment supplies inputs to the agent.

- $A = A_i \times A_j$ is a set representing the joint actions of both agents
- $T_i : S \times A \times S \mapsto [0,1]$ is the transition function for agent i
- Ω_i is a set of observations on the environment
- $O : S \times \Omega_i \times A \mapsto [0,1]$ is the observation function
- $R_i : IS_i \times A \times IS_i \mapsto \Re$ is the reward function.

Thus, the IPOMDP representation introduces a clearer distinction between world states and agent models. IPOMDPs have been primarily considered in adversarial problems such as money laundering (Ng et al., 2010) and behaviour learning from human teachers (Woodward and Wood, 2012). However, to the best of our knowledge, IPOMDPs have not been used in physical robotic environments, due to the combined complexity of the state and model spaces.

An IPOMDP can be further extended to incorporate different *nested levels of reasoning* in interactive decision-making. These levels represent the depth of reasoning executed by agent i on the behaviour of agent j. At the base (0-th) level, the beliefs of i are simply probability distributions over the state space, S. At the next (first) level, beliefs are augmented to additionally include the 0-level models of agent j. This process can be recursively extended up to an arbitrary level l, such that the beliefs of i at level l incorporate all models of j up to level $l-1$. Thus, agent i can model agent j as a decision maker who also has a bounded depth of reasoning. The incorporation of reasoning levels leads to the *finitely nested* I-POMDP, whose modified tuple is $\langle IS_{i,l}, A, T_i, \Omega_i, O_i, R_i \rangle$, where l is the reasoning level, and the remaining components are defined as in the original IPOMDP formulation.

The transition model of the IPOMDP formulation does not differ fundamentally from the corresponding POMDP one. However, this similarity is enforced by the following assumption:

Definition 2.1.1 ***Model Non-manipulability Assumption (MNM)*** *(Gmytrasiewicz and Doshi, 2005): Agents' actions do not change the other agents' models directly.*

The interaction shaping problem considered in this thesis similarly assumes that the states and actions of interacting agents cannot be directly manipulated. Instead, our problem is primarily concerned with *indirect* influence over the actions of a strategic adversary. Thus, we are interested in learning the transition dynamics of an interactive system, in order to compute policies with which the adversary is likely to comply.

Both POMDPs and IPOMDPs are known to be intractable in problems with large state, action, and/or opponent model spaces (Papadimitriou and Tsitsiklis, 1987). Thus, several approximation algorithms have been proposed instead, such as point-based methods and sampling-based optimisation techniques (Thrun, 2000; Lusena et al., 2001; Pineau et al., 2003; Porta et al., 2006; Kurniawati et al., 2008, 2011; Doshi and Gmytrasiewicz, 2009). These algorithms operate by sampling points from the relevant belief spaces, in order to approximate optimal policies in uncertain environments.

A related variant of POMDPs are Decentralised POMDPs (DEC-POMDPs) (Bernstein et al., 2000). DEC-POMDPs consider models where transitions and reward functions are defined in terms of the states and actions of multiple agents, which jointly operate in a shared environment.

2.1.1.1 POMDPs and interaction shaping

The interaction shaping problem deals with similar representational and computational issues as POMDPs and IPOMDPs, arising from the need to find optimal actions against an unknown adversary. In Chapter 5, we present a Bayesian framework for interactive learning of shaping behaviours, which is similarly based on inference over large and potentially uncertain state, action, and opponent model spaces. However, a major constraint in our problem is that the interacting adversaries are primarily *human-controlled*, exhibiting behaviours that may change dynamically during an interaction. This constraint introduces the need for *online*, empirical learning mechanisms, which can adapt to perceived variations in the interacting adversary's responses. Thus, *offline* optimisation, through point-based or other approximation methods, does not fully address the needs of our domain when used in isolation. Similarly, decentralised processes like DEC-POMDPs typically assume commonly aligned rewards between interacting agents, so they are also incompatible with our setup, where agents are not aware of each other's reward processes.

Our proposed approach is based on a combination of offline sampling and online learning. In the offline sampling phase (Chapters 4, 5), we collect traces of human demonstrations of the desired strategic interactive behaviours. These traces are used to

learn the salient modes of the state and action spaces, thus addressing the complexity of planning over large spaces analogously to point-based methods. In the online learning phase (Chapter 5), the robot empirically collects samples of the adversary's responses to executed actions. These responses are used to *iteratively predict* the evolution of the interaction, following the execution of a temporally extended sequence of actions. This prediction models the compliance of the adversary with the given sequence, thus recreating the nested reasoning effect of IPOMDPs. Through this formulation, we address the dual challenge of interaction shaping in physically grounded systems, where autonomous robots must not only learn to influence a non-cooperative agent, but also to do so interactively from empirically sampled sensory observations.

2.1.2 Temporally extended action planning

Semi-Markov Decision Processes (SMDPs) (Howard, 1971; Bradtke and Duff, 1994; Mahadevan et al., 1997) define actions that take variable amounts of time and can be extended over a specified time horizon. This effect is illustrated in the formulation of Markov *options* (Sutton et al., 1998, 1999), which are generalisations of action selection policies with input and termination conditions. An option is defined as a tuple $\langle I, \pi, \beta \rangle$, where

- $I \subseteq S$ is the set of *input* states
- $\pi : S \times A \mapsto [0,1]$ is the *policy*
- $\beta : S \mapsto [0,1]$ is the *termination condition*

An option $o = \langle I, \pi, \beta \rangle$ can be invoked from a state s if and only if $s \in I$. If o is selected, actions are selected according to π, until the (probabilistic) termination condition β is met. This formulation allows for the creation of localised action policies, which can be invoked from specific regions of the state space, and which can be followed for a variable period of time.

In robotic soccer, a notable application of options has been the keepaway scenario (Stone et al., 2005), where a team of robots tries to pass a ball around while avoiding interceptions from its opponents. The formulation introduced in that paper was analogous to a distributed SMDP with no shared knowledge between teammates, where each player is responsible for a fraction of the overall team decision process. Options were used to encode various macro-actions relevant to the game, such as ball holding,

passing, and blocking a pass. Optimisation over these micro-actions was conducted through the popular reinforcement learning SARSA(λ) algorithm (Sutton and Barto, 1998).

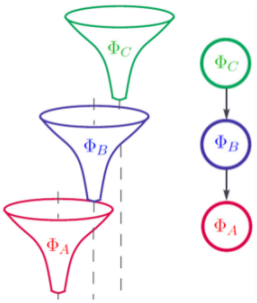


Figure 2.1: The sequential controller switching idea introduced in (Burridge et al., 1999). Controllers are represented as funnels with initial and termination conditions, which can be intermittently activated and sequenced in a global policy. Photo: Havoutis (2012).

A related problem in the robot control community is the composition of actions into a global policy that leads to an overall goal. A classic example is the sequential composition of local controllers, represented as *funnels* with initial conditions and goal states, which can be chained and activated intermittently (Burridge et al., 1999) (Figure 2.1). This method is extended to motion planning (Conner et al., 2006; Tedrake, 2009), where funnels represent workspace constraints or more formally defined linear quadratic regulators.

Belta et al. (2007) present a generalisation of this sequential paradigm towards more general symbolic motion planning. They introduce a hierarchical abstraction that takes as input a high-level system specification and outputs a controller implementation for an autonomous robot. To achieve this objective, the provided specification is partitioned into predicate regions, which represent the possible discrete solutions to the problem. A graph search over these regions is then performed to select an appropriate execution sequence, subject to specified sensing, mechanical, and other robot-related constraints. At the implementation level, the execution is converted into a hybrid automaton specifying the overall control strategy. The approach also allows for hierarchical synthesis of multiple controllers developed using the above method.

2.1.2.1 Connection to interaction shaping

In Chapters 4 and 5, we present decision-making algorithms which are motivated by the above concepts. Our approach is based on the decomposition of the state and action spaces into *regions* and *tactics*, respectively, based on provided human demonstration examples. Tactics are similar to options, in being temporally extended actions with specific preconditions and execution-time constraints. However, the input and target states of shaping tactics are interactive, so they account for both agents (in contrast the distributed approach followed by Stone et al. (2005), which does not feature explicit opponent models). This extends traditional SMDP formulations where there is no explicit reasoning about the adversary. Moreover, the synthesis of strategies as tactic sequences bears a similarity to funnel controllers in an interactive setting. In our approach, tactics are chained together through empirical distributions measuring the reachability of shaping regions. Furthermore, similarly to Belta et al. (2007), we also begin with instances of the solution space (which in our case take the form of interactive human demonstrations), over which we perform search, synthesis, and inference, in order to generate the desired autonomous behaviour. Thus, this formulation leads to policies that are expected to maximise the probability of attaining a desired interactive target state within a specified time horizon.

2.2 Opponent modeling and behavioural influence

2.2.1 Intent inference and plan recognition

Throughout this thesis, we consider the problem of modeling the intent of other strategic agents. In robotics, *intent inference* (Demiris, 2007; Valtazanos and Ramamoorthy, 2011a; Wang et al., 2012) is the process of determining the underlying structure of the behaviour of an interacting robot, from limited sensory information. *Plan recognition* is a related concept concerned with the classification of an agent's actions into a predefined library of plans; see (Carberry, 2001) for a comprehensive review of related literature in this domain. Techniques in this domain include fast symbolic methods for processing multi-featured observations, e.g. (Avrahami-Zilberbrand and Kaminka, 2005), vision-based tracking methods, e.g. (Bobick and Davis, 2001; Messing et al., 2009), and probabilistic algorithms, e.g. (Charniak and Goldman, 1993; Bui, 2003; Geib and Harp, 2004; Baker et al., 2009; Geib and Goldman, 2009).

2.2.1.1 Connection to interaction shaping

The opponent modeling techniques presented in this thesis are inspired by the above symbolic and probabilistic methods. However, our primary focus is on techniques that can simultaneously account for both the strategic and the sensing uncertainty in an adversarial domain (Chapter 3), and exploit the acquired observations in order to influence an interacting agent (Chapter 5).

2.2.2 Regret minimisation and the bandit problem

Game-theoretic techniques have also been used in adversarial decision-making problems. Regret minimisation is a general method used to determine the utility of actions against adversaries with unknown strategies (see Nisan et al. (2007) for an overview of the problem). A related application of regret minimisation is the bandit problem, originally proposed by Robbins (1952), which models decision making as a set of actions of initially unknown utility. A wide range of solutions have been proposed for this problem, ranging from non-stochastic (Auer et al., 2003) to combinatorial (Cesa-Bianchi and Lugosi, 2012) methods. Moreover, a common theme in bandit-style problems is the provision of external *expert* inputs, e.g. (Cesa-Bianchi et al., 1997; de Farias and Megiddo, 2004), which are used to inform action selection.

Definition 2.2.1 ***Adversarial bandit problem*** *(Auer et al., 2003): An adversarial bandit problem is specified by the number K of possible actions, where each action is denoted by an integer $1 \leq i \leq K$, and by an assignment of rewards, i.e. an infinite sequence $\mathbf{x}(1), \mathbf{x}(2), \ldots$ of vectors $\mathbf{x}(t) = \langle x_1(t), \ldots, x_K(t) \rangle$, where $x_i(t) \in [0,1]$ denotes the reward obtained if action i is chosen at time step t.*

2.2.2.1 Connection to interaction shaping

Our approach to the interaction shaping problem is inspired by the bandit and regret minimisation ideas. Our goal is to recover the utility of different actions against a given adversarial agent, in order to compute temporally extended policies that are likely to achieve a global strategic goal. In Chapter 3, we extend the above concepts to multi-robot games, where uncertainty and observability limitations pose additional constraints that must be addressed by an autonomous agent. Then, in Chapter 4, we introduce human demonstrations of strategic behaviours as expert knowledge in the process of shaping interactions. Finally, in Chapter 5, we build an empirical Bayesian

learning framework on top of these demonstrations, which is intended to minimise the regret of selected strategies over a specified time horizon.

2.2.3 Ad hoc coordination

An emerging problem in the autonomous decision-making literature is the formalisation of ad hoc coordination in multi-agent teams (Stone et al., 2010). The challenge in ad hoc coordination lies in developing autonomous agents that can form teams with a priori unknown teammates, and collaborate effectively in a joint task. The original approach to this problem (Stone and Kraus, 2010) was based on a multi-armed bandit framework (similar to the one described in the previous section), where the arms correspond to the possible actions each agent may take in this collaborative setting. Subsequent work on this domain has focused on the development of role formation algorithms (Genter et al., 2011), and their empirical evaluation in relevant scenarios, such as the pursuit domain (Barrett and Stone, 2012).

A recent line of work that comes close to our interaction shaping framework deals with leadership protocols in ad hoc teamwork, where an agent is tasked with guiding a group of other teammates towards a desired goal. Agmon and Stone (2012) introduce a graphical model for leadership in joint action settings, discussing different variants such as having multiple leaders or considering an extended history of states in action selection. This model groups related sets of joint actions together, with edges indicating the ability of switching from one action to another. The optimal set of joint actions is computed through a polynomial time search algorithm. Furthermore, Genter et al. (2013) model collaborating agents as a leader-based flock, whose aim is to jointly optimise a team-level utility function. This work distinguishes between the cases when the flock consists of stationary and non-stationary agents, discussing how convergence is affected in each case.

2.2.3.1 Connection to interaction shaping

Despite their different scope, there are some parallels between the ad hoc coordination and interaction shaping problems. Both problems deal with behavioural influence in multiagent domains, where an agent seeks to impact the decisions of one or multiple other agents towards a desired outcome. Moreover, both frameworks are focused on algorithms that are robust with respect to a wide range of behavioural profiles and interacting agent characteristics. However, we also note some importand differences be-

tween the two approaches. First, the ad hoc problem is aimed at cooperative decision-making problems, whereas interaction shaping focuses on influence over adversarial agents and features no collaboration. Thus, a shaping agent does not share resources or select roles within a team; instead, it must exploit its own capabilities in order to attain its goal. Second, the interaction shaping problem accounts for human-controlled adversaries whose behaviour may change during the interaction. By contrast, in ad hoc coordination, non-stationary agents are typically endowed with some additional capability (e.g. the ability of ad-hoc agents to move in the flocking problem) that impacts their ability to influence other agents. Third, interaction shaping is focused on empirical, interactive learning problems in robotic and human-robot systems, whereas ad hoc teamwork is currently geared towards analytical evaluation and theoretical guarantees in simulated domains. Thus, the two frameworks can be essentially viewed as two heterogeneous approaches to broadly related computational problems, with the common aim of influencing interacting agents.

2.2.4 Influence over adversarial agents

In multi-agent systems, opponent modeling is often concerned with influence over the beliefs of other adversarial agents. The aim is to attain intelligent behaviours that emulate human traits, for example, *bluffing* in poker games (Southey et al., 2005). In experimental robotic systems, these types of influencing behaviours largely remain an open problem. Perhaps the most notable recent such example is the work by Wagner and Arkin (2011), which studies the concept of *deception* in interactions between adversarial robots. The experimental domain considered in that paper (Figure 2.2) is structured around a hide-and-seek game between two adversarial robots, where the hider attempts to provide misleading information to the seeker in order to conceal its true intentions. This approach is based on a simple Bayesian network representing the causal relationship between the manipulable obstacles (left, right, centre marker), and the belief of the seeker on the hider's location.

In the domain of human-robot interaction, Short et al. (2010) investigate how people perceive a robot that attempts to deceive them in a rock-paper-scissors game. The key finding of their experiment was that several subjects exhibit greater social engagements when interacting with a cheating robot. Another related study (Vázquez et al., 2011) focuses on the responses of human participants to the decisions of a deliberately deceptive robot referee, in the context of a simple interactive game.

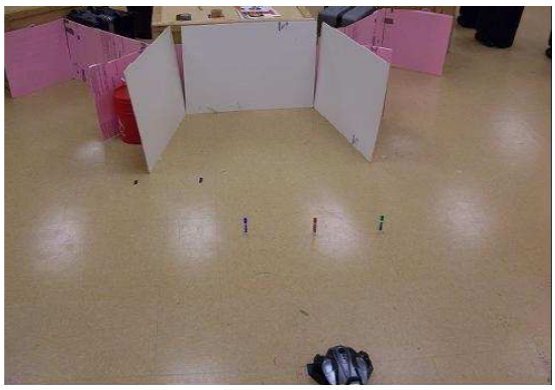


Figure 2.2: Experimental setup for the deception problem studied by Wagner and Arkin (2011). A robot hider (shown in the figure) must navigate to one of three possible hiding locations (left, centre, right). Another robot (seeker, not shown) must correctly identify the location selected by the hider. The environment also features three different obstacles (represented by whiteboard markers), which are positioned along the paths to the three hiding locations. The hider can deceive the seeker by knocking off a marker that does not reveal its intended destination (e.g. knock off the left marker while heading towards the right location).

2.2.4.1 Connection to interaction shaping

Despite these early approaches towards influencing behaviours in interactive robotic systems, there is a lack of a clear underlying theoretical model of behaviour shaping. Furthermore, these examples represent fairly simple scenarios with very small state and action spaces, while also suffering from lack of diversity in the exhibited strategies. By contrast, there is no clear notion of how an autonomous robot can influence an interacting adversary in a complex, continuous space-action domain, which also features physical limitations and sensory uncertainty. In this thesis, we seek to *learn* and reproduce influencing behaviours in a principled manner, in experimental environments where the characteristics (e.g. human/autonomous) of the adversary are not known a priori. To achieve this objective, our learning formulation combines several established probabilistic motion planning and decision making techniques, such as action sampling, iterated reasoning, data-driven approximation, and Bayesian inference. Thus, the resulting framework leads to interactive strategic learning in physically grounded adversarial multi-robot environments.

2.2.5 Adversarial interactions in graphics

Adversarial modeling is also important in computer graphics an animation, where there is a need for fluent control of virtual interacting characters. Wampler et al. (2010) present a game-theoretic control method, based on the assumption that characters act simultaneously and not in turns. This technique combines offline learning of a controller function through iterative approximation and compression heuristics, and online adaptation to the observed actions of the adversary. Similarly, Shum et al. (2008) describe a character control method which is split into a preprocessing and a run-time stage. In the first phase, samples of the desired motion are collected and encoded as an interaction graph, which can be queried for the optimal action through dynamic programming or min-max search. In the second phase, actions are selected based on the outcome of the offline stage, with the option of dynamically recomputing the policy based on user input.

Our approach to interaction shaping is similarly divided into an offline phase, where the desired interaction templates are learned from provided demonstrations, and online adaptation to the actions of the adversary. However, our model is more focused towards long-term influence over adversaries, based on an iterated prediction and sampling. Furthermore, unlike the above works, we empirically learn a model of the interacting agent's behaviour, and use it both to predict the future state of the interaction and to select the optimal action in a Bayesian manner.

2.3 Shaping in decision making

The term "shaping" has been used in various contexts in the autonomous agents literature. *Reward shaping* (Ng et al., 1999) is the process of affecting an agent's learning process by providing additional rewards, as a means of inciting desirable behaviours. Given an MDP M, the reward function, $R : S \times A \times S \mapsto \Re$ (defined as in Section 2.1.1), is augmented with an additional, manually specified *shaping reward function*, $F : S \times A \times S \mapsto \Re$, operating on the same spaces. This modification leads to a new MDP, M', where the agent experiences a reward $R(s,a,s') + F(s,a,s')$ when transitioning from s to s' through action a, as opposed to just $R(s,a,s')$ in M. This is followed by a discussion on the conditions under which an optimal policy in the modified MDP, M', will also be optimal in the original MDP. Thus, shaping here refers to the external influence of an agent's learning process, through the provision of appropriately defined

rewards.

Autonomous shaping (Konidaris and Barto, 2006) considers an extension to the above concept, where an agent learns a shaping function from its own experience and not through external specification. In this approach, the agent is tasked with solving a sequence of reward-linked goal directed problems. The main idea is that prior experience in solved tasks can be used to inform solutions in later, related tasks. Given a sequence of n problems, $S_1, \ldots, S_n$, the agent learns a value function V_j for each problem S_j. Then, training examples from different tasks are jointly used to learn a transfer function L, which can be applied to estimate the value of novel problem instances. Thus, the agent can autonomously shape its future rewards by reusing knowledge from previously solved problems.

Other approaches use the term "shaping" as defined in the animal learning literature, i.e. "the process of training by reinforcing successively improving approximations of the target behaviour" (Bouton, 2007). In this context, Knox and Stone (2009) define a shaping problem similar to the above, where an agent interactively uses human reinforcement signals to select actions with expected high reward.

Definition 2.3.1 ***The Shaping Problem*** *(Knox and Stone, 2009) Within a sequential decision-making task, an agent receives a sequence of state descriptions, $s_1, s_2, \ldots \in S$, and action opportunities, $a_i \in A$ at each state. From a human trainer who observes the agent and understands a predefined performance metric, the agent also receives occasional positive and negative reinforcement signals correlated with the trainer's assessment of recent state-action pairs. The problem is, how can an agent learn an optimal policy, $\pi : S \mapsto A$, with respect to the performance metric, given the information contained in the input?*

Thus, this version of shaping involves more explicit interaction between the learning agent and the human trainer, by modeling the provided reinforcement signals and using them to interactively update the action selection policy.

2.3.1 Related concepts

A different concept that is closely related to shaping is *active indirect elicitation* (Zhang and Parkes, 2008), where an agent's reward function is inferred from incentives supplied by an external interested party. The aim is to influence the agent into following a policy that maximises the expected value for the interested party. This approach bears

similarities to the reward shaping problem discussed above, as the supplied incentives carry the form of user-defined functions, which are applied on top of MDP-based reward functions.

An alternative approach to reasoning about adversaries is the use of external advice in the form of a coach or other expert agent. Riley and Veloso (2002) introduce a distributed planning method coupled with probabilistic opponent modeling, where a simple temporal network is used to represent team-level coordinated movements. Furthermore, Riley et al. (2002) extend this idea to extract models from past robotic soccer games and use them in specific game play contexts, such as set plays, formation learning, and passing. In both works, a coach acts as an external agent who observes the actions of the opposing side, and devises a plan that is suited to these observations and the predicted behaviour of the adversaries.

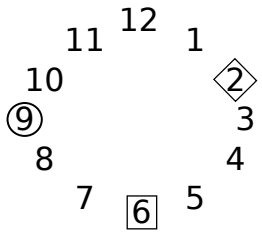


Figure 2.3: The lemonade stand game (Zinkevich, 2012). The game is played by three lemonade vendors (indicated by the circle, square, and diamond shapes) on a circular island with 12 different beaches. The vendors must determine, over a series of episodes, where to place their stands, in order to maximise their revenue. Vendors are not initially aware of the choices of their competitors. The expected reward can be formulated as the sum of the distances to all neighbouring vendors, assuming that customers are uniformly distributed across the beaches.

Behaviour shaping has also been considered in multi-agent interactions combining elements of competition and collaboration. One such example is the lemonade stand game (Zinkevich, 2012), where the reward experienced by an agent depends both on its own decisions and those of the other interacting agents (Figure 2.3). Thus, an important challenge in the lemonade stand game is to determine, over a series of episodes, a behavioural model for the responses of the interacting agents. Wunder et al. (2011) propose an iterative reasoning framework for this problem, based on a parametrised version of IPOMDPs. This framework is shown to outperform previously deployed implementations in lemonade-stand game competitions.

2.3.2 Connection to interaction shaping

The interaction shaping problem considered in this thesis is similarly concerned with influence over the behaviour of an interacting agent. Our problem is different to the studies of Ng et al. (1999), Knox and Stone (2009), Zhang and Parkes (2008), Riley and Veloso (2002), and Riley et al. (2002) in that there is no interaction in the reward learning process with human experts or trainers. Instead, human demonstrations are provided only in an offline phase (Chapter 4), and then used as a basis for an online learning algorithm. Thus, our problem comes closer to the works by Konidaris and Barto (2006) and Wunder et al. (2011), where the agent learns from experience fully autonomously. However, our scope is quite different to Konidaris and Barto (2006), whose work primarily focuses on the transfer problem, while also not featuring interaction with strategic adversaries. Furthermore, in relation to the coaching works discussed above, our method does not feature coordination with other agents or an external expert (the coach), but focuses instead on how a single agent can learn to shape an interaction. Moreover, in relation to Wunder et al. (2011), we consider purely *adversarial* interactions, where the learning agent is tasked with shaping the behaviour of *non-cooperative* agents, whose goals are conflicting with its own. This is a challenging problem that comes closer to most realistic human-robot interactions (which are beyond the scope of all the above works), where robots do not have explicit control over human actions.

2.4 Human demonstration and interaction strategies

In Chapter 4, we formulate an algorithm for learning interaction strategy templates from potentially imperfect human demonstrations. Learning from human demonstration is a common choice for programming autonomous robots; see Billard et al. (2008) for an overview of the problem and Argall et al. (2009) for a comprehensive survey of related techniques. Many of these works deal with demonstrating specific motor skills to robots, e.g. (Pastor et al., 2009; Lee et al., 2011; Chatzis et al., 2012), using a wide range of statistical machine learning tools. Our approach is based on the use of Gaussian Mixture Models (GMMs), which have been previously used for interactive policy learning from human demonstrations (Chernova and Veloso, 2007) and motor task learning (Calinon et al., 2006, 2010). In robotic soccer, the primary experimental domain of this thesis, Gaussian techniques have been applied to provide illustrations of

specific manoeuvres, such as ball grasping for four-legged robots (Grollman and Jenkins, 2007). In simulated robotic soccer domains, human demonstrations have been previously used in the design of more general, team-level strategies (Aler et al., 2005).

2.4.1 Connection to interaction strategy learning

Unlike many of the above works, our focus is on learning *interaction strategies* instead of specific single-agent skills. To learn such strategies, we make demonstrators exhibit a strategic behaviour by teleoperating a robot in interactions with another adversarial robot. In this context, the recorded training data set does not only consist of the demonstrator's control inputs and the trajectories of the teleoperated robot, but also of the trajectories of the adversarial robot. These inputs are jointly used to learn a set of strategy *templates*, which are defined interactively to account for both adversaries. These templates can then be autonomously generalised and adapted to novel strategic opponents, based on the observed state of the interaction, without the provision of any external human guidance. Thus, even when the demonstrations are provided in the context of an interaction with a baseline heuristic autonomous opponent, they can be subsequently synthesised to form strategies that can challenge more sophisticated human adversaries. This form of interaction strategy learning constitutes a novel connection between the human demonstration and opponent modeling fields.

Another distinguishing feature of our approach is the nature of the collected traces. In the robotic soccer problem we consider, demonstrations are provided in the context of interactive games between robots. These demonstrations are annotated only based on their success and not on their optimality. For example, when learning strategies for the striker, we only record whether a goal was scored in a trial, and ignore features such as lags in demonstrator decisions. Thus, a significant proportion of our traces are *suboptimal*, as users are focusing on the overall game objective and not explicitly trying to demonstrate an optimal behaviour. This distinguishes our approach from several state-of-the-art learning works, (e.g. Abbeel and Ng (2004)), where demonstrations are treated as expert solutions to a single optimal control problem.

Learning from imperfect demonstrations has been studied by Nemec et al. (2011), where previous experiences acquired by the robots are used to guide sensorimotor learning, and by Grollman and Billard (2011), where unsuccessful traces form an integral part of the learning process. However, these works also consider single-robot motor skill learning, so there is no notion of interacting agents as in our approach.

2.5 Human-robot interaction and perception

2.5.1 Forms of interaction

In Chapter 6, we present the results of a user study on the effects of limited perception on human decisions in interactive multi-robot tasks. In this study, subjects are evaluated in different teleoperation tasks requiring interaction with an autonomous robot. The tasks range cooperative forms of interaction to fully adversarial, the latter requiring subjects to outperform the autonomous robot in an interactive setting.

Human-robot interaction is often considered in the context of cooperative tasks, where the interacting parties must collaborate to achieve a common goal, e.g. cooperative object manipulation (Edsinger and Kemp, 2007; Dominey et al., 2008). Many such interactions are centred around the ability of the robot to follow instructions from a human, in order to fulfil its role in the task, e.g. (Tenorth et al., 2010; Lallée et al., 2010b). Furthermore, several studies in human-robot collaboration are concerned with modeling human intentions; various approaches have been proposed to this effect, such as velocity-based impedance control (Duchaine and Gosselin, 2007), Dynamic Bayesian Networks (Schrempf et al., 2005), or interaction history records (Dominey et al., 2008).

2.5.1.1 Connection to our interactive domains

In our work, we look at the related problem of how humans account for the intent of autonomous robots in dynamic interactions, in both collaborative and adversarial settings. Moreover, we do not allow communication or any form of information exchange between interacting parties; the human must infer the robot's intent only through visual observation which is progressively restricted. Thus, even the cooperative tasks we consider present users with different challenges than corresponding problems in the existing literature. Our study therefore analyses the impact of novel factors (strategic content of task combined with limited perception) affecting interactive teleoperation, and thus introduces scenarios where human decisions may be less robust than those made by autonomous agents. This comparison and division of labour between human control and robot autonomy is an important area of study in the robotics literature; see Parasuraman et al. (2000) for a more detailed discussion on this problem.

2.5.2 Perceptual constraints

The influence of perception in human-robot cooperation has been previously examined in the context of recognising actions and learning skills from observation, e.g. (Johnson and Demiris, 2005; Lallée et al., 2010a). Learning from demonstration under perceptual constraints was the focus of a study by Crick et al. (2011), which showed that robots can learn more effectively when the perception of human demonstrators is restricted to be similar to their own. An interesting result in that study was that robots were able to learn more quickly from restricted-perception demonstrations, even though their quality was often inferior to full-perception ones. A similar study on human teleoperation was conducted by Lathan and Tracey (2002), where user performance was found to be correlated to the availability of perceptual information.

2.5.2.1 Connection to our study

In our study, we are similarly interested in assessing the effects of constrained visibility on human performance. However, our focus is not on learning from demonstrations provided independently by a human user, but instead on evaluating human performance in a purely *interactive* environment, where humans and robots *simultaneously* engage in cooperative and adversarial tasks. Our results demonstrate that restricted perception may have a significant impact on decisions in complex, strategic tasks – this complements the results of existing studies, which show that alternative factors such as fatigue have a limited impact on human teleoperation performance (Mavridis et al., 2012).

2.6 Related domains in robotic soccer

Throughout this thesis, we use the robotic soccer penalty shooting example as an illustrative experimental problem. A domain that closely resembles our problem is Segway soccer (Browning et al., 2004; Argall et al., 2006), which involves mixed teams of robots and humans mounted on Segways (Figure 2.4). Browning et al. (2004), use teleoperation to provide demonstrations of human play, which are then generalised using locally weighted regression (Schaal and Atkeson, 1998).

Our experimental setup shares a similar motivation in that humans and robots compete in an adversarial task with the same physical capabilities. However, we note the following important differences between the two domains. First, the pace in penalty shooting is faster (as demonstrated in the supporting videos (Valtazanos, 2012a,b)) than

Figure 2.4: Illustration of Segway soccer (Argall et al., 2006). Autonomous robots and humans mounted on Segway compete in a soccer match using identical platforms. This setup comes close to our penalty shooting experiments, where – otherwise identical – autonomous and teleoperated robots interact in a competitive game.

in Segway soccer, requiring more frequent interactive decisions by both sides. Second, being a one-to-one challenge between a teleoperated and an autonomous robot, our task is more explicitly adversarial than Segway soccer, which features mixed teams and elements of coordination. Thus, although our domain features fewer robots, it ensures that humans and robots get equal "playing time". By contrast, in a mixed-team game, humans could dominate the interaction and supplant the role of the robots.

Robotic soccer penalty shooting has been previously studied by Hester et al. (2010) as a reinforcement learning problem for learning better kicking motions. However, the goalkeeper was a static player who did not move, so there was no strategic adversary.

2.7 Hybrid systems and particle filtering

In Chapter 3, we present the Reachable Set Particle Filter, an algorithm for computing the state of adversarial agents from noisy sensory observations. This algorithm combines dynamical constraints (hybrid system formulation) and data-driven estimation (particle filtering). In this section, we review key concepts from these two domains.

2.7.1 Hybrid systems

A key concern in our work is the modeling of an opponent's behaviour, which is dictated by choices over discrete behavioural modes and underlying continuous dynamics. A good framework for thinking about such problems is available within the control theory literature, where systems with joint discrete and continuous dynamics are known as *hybrid systems* (Tomlin et al., 2000, 2003; Mitchell et al., 2001, 2005).

A major application of hybrid system modeling has been the formal description

of aircraft collision avoidance as a *pursuit-evasion game* (see Vidal et al. (2002); Gerkey et al. (2004); Karaman and Frazzoli (2010); Bhattacharya and Hutchinson (2010) for examples of related pursuit-evasion applications) between two adversaries (Figure 2.5). Each aircraft assumes the role of an *evader* seeking to avoid collision with an adversary, who is modeled as a *pursuer* with the exact opposite goal. An evader has a notion of a *target* set of unsafe states, which must be avoided to prevent collision.

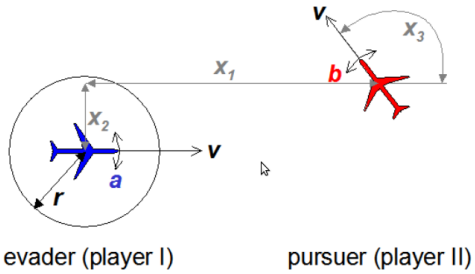


Figure 2.5: Illustration of the two-aircraft collision avoidance example (Mitchell, 2007). The second aircraft is modeled as a pursuer seeking to cause a collision with the first aircraft. Integration of system dynamics (defined in terms of velocity bounds) over time lead to the computation of control inputs that may cause a collision in a future time.

A key innovation of this approach is the introduction of *reachable* sets of states, which can be classified in one of two ways. A *forward reachable set* is the set of states that can be reached from some given initial configurations. A *backward reachable set* is the set of states that may give rise to trajectories terminating in a target set of unsafe states. By computing these unsafe states, one can also determine trajectories that can lead to their avoidance. In prior work (Valtazanos and Ramamoorthy, 2011c), we presented a procedure through which backward reachable sets can be used in the context of motion planning in adversarial robotic environments.

2.7.2 Particle filtering

The particle filter (Gordon et al., 1993) has become a popular tool for state estimation in uncertain environments, due to the ability to flexibly model arbitrary probability distributions. A particle filter typically comprises a *prediction step*, where a fixed set of hypotheses is computed based on a known prior distribution, and an *update step*, where the likelihood of these hypotheses is updated based on the most recent observations. These algorithms have been used to estimate adversarial models from experience in strategic games such as poker (Bard and Bowling, 2007). A related game-theoretic

concept is found in empirical games (Jordan and Wellman, 2009), where one attempts to extract strategic profiles in a data-driven fashion.

2.7.3 Connection to our algorithm

Our proposed algorithm combines the relative merits of hybrid systems and particle filters in the context of interactive adversarial state estimation. In particular, we use formal dynamical constraints, as modeled in hybrid systems, in the prediction step of a particle filter algorithm tracking the state of an interacting agent. Thus, sensory observations that are inconsistent with the generated predictions can be discarded in the update step. This differentiates our approach to most prior works in hybrid systems, where uncertainty in sensory readings is not modeled directly in the system dynamics. Furthermore, we introduce a tighter coupling between probabilistic and game-theoretic concepts, which are jointly used to estimate the state of interacting adversarial robots.

2.8 Unconstrained motion capture

In Chapter 7, we propose an algorithm for simultaneous posture and position tracking in unconstrained environments, with application to physical strategic human-robot interaction. Our approach is based on a combination of optical and inertial sensing motion capture, which are jointly used to learn a model of human motion through low-dimensional manifolds. This section reviews related motion capture literature and technologies, illustrating the key innovations introduced by our method.

2.8.1 Human motion capture systems

Traditional *optical* motion capture systems, e.g. the Vicon System (`http://www.vicon.com`), use an ensemble of high-resolution cameras to track the locations of reflective markers placed on the body of a subject. The marker positions are used to compute the full pose (position and posture) of the subject. However, optical systems suffer from a number of drawbacks that impact their applicability in complex human-robot interaction scenarios. First, motion capture must be carried out in dedicated studios, which are often expensive to set up and maintain. Second, these systems have a small area of capture, and subjects cannot be tracked outside its boundaries, e.g. when moving in and out of rooms, or navigating along corridors in a building. Third, occlusion problems impact the ability of these systems to track subjects consistently.

A recent development has been the creation of devices combining stereo cameras and depth-estimating sensors, most notably the Microsoft Kinect (Figure 2.6(a)). The combined output of these sensors is used to generate a three-dimensional point cloud, which can subsequently be analysed to determine the pose of a tracked subject, or fit the data to a skeleton model (Shotton et al., 2011). These devices are significantly cheaper than traditional optical systems and also remove the need for markers on the subject's body. Furthermore, the portability of the sensors allows for anyplace motion capture. However, like traditional optical systems, a Kinect must also be fixed in order to track subjects reliably, and the volume of capture is limited to approximately 15m^3. This makes it unsuitable for tracking subjects in large or unconstrained spaces.

(a) Microsoft Kinect. (b) Orient-4 device.

Figure 2.6: Motion tracking platforms. (a): Optical source – Kinect device with two cameras and a depth-finding sensor. (b): Inertial measurement unit – Orient-4 device with tri-axial gyroscopes, accelerometers, and magnetometers.

An alternative to optical systems are wireless *inertial* sensing platforms, such as the Orient platform (Young et al., 2007) (Figure 2.6(b)). These systems collect data from an ensemble of sensor nodes placed on the subject's body. Each device typically consists of 3-axis inertial sensors such as gyroscopes, accelerometers, and magnetometers. These sensors jointly estimate the rotation of the body part the device is placed on, relative to a fixed point on the subject's body. Data from the different body parts are transmitted wirelessly to a base station and aggregated to determine the overall posture of the subject. Inertial sensor networks have been successfully used in tracking various complex physical activities, such as Tango dancing (Arvind and Valtazanos, 2009).

Due to the wireless transmission and capture of data, inertial sensing avoids the occlusion problems arising in optical systems. More importantly, as no fixed tracking source is required, inertial sensing can track subjects in a greater variety of environ-

ments and in larger areas than optical systems. The main drawback of inertial sensing is the relative rotational nature of sensory estimates, which means that only postures can be determined directly. Unfortunately, this approach does not extend to absolute spatial positions, as computations are performed relative to a stationary reference point.

Position tracking using inertial measurement units has been the subject of several studies. Most of these works use a *model-based* approach, where inertial sensor measurements are filtered through a position tracking model to predict the most likely translation of the tracked subject. The employed models range from analytically defined Kalman filters (Young, 2010; Corrales et al., 2008; Foxlin, 2005) or particle filters (Kobayashi and Kuno, 2010), to alternative heuristic approaches based on gait event detection (Ojeda and Borenstein, 2007; Yun et al., 2007; Feliz et al., 2009).

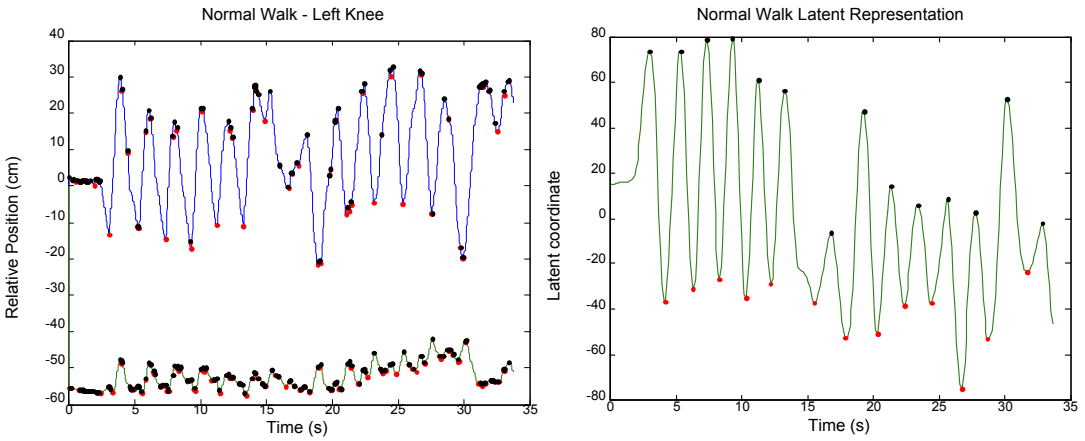
Optical and inertial sensing systems have been previously combined in the context of motion reconstruction. Tautges et al. (2011) use sparse accelerometer data to search over a large database of marker-based motions, and retrieve suitable fragments that can be synthesised to form plausible motions. This approach is based on a combination of local graph search and cost-based energy minimisation. This latter part of the algorithm is related to our method, which similarly uses a set of weights in order to map postures to translations. However, we do not use any motion databases or graph search in the mapping phase; instead we learn a generative model from recorded optical-inertial motion sequences, and use it to compute translations for novel instances.

2.8.2 Dimensionality reduction in sensor networks

Dimensionality reduction has been used in inertial sensor networks as a *discriminative* model for activity recognition and gait phase detection (Yang et al., 2008; Valtazanos et al., 2010, 2013a). In this context, a high-dimensional motion sequence is projected into a latent, low-dimensional space, where salient patterns can be detected more effectively (Figure 2.7). Thus, this approach alleviates the need to look at individual sensors when analysing multi-dimensional motion sequences.

2.8.2.1 Connection to our method

Compared to prior work, our motion learning framework is different in being a *model-free* method with respect to the measurement units, where no assumptions are made on the sensor placements or the nature of the motion being performed. Instead, the aggregated sensory data is treated as a single *feature vector*, from which a mapping to



(a) (b)

Figure 2.7: Example of dimensionality reduction of inertial sensor network data from a walking motion sequence (Valtazanos et al., 2010). (a): Example of individual joint data readings. (b): Representation of all sensor readings in a latent, low-dimensional subspace. Black and red dots indicate local maxima and minima, respectively, which correspond to hypotheses on the boundaries between consecutive steps. In the latent space, false positives are avoided, and segmentation points are more easily identifiable.

whole-body translations is learned. Thus, in our experiments (Chapter 7), we compare against the established model-free method of double integration of accelerometer data. Nevertheless, our approach can in principle be combined with existing models, where generated translations can be used as predictive estimates for a filter.

In our work, we use low-dimensional subspaces as *generative* models for whole-body translations. In this generative context, learned manifolds have been used in robotics, in order to facilitate imitation of human gaits by humanoid robots (MacDorman et al., 2004; Chalodhorn et al., 2007). By adopting this flexible representation, our objective is to similarly approximate a wide variety of motion dynamics and structure.

2.9 Summary and motivation for our approach

One of the main open problems in the existing literature is the lack of a theoretic model for behavioural shaping in interactions between humans and robots (as discussed in Section 2.3). This thesis seeks to address this issue by proposing and experimentally validating a model for strategic interaction. However, since addressing this problem directly in a physical setting is hard, the thesis follows an incremental approach to-

wards the desired end. Our choice of robotic soccer as the main experimental domain is inspired by existing approaches (Section 2.6) to strategic human-robot interaction. In this context, Chapter 3 introduces simple interactions between adversarial agents in simulated worlds, focusing on the simultaneous reasoning about sensing (c.f. Section 2.7) and strategic (c.f. Section 2.2) uncertainty. Chapter 4 raises the complexity of the experimental setup through the incorporation of physical NAO humanoid robots. Furthermore, humans are introduced in the interaction loop, both as strategic agents and as demonstrators of intelligent interactive behaviours (c.f. Section 2.4). The interaction templates presented in this chapter extend the opponent modeling formulation described in Chapter 3. These concepts are further unified in Chapter 5, which introduces our main approach to the interaction shaping problem. Here, the agent learns the utility of different interaction templates through repeated interaction, and selects strategies that are likely to attain a strategic goal in a future time horizon (c.f. Section 2.1). Nevertheless, the effectiveness of these shaping behaviours largely depends on the availability of sensory information to the interacting agents, which inevitably influences their actions. Thus, Chapter 6 evaluates the effects of asymmetry of information on human decision making, inspired by the studies described in Section 2.5. Finally, Chapter 7 is a step towards attaining direct strategic interactions, where humans are physically present and not merely operators of robots. Our approach to this problem is inspired by existing motion capture systems and inference algorithms (c.f. Section 2.8) – here, we show how these systems can be combined to yield the sensory information that is needed for decision shaping in complex physical settings.

Chapter 3

Sensing and strategic uncertainty in multi-robot interactions

3.1 Overview

Even as the autonomous robotics community pushes the frontier of what is possible by a robot requiring decreasingly less guidance from any external sources, we realise that some of the most exciting opportunities are to be found in a middle ground where an autonomous robot interacts with other agents – including people – in a mixed-initiative or social setting. However, we also find that such interactive behaviour – involving multiple objectives, constraints, ambiguity and incompleteness of knowledge – can be even more challenging than the fully autonomous scenario. Part of the reason for this is the difficulty of modeling sophisticated strategic interactions in a way that is both principled and practicable.

When these conceptual difficulties are coupled with more pragmatic considerations of hardware and processing power limitations, we find that even seemingly simple "intelligent" moves, such as passes and dribbles in a robotic soccer game, are scarce and often require explicit hand crafting of everything but a few open parameters. In a domain such as robotic soccer, autonomous robots must face uncertainty in their own egocentric beliefs, incompleteness and uncertainty in their knowledge of the strategies of their adversaries and physical limitations such as a very limited field of view using a relatively mediocre camera.

In this chapter, we focus on the problem of robust strategic decision making in adversarial games with physical limitations in action and perception. We propose an approach to devising strategic interactive behaviours for autonomous robots, illustrated

using the robotic soccer domain. We also argue that constraints (in action/perception) need not be viewed only as a feature to be overcome or eliminated. As we show, successful interactive behaviour often requires the exploitation of these constraints, leading to interesting forms of motion strategies. We develop this theme in this chapter, a decision-making framework for physical multi-robot games, based on the following high level concepts:

- **Intent inference**: In the absence of a precise model of its adversary's strategic behaviour, a robot may approximate it using a finite collection of *intent templates*, each modeling a single *coarse behavioural class*. These templates are combined probabilistically into a single distribution that predicts the (re)actions of the adversary.

- **Escape strategies**: In a noisy, physically embodied, partially observable environment, robots may benefit from exploiting the observability limitations of their adversaries. We refer to such moves as *escape strategies*, as they seek to reduce the amount of information available to the adversaries and influence their decisions.

- **Probabilistic adversary state estimation**: As robots must deal with noisy and incomplete information, they require a mechanism for 'filtering' their observations of the adversary. We propose the *Reachable Set Particle Filter*, a probabilistic state estimation algorithm combining a formal characterisation of the dynamical constraints of a robotic system, with a data-driven estimation procedure. A reachable set characterises, *for all* instances of a class of strategies, the states that might be reached at *some* future point. Thus, it forms a powerful addition to the particle filter which, in the basic formulation, does not account for such constraints.

- **Regret minimisation**: In realistic games with uncertainty, robots can achieve the full benefit of strategic modeling only if they *adapt* to and *learn* from the actions of their adversaries. We use the regret minimisation to infer online the effects of probabilistically selected intent templates, and adjust their distributions to reward retrospectively optimal strategies.

The proposed framework brings together ideas from probabilistic modeling, game theory, and strategic reasoning, allowing for online decision making in adversarial robotic environments with physical constraints.

In the remainder of this chapter, we first summarise our method and system formulation (Section 3.2), and then we present results from experimental evaluation in simulation (Section 3.3). Finally, we review the key contributions of the chapter, and discuss the connection of this work to the remainder of this thesis (Section 3.4).

3.2 Method

We are interested in the problem of decision making by an autonomous robot with *physical limitations*. Such limitations include: limited velocities (including, perhaps, non-holonomic constraints), noisy locomotion, noisy perception with limited sensing resources, limited methods of object manipulation (e.g. kicking a ball to a specific point). We develop our framework in the context of the *robotic soccer* domain, as it features the above types of uncertainty, together with strategic adversaries of unknown capabilities. However, the underlying ideas could be extended to other strategic games and general forms of interaction.

3.2.1 Preliminaries

3.2.1.1 Notation

We consider an interaction involving a total of N autonomous robots. Let r_i refer to the i-th robot, $i = 1..N$. Furthermore, let $s \sim D$ be an abbreviation for drawing a sample s from a set or distribution D, and let $\texttt{dist}(P_1, P_2)$ denote the Euclidean distance between two points P_1 and P_2, with the special case $\texttt{dist0}(P) \doteq \texttt{dist}(P, \langle 0, 0 \rangle)$ (distance from origin $\langle 0, 0 \rangle$ of egocentric frame). We consider **discrete time, continuous space** decision making (at time instants t), though in later chapters (Chapters 4, 5) we show how this process can be extended to accommodate continuous time.

3.2.1.2 State Estimation

As most autonomous robots are restricted to egocentric sensing, they compute the states of other robots *relative* to their own coordinate frame. For r_j, the collection of relative states of all other robots at time t gives the set of *robot beliefs*:

$$\mathcal{RB}_{j,t} = \{ \langle x^i_j, y^i_j, \theta^i_j, c^i_j \rangle_t \mid i = 1..N, i \neq j \} \tag{3.1}$$

where $\langle x_j^i, y_j^i, \theta_j^i \rangle_t$ denotes the relative state of r_i as computed by robot j, in terms of planar coordinates x, y and orientation θ, and $0 \leq c_j^i \leq 1$ is a weight representing the robot's *confidence* on the belief. At the simplest level, the position component $\langle x_j^i, y_j^i \rangle$ of a belief is equal to a raw sensor reading, whereas the orientation is inferred from a history of positions (see Section 3.2.1.3). Correspondingly, relative soccer ball beliefs are given by $\mathcal{BB}_{j,t} = \langle x_j^B, y_j^B, c_j^B \rangle_t$. If the ball or a robot is visible at time t, its confidence weight is set to 1, otherwise it is set to the weight of time $t-1$ multiplied by a decay constant δ_c, $0 \leq \delta_c \leq 1$.

3.2.1.3 Orientation Estimation

Robots endowed with some sensing mechanism (e.g. vision and/or sonar) may compute the relative planar positions of their adversaries up to some approximation. Unfortunately, this approach does not extend to the relative orientations[1]. Instead, we use the autoregressive procedure INFERORIENTATION (Algorithm 1) to compute orientations based on past robot and ball beliefs. The algorithm relates the flow of a robot's motion to the position of the ball and the other robots, and probabilistically infers the orientation that is most likely leading to these movements.

3.2.2 The Reachable Set Particle Filter

Particle filtering helps robots overcome their sensing limitations. We present a variant to the original particle filter algorithm (Gordon et al., 1993) for autonomous robots, which we term **Reachable Set Particle Filter (RSPF)**. The main innovation is the definition of the proposal distribution for particle updates in terms of *backward reachable sets*, as described by Tomlin et al. (2003). If the dynamics of a system of robots is known, together with their corresponding minimum and maximum velocity bounds, then it is possible to compute future sets of states up to a – potentially infinite – time horizon. The worst-case backward reachable set $\mathcal{BRS}$ for r_i relative to r_j (assuming both robots are moving with their maximum linear velocities v_i and v_j) is obtained through the Hamilton-Jacobi-Isaacs Partial Differential Equation:

$$\frac{\partial v(q,t)}{\partial t} + \min[0, H(q, \nabla v(q,t))] = 0,\ v(q,0) = g(q), \tag{3.2}$$

[1]Unless sophisticated visual pattern recognition algorithms are used, whose computational cost would be prohibitive for the real-time decision making problems we are considering, and the kinds of humanoid robots we are targeting our approach at.

Algorithm 1 Relative Orientation Inference

INFERORIENTATION($\mathcal{RB}^i_j$,$\mathcal{BB}_j$)
Input: Robot beliefs $\mathcal{RB}^i_j$ for r_i, ball beliefs $\mathcal{BB}_j$
Auxiliary methods: `rand` {random number $\in [0..1]$}
$distTh \leftarrow 0.7m$ {distance threshold for interacting robot}
$\langle x^i_j, y^i_j, \theta^i_j, c^i_j \rangle_t \leftarrow \mathcal{RB}^i_{j,t}, \quad \langle x^B_j, y^B_j, c^B_j \rangle_t \leftarrow \mathcal{BB}_{j,t}$
$\langle x^i_j, y^i_j, \theta^i_j, c^i_j \rangle_{t-1} \leftarrow \mathcal{RB}^i_{j,t-1}$
$cdr_i \leftarrow$ `dist0`$(\langle x^i_{j,t}, y^i_{j,t} \rangle)$ {current distance of r_i}
$cdb \leftarrow$ `dist0`$(\langle x^B_{j,t}, y^B_{j,t} \rangle)$ {current distance of ball}
$cdbr_i \leftarrow$ `dist`$(\langle x^B_{j,t}, y^B_{j,t} \rangle, \langle x^i_{j,t}, y^i_{j,t} \rangle)$ {current distance of ball from r_i}
$ldbr_i \leftarrow$ `dist`$(\langle x^B_{j,t}, y^B_{j,t} \rangle, \langle x^i_{j,t-1}, y^i_{j,t-1} \rangle)$ {last distance of ball from r_i}
if `rand` < 0.7 **and** $((cdr_i > cdb$ **or** $cdbr_i < ldbr_i)$ **and** $cdr_i > distTh)$ **then**
    $\tilde{\theta}^i_{j,t} \leftarrow$ `atan2`$(y^B_{j,t} - y^i_{j,t}, x^B_{j,t} - x^i_{j,t})$ {r_i is further than the ball but has moved closer to it $\rightarrow$ infer that it is facing towards it}
else
    if `rand` < 0.5 **and** $cdr_i < distTh$ **then**
        $\tilde{\theta}^i_{j,t} \leftarrow$ `atan2`$(y^i_{j,t}, x^i_{j,t}) + \pi$
    else
        $\tilde{\theta}^i_{j,t} \leftarrow \pi$ {r_i is facing in the direction of r_j}
    end if
end if
return $\tilde{\theta}^i_{j,t}$

with Hamiltonian

$$H(q,p) = \max_{a \in \mathcal{U}_i} \min_{b \in \mathcal{U}_j} p \cdot f(q,a,b,v_i,v_j), \tag{3.3}$$

where $q = \langle x^i_j, y^i_j, \theta^i_j \rangle$, $f(q,a,b,v_i,v_j) = \dot{q}$ denotes the relative system dynamics, $g(q)$ is a scalar function representing the reachable set at $t = 0$ (e.g. $g(q) = \sqrt{{x^i_j}^2 + {y^i_j}^2} - C$, with C constant), and $\mathcal{U}_i, \mathcal{U}_j$ are the sets of permissible angular velocities. The HJI PDE is solved backwards in time until convergence. Tomlin et al. (2003) discuss in more detail the convergence properties of this method.

We assume that all robots have the same physical velocity constraints, so we compute a single reachable set $\mathcal{BRS}$ up to a time horizon of 1s. We now show how $\mathcal{BRS}$ can be used in a particle filter.

Each robot r_j maintains a separate particle filter for every other robot r_i. In each case, a set of P particles and weights:

$$\mathcal{RP}^i_j = \{\langle p_k, pw_k\rangle \mid k = 1..P\} \tag{3.4}$$

is maintained, where every $p_k = \langle \tilde{x}^i_j, \tilde{y}^i_j, \tilde{\theta}^i_j\rangle$ is a state hypothesis and pw_k is its associated weight, such that $\sum_{k=1}^{P} pw_k = 1$. Furthermore, we define $\mathcal{RM}^i_j$ as a second set of Q particles over the potential *one-step responses* of r_i:

$$\mathcal{RM}^i_j = \{\, \langle m_k, mw_k\rangle \,|\, m_k = \langle \tilde{dx}, \tilde{dy}, \tilde{d\theta}\rangle, k = 1..Q\,\}, \tag{3.5}$$

i.e. the potential moves r_i can take in a single discrete time step. Each set $\mathcal{RM}^i_j$ is initialised randomly. The *candidate* move at time t is $\bar{m} = \langle \langle x^i_{j,t}, y^i_{j,t}\rangle - \langle x^i_{j,t-1}, y^i_{j,t-1}\rangle$, $\textsc{InferOrientation}(\mathcal{RB}^i_j, \mathcal{BB}_j)\rangle$. The weight $\bar{mw}$ of a candidate is defined in terms of the reachable set $\mathcal{BRS}$, so that:

$$\bar{mw} = \begin{cases} 1/Q, \bar{m} \in \mathcal{BRS} \\ 0, \quad \bar{m} \notin \mathcal{BRS} \end{cases} \tag{3.6}$$

The oldest particle $\langle m_o, mw_o\rangle$ in $\mathcal{RM}^i_j$ is replaced by $\langle \bar{m}, \bar{mw}\rangle$. Following the replacement, all weights are normalised so that they add to 1. Thus, $\mathcal{RM}^i_j$ essentially acts as a predictive distribution for r_i, by combining both egocentric estimates and ground truth dynamics from the reachable set.

To complete the Reachable Set Particle Filter, we set $\mathcal{RM}^i_j$ as the proposal distribution for the prediction step of $\mathcal{RP}^i_j$. Then, at time t:

$$p_k \leftarrow p_k + \tilde{m}, \langle \tilde{m}, \tilde{mw}\rangle \sim \mathcal{RM}^i_j, k = 1..P \tag{3.7}$$

The remaining steps are similar to the algorithm described by Gordon et al. (1993). The observation likelihood distributions for the filter correction step should be suited to the robot's sensor specification. For example, the distribution for sonar readings should account for the minimum and maximum sensing range. Thus, noisy estimates that are inconsistent with the robot's hardware limitations can be discarded.

Through this filtering process, the state component of a belief is revised as the weighted sum of its associated particles:

$$\langle x^i_j, y^i_j, \theta^i_j\rangle = \sum_{k=0}^{P} p_k \cdot p_{wk} \tag{3.8}$$

3.2.3 Action types, actions, and strategic modes

Every robot has access to the following set of parametrisable **action types**:

$$\mathbf{AT} = \{\textsc{Move}(dx, dy, d\theta), \textsc{Kick}(kt, ks), \textsc{Scan}(dy, dp)\} \tag{3.9}$$

$\textsc{Move}(dx, dy, d\theta)$ corresponds to a desired displacement and turn; $\textsc{Kick}(kt, ks)$ executes a kick of a given type $kt \in \{left\_straight, right\_straight, left\_side, right\_side\}$ and speed factor $0 < ks \leq 1$, where $ks = 1$ corresponds to full speed; and $\textsc{Scan}(dy, dp)$ alters the robot's head yaw and pitch by dy and dp respectively, to allow scanning of a different region of the environment. An **action** α is an *instantiation* of an action type α_τ, e.g. $\textsc{Move}(0.1, 0.0, 0.0)$.

In order to improve the tractability of the action selection problem and cluster similar behaviours together, we also define a set of roles, or **strategic modes**:

$$\mathbf{M}^- = \{\textsc{Kicker}, \textsc{Defender}\} \tag{3.10}$$

A KICKER will always try to navigate to the ball and kick it, with additional constraints to avoid (potentially strategic) obstacles such as the adversaries. The DEFENDER mode is triggered when a robot cannot see the ball but is close to an adversary, so it instead attempts to block its path.

The mode μ and action type $\alpha\tau$ for r_j at time t is chosen **deterministically** using a decision tree, based on the actual beliefs $\mathcal{RB}_{j,t}$ and $\mathcal{BB}_{j,t}$. The chosen mode depends on the proximity of the ball and the relative position of the opponents. We label this procedure:

$$\langle \alpha\tau_t, \mu_t \rangle \leftarrow \textsc{SelectActTMode}(\mathcal{RB}_{j,t}, \mathcal{BB}_{j,t}). \tag{3.11}$$

3.2.4 Intent inference

A robot should be able to distinguish between different types of adversaries, and adapt accordingly to achieve its strategic goal. Clearly an exhaustive search over all possible actions and strategies available to a robot and its adversary would be both intractable and inflexible. Instead, we propose and define an **intent filter**, which is used to classify the observed movements of an adversary into coarse classes of strategic behaviours. The intent filter for the adversary r_i with respect to r_j is a set

$$I_j^i = \{\, \langle I_k, im_k, iw_k \rangle \,|\, k = 1..K \,\} \tag{3.12}$$

where I_k is one of K predefined **intent templates**, im_k is the next move currently predicted by I_k for r_j, and iw_k is its associated weight. In our robotic soccer model, we define the following coarse intent templates:

$$\mathbf{I}^{-} = \{\textsc{Static}, \textsc{Ball}, \textsc{Pursue}, \textsc{PredMove}\} \tag{3.13}$$

where STATIC predicts that r_i will not move at the next time step, BALL predicts a movement towards the ball, PURSUE predicts a movement towards r_j, and PREDMOVE sets the next move to a random weighted sample from $\mathcal{RM}^i_j$. Note that these templates are both *logic-based* (e.g. STATIC) and *data-driven* (e.g. PREDMOVE). For every logic-based template I_k, the rule for the next expected adversarial move im_k is explicitly defined (e.g. for BALL, the relative position of the ball to r_i is used to determine how the adversary will move towards it). Moreover, the adversary r_i may have an intent inference mechanism of its own, in which case its strategy may vary depending on the action chosen by r_j at time t (e.g. r_i may be more aggressive if it determines that r_j is playing defensively). Thus, we define a separate intent filter $I^i_{j,\mu}$ for every mode μ, each with its own distribution over the intent templates. By creating such a decomposition, intent inference becomes analogous to a probabilistic game between the two agents, where r_j selects a behavioural mode and action, r_i independently selects an action, and r_j subsequently attempts to synthesise its observations to infer a behavioural profile for r_i.

3.2.5 Strategic escape

The modes and intent templates defined above can be extended to include strategies that *exploit* the observability constraints that characterise multi-robot physical games. We call this class of behaviours **escape strategies**, as they strive to reduce the amount of information available to an adversary by moving objects out of their sensing range. To support the selection of such strategies, we first compute the *observability bounds* of r_i with respect to r_j as the set:

$$\mathcal{OB}^i_j \equiv \{\, vbs^i_j, sbs^i_j \,\} \leftarrow \textsc{ObservBounds}(\mathcal{RB}^i_{j,t}, \mathcal{BB}^i_{j,t}) \tag{3.14}$$

where vbs^i_j and sbs^i_j are trapezoidal approximations to the vision and sonar sensing ranges of r_i, respectively; furthermore, let $\overline{vbs^i_j}$ and $\overline{sbs^i_j}$ be their corresponding barycentres.

In our problem domain, two examples of strategic escape actions would be:

- Kick the ball so that the resulting trajectory maximises the distance from the adversary's field of view. Given a set of m candidate ball trajectories (of varying sizes), $\mathcal{BT} \doteq \{\, \beta_m \equiv \{\beta_{mk} \equiv \langle x^b_{mk}, y^b_{mk} \rangle \mid k = 1..|\beta_m|\} \,\}$, the optimal ball escape trajectory $\widehat{\beta}$ is given by:

$$\widehat{\beta} = \underset{\beta_m \in \mathcal{BT}}{\operatorname{argmax}} \frac{1}{|\beta_m|} \sum_{k=1}^{|\beta_m|} \texttt{dist}(\beta_{mk}, \overline{vbs^i_j}) \tag{3.15}$$

- Move so that the resulting path trajectory maximises the distance from the adversary's sonar sensing range. As above, given a set of n candidate robot trajectories $\mathcal{RT}$, the optimal robot escape trajectory $\widehat{\rho}$ is:

$$\widehat{\rho} = \underset{\rho_n \in \mathcal{RT}}{\operatorname{argmax}} \frac{1}{|\rho_n|} \sum_{k=1}^{|\rho_n|} \texttt{dist}(\rho_{nk}, \overline{sbs^i_j}) \tag{3.16}$$

Finally, (3.10) and (3.13) can be augmented to become:

$$\mathbf{M}^+ = \{\textsc{Kicker}, \textsc{Defender}, \textsc{Exploiter}\}, \tag{3.17}$$

$$\mathbf{I}^+ = \{\textsc{Static}, \textsc{Ball}, \textsc{Pursue}, \textsc{PredMove}, \textsc{Escape}\}. \tag{3.18}$$

An EXPLOITER is endowed with the additional capability of probabilistically selecting escape trajectories. This is modeled as an additional ESCAPE template, which represents the utility of choosing an escape strategy with respect to each adversary. This feature is then incorporated into the overall optimal action selection procedure (Algorithm 2).

3.2.6 Regret minimisation

Most of the components described so far operate on various distributions; however, only the weights of the particle filter are updated over time. We now consider *online learning* of the intent filter weights as a means of adapting to the adversary (Algorithm 3). At time t, each intent template I_k predicts a move im_k (Eq. 3.12); however, a robot probabilistically picks just one template and acts based on its prediction. Then, at $t+1$, regret minimisation assesses the correctness of *all* predictions, and weights are modified accordingly.

Algorithm 2 Optimal Action Selection

OPTACTION$(j, \mathcal{RB}_{j,t}, \mathcal{BB}_{j,t}, \alpha\tau_t, \mu_t, I_j)$
Input: Robot j, robot/ball beliefs $\mathcal{RB}_{j,t}/\mathcal{BB}_{j,t}$, action type $\alpha\tau_t$, strategic mode μ_t, current intent filters I_j
$i \leftarrow$ CHOOSEADVERSARY {find nearest adversary r_i}
$\langle I^i, im^i \rangle \sim I^i_j$ {sample template and predicted move of r_i from intent filter I^i_j}
$\mathcal{RB}^i_{j,t} \leftarrow \mathcal{RB}^i_{j,t} + im^i$ {incorporate prediction}
$\mathcal{OB}^i_{j,t} \leftarrow$ OBSERVBOUNDS$(\mathcal{RB}^i_{j,t}, \mathcal{BB}_{j,t})$ {Eq. 3.14)}
if $\alpha\tau_t ==$ MOVE$(\cdot,\cdot,\cdot)$ **then**
    if $I^i ==$ ESCAPE **then**
        $\mathcal{RT} \leftarrow$ ESCAPERT$(\mathcal{RB}^i_{j,t})$ {find candidate escape trajectories for current belief}
        $\widehat{\rho} =$ OPTRESCAPE$(\mathcal{RT}, \mathcal{OB}^i_{j,t})$ {(Eq. 3.16)}
    else
        $\mathcal{RT} \leftarrow$ NORMTRAJ$(\mathcal{RB}^i_{j,t})$
        $\widehat{\rho} =$ OPTRNORM$(\mathcal{RT})$ {no observability}
    end if
    $\alpha_t =$ MOVE$(dx, dy, d\theta) \leftarrow$ CHOOSEMOVE$(\widehat{\rho})$ {find appropriate path/move for chosen trajectory}
else if $\alpha\tau_t ==$ KICK$(\cdot,\cdot)$ **then**
    if $I^i ==$ ESCAPE **then**
        $\mathcal{BT} \leftarrow$ ESCAPEBT$(\mathcal{BB}_{j,t})$ {find candidate escape trajectories for current ball belief}
        $\widehat{\beta} =$ OPTBESCAPE$(\mathcal{BT}, \mathcal{OB}^i_{j,t})$ {(Eq. 3.15)}
    else
        $\mathcal{BT} \leftarrow$ NORMTRAJ$(\mathcal{BB}_{j,t})$
        $\widehat{\beta} =$ OPTBNORM$(\mathcal{BT})$ {no observability}
    end if
    $\alpha_t =$ KICK$(kt, ks) \leftarrow$ CHOOSEKICK$(\widehat{\beta})$ {find kick type and speed for chosen trajectory}
else
    $\alpha_t =$ SCAN$(dy, dp) \leftarrow$ CHOOSESCAN$(\mathcal{BB}_{j,t})$ {ball not visible, retrack}
end if
return α_t

Algorithm 3 Intent Regret Minimisation

1: **REGMIN**($\mathbf{I}$,t,μ_{t-1},j)
2: **Input:** Intent templates $\mathbf{I}$, time t, strategic mode μ_{t-1} at time $t-1$, estimating robot index j
3: **Auxiliary methods:** `sort`(A) {sort array A in ascending order}, `normaliseWeights` {Normalise adjusted distributions so that weights add to 1}
4: $\mu \leftarrow \mu_{t-1}, \quad \varepsilon \leftarrow 0.05$
5: **for** $i = 1$ to N ; $i \neq j$ **do**
6: $WA = \{\, \varepsilon - 2(k-1)\varepsilon/(|\mathbf{I}|-1) \quad | \quad k = 1..|\mathbf{I}| \,\}$ {weight adjustments, $+\varepsilon ... -\varepsilon$}
7: $Rs \leftarrow \emptyset$ {regrets}
8: **for** $k = 1$ to $|\mathbf{I}|$ **do**
9: $\langle I, im, iw \rangle \leftarrow I^i_{j,\mu}[k]$
10: $PP \leftarrow im + \langle x^i_j, y^i_j \rangle_{t-1}$ {predicted position}
11: $Rs[k] \leftarrow \texttt{dist}(PP, \langle x^i_j, y^i_j \rangle_t)$ {regret $\propto$ |predicted position - actual position|}
12: **end for**
13: $Rs \leftarrow \texttt{sort}(Rs)$
14: **for** $k = 1$ to $|\mathbf{I}|$ **do**
15: $I^i_{j,\mu}[Rs[k]].iw \leftarrow I^i_{j,\mu}[Rs[k]].iw + WA[k]$
16: **end for**
17: **end for**
18: `normaliseWeights` {ensure weights sum to 1 after updates}
19: **return** $I_{j,\mu}$

Algorithm 4 Complete Decision Making Algorithm

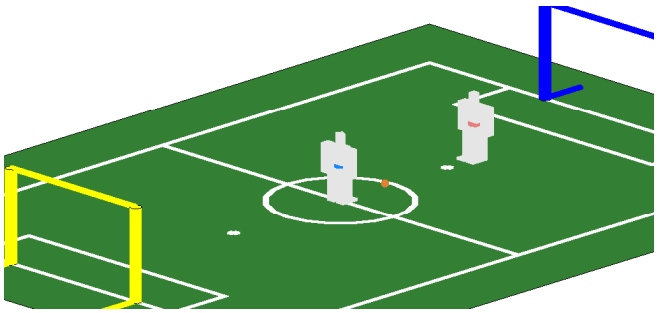
DECMAKER(I,M,rm,$I\mathcal{W}$,j)
Input: Intent templates **I**, strategic modes **M**, boolean rm for regret minimisation, initial distributions for intent template weights $I\mathcal{W}$, estimating robot index j
$t \leftarrow 0$
$I_j \leftarrow$ INITIALISEIFS(**I**, $I\mathcal{W}$) {Initialise intent filters}
while TRUE **do**
    SENSEWORLD {get latest sensor data}
    $\langle \mathcal{RB}_{j,t}, \mathcal{RP}_j, \mathcal{RM}_j \rangle \leftarrow$ RSPF {c.f. Sec. 3.2.2}
    $\langle \mathcal{RB}_{j,t}, \mathcal{BB}_{j,t} \rangle \leftarrow$ FILTERBELIEFS {keep only high confidence beliefs for decision making}
    if $rm ==$ TRUE **and** $t > 0$ **then**
        $I_{j,\mu_{t-1}} \leftarrow$ REGMIN(**I**,ld,t,μ_{t-1},j) {Alg. 3}
    end if
    $\langle \alpha\tau_t, \mu_t \rangle \leftarrow$ SELECTACTTMODE($\mathcal{RB}_{j,t}, \mathcal{BB}_{j,t}$)
    $\mathbb{I}_t \sim \{\langle I_k, im_k, iw_k \rangle \leftarrow I_{j,\mu_t}[k] \mid k = 1..|\mathbf{I}|\}$ {Select intent filters based on current weights iw_k}
    $\alpha_t \leftarrow$ OPTACTION($j, \mathcal{RB}_{j,t}, \mathcal{BB}_{j,t}, \alpha\tau_t, \mu_t, \mathbb{I}_t$)
    EXECUTEACTION(α_t)
    $t \leftarrow t+1$
end while

3.2.7 Summary

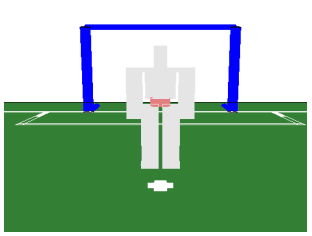
Algorithm 4 summarises the overall decision making procedure, unifying all components and ideas described so far.

3.3 Results

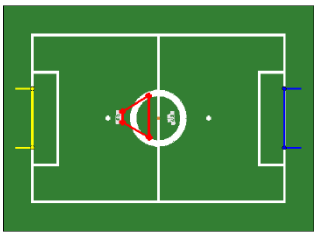
We evaluate our probabilistic state estimation on a simulated robotic soccer environment with realistic physical constraints (Valtazanos and Ramamoorthy, 2011b). Figure 3.1 shows a panoramic view of the soccer field, along with the associated field of view and sonar range representations. MOVE and KICK commands are also perturbed by random noise, with additional constraints imposed on their maximum allowed magnitudes.



(a) Panoramic view of the field and the robots.



(b) Perspective field of view



(c) Sonar sensing range

Figure 3.1: Soccer simulator environment.

3.3.1 Reachable Set Particle Filter

We first evaluate the Reachable Set Particle Filter (RSPF) proposed in Section 3.2.2, and compare it against a number of other state estimation variants, in the context of determining the state of an adversary. These variants are:

- **No filtering (NF)**: the extracted (noisy) observations from vision and sonar are converted directly to beliefs, without any probabilistic motion model or observation distributions taken into account.

- **Simple Particle Filter (SPF)**: This is a particle filter algorithm without the additional reachable set constraint. In other words, Equation 3.6 is modified so that all candidate particles are assigned a probability of $1/Q$.

- **Intent-based state estimation (IBSE)**: This procedure attempts to estimate both a robot's state and intent *in one pass*. The probabilistic motion model of Section 3.2.2, which was based on the pre-computed reachable set and the collected moves, is replaced with an intention-based distribution similar to Equation 3.12. The algorithm attempts to map intents to adversary observations directly, without *explicitly* taking into account the motion model for the adversary or the likelihood of the observations.

We evaluate the different algorithms on a series of soccer games between two robots; the initial configuration is shown in Figure 3.4. Robots use their sonar sensors to estimate the state of their adversaries, so the observation likelihood distribution accounts for the angle of the sensor cone and the maximum sensing range. Agents execute a simple algorithm that makes them move towards the ball, though their exact strategy is not of interest at this point. As in previous sections, we consider the case of r_j estimating the state of adversary r_i. For each filtering algorithm $\mathbf{f}$, we compute the error in terms of the mean distance of r_j's egocentric estimates, $\mathbf{x}_{j,t}^{i,\mathbf{f}}$, to the true location of r_i, $\bar{\mathbf{x}}_t^i$:

$$\mathbf{MDTL}(\mathbf{f}) = \frac{1}{T}\sum_{t=1}^{T}\sqrt{\mathbf{x}_{j,t}^{i,\mathbf{f}} - \bar{\mathbf{x}}_t^i}^2 . \tag{3.19}$$

For the RSPF algorithm, the backward reachable set $\mathcal{BRS}$ (Section 3.2.2) was computed with respect to the relative dynamics of the two robots (Figure 3.2), using the level set toolbox developed by Mitchell (2007). Figure 3.3 visualises the evolution of the particle filter and one-step reaction distributions over time, illustrating the utility of the reachable set as a filtering tool.

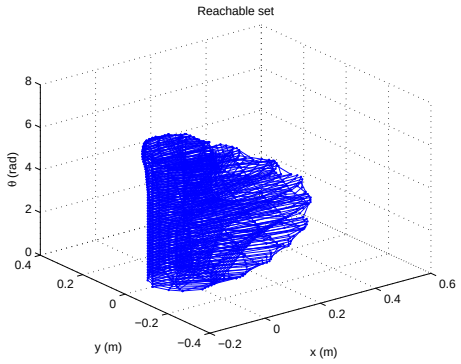
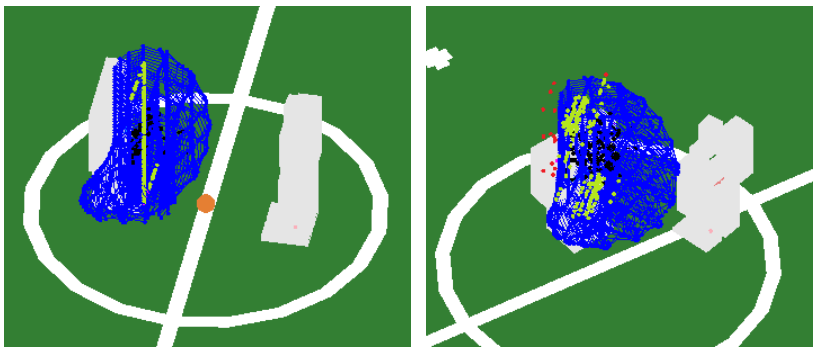


Figure 3.2: Three dimensional (x,y,θ) reachable set computed for a time horizon of 1s based on the relative dynamics of the two robots – maximum linear velocity = 0.2m/s, maximum angular velocity = 0.2rad/s.

Table 3.1 summarises the results, as averaged over 20 trials. At a first glance, the 6.96% gain obtained when using RSPF instead of no filtering (NF) may seem small, but one must acknowledge the complexity of the task: robots must estimate the state of dynamic adversaries whose exact behavioural and motion model is unknown, using only noisy sensor data. More importantly, any improvement at this level – in rejecting spurious trajectories – has a substantial impact on the following steps that attempt



(a) Initial distributions. (b) Update after 15 time steps.

Figure 3.3: Reachable sets and particle filter distributions. *Blue*: reachable set (Figure 3.2). *Black*: state particles $\mathcal{RP}$. *Light green*: one-step reaction distribution $\mathcal{RM}$. *Red*: discarded particles lying outside the reachable set.

Filtering method (f)	**MDTL(f)**	**Error gain wrt. NF**
NF	14.79 cm	-
SPF	17.63 cm	-19.2%
RSPF	13.76 cm	+6.96%
IBSE	15.16 cm	-2.5%

Table 3.1: Mean error per filtering method

to learn strategic profiles and responses on top of this information. Thus, it is very useful that the RSPF succeeds in rejecting movement observations that are physically implausible, as seen by its performance compared to the simple particle filter (SPF), an effective 25% gain. Moreover, the performance of RSPF relative to IBSE supports our claim that in games characterised by both strategic and sensory noise, state estimation should be decoupled from strategy estimation. As we demonstrate in the next section, our decision making algorithm benefits from reasoning on data that has been filtered by the RSPF.

3.3.2 Strategic decision making

3.3.2.1 Preliminaries

In the second part of our experiments, we fix the RSPF as the state estimation algorithm, and focus on evaluating different permutations of decision making strategies, based on the concepts described in Section 3.2. Each decision making strategy is a

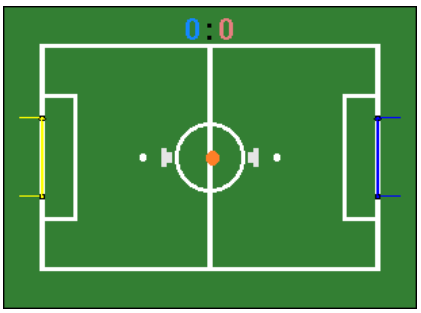


Figure 3.4: Soccer game initial configuration - the ball is at the centre of the pitch, with the two robots symmetrically placed just behind the circle.

tuple $\langle \mathbf{e}, \mathbf{r} \rangle$, where:

- $\mathbf{e} \in \{(N)\text{one}, (E)\text{scape}\}$ denotes the escape strategy used. In other words, $\mathbf{e} \leftarrow N$ uses $\mathbf{M}^-$ and I^- (Equations 3.10-3.13), whereas $\mathbf{e} \leftarrow E$ uses $\mathbf{M}^+$ and I^+ (Equations 3.17-3.18) as their strategic modes and intent templates, respectively,
- $\mathbf{r} \in \{(N)\text{one}, (I)\text{ntent regret minimisation}\}$ denotes the type of regret minimisation used - this is the input parameter *rm* in Algorithm 4.

This formulation leads to a total of 4 valid permutations, namely $\langle N,N \rangle$, $\langle N,I \rangle$, $\langle E,N \rangle$ and $\langle E,I \rangle$.

We compare these strategies in the context of a round-robin one-versus-one soccer tournament, where all strategies are played in four games against each other. The first two games (**10-1** and **10-2**) consist of 10 *episodes*, and the second two of 20 episodes (**20-1** and **20-2**). An episode terminates if a robot scores a goal, if the ball leaves the field bounds, if there is a collision between robots, or if the maximum episode time (set to 100 time steps for each robot) elapses. As before, Figure 3.4 shows the initial state of the game. However, note the *strategic* constraints that this setup introduces:

- The ball is too far from the goals, so robots require a *sequence* of actions in order to score.
- The robots are initialised very close to each other, so they require dexterous manoeuvres to evade their opponents and kick the ball past them.

3.3.2.2 Results and statistics

In addition to the final scores of the games, we also recorded other relevant statistics, such as the mean time to score a goal **(MTS)** and to evade the adversary **(MTE)**, and

(a) Summary of scores and strategy statistics

Strategy	**P**	**GF**	**GA**	**MTS**	**MTE**	**A-MTS**	**A-MTE**
$\langle N,I\rangle$	**35**	**131**	**121**	71.86	**53.32**	**73.04**	**58.67**
$\langle E,I\rangle$	34	134	102	**67.56**	55.06	71.65	55.36
$\langle E,N\rangle$	18	40	48	71.67	58.68	66.50	51.08
$\langle N,N\rangle$	14	41	52	71.87	55.50	72.04	56.84

(b) Soccer tournament results

	10-1	**10-2**	**20-1**	**20-2**	**50-1**	**50-2**
$\langle N,N\rangle$-$\langle N,I\rangle$	3-4	2-4	6-8	4-7	14-14	10-12
$\langle N,N\rangle$-$\langle E,N\rangle$	1-4	4-4	5-4	5-5	14-16	13-12
$\langle N,N\rangle$-$\langle E,I\rangle$	1-2	1-2	5-5	4-3	12-12	15-17
$\langle N,I\rangle$-$\langle E,N\rangle$	3-0	4-1	9-5	6-4	17-10	12-15
$\langle N,I\rangle$-$\langle E,I\rangle$	4-2	3-3	4-7	4-7	7-19	9-17
$\langle E,I\rangle$-$\langle E,N\rangle$	1-2	4-1	5-5	3-5	12-11	13-9

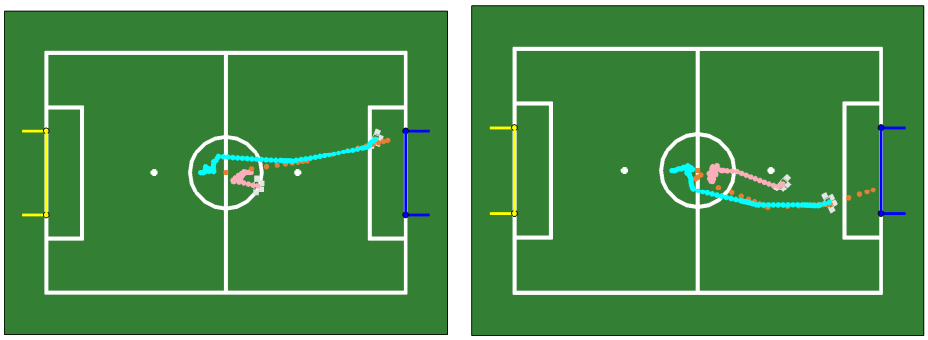
Table 3.2: Results and statistics

the mean time taken by the adversary to score **(A-MTS)** and to evade **(A-MTE)**. Table 3.2(a) summarises these statistics, together with the total goals scored for **(GF)** and against **(GA)**. The entries are sorted with respect to the total points **(P)**, which are determined through the standard soccer point system (3 points for victory, 1 for draw, 0 for defeat - maximum is 3x12 = 36). Point ties are resolved by the goal difference **(GD) = GF-GA**. The best entries in each column are given in boldface. The results from all games are summarised in Table 3.2(b).

Trajectory traces from two game episodes are given in Figures 3.5(a) (episode between a "good" and "bad" strategy) and Figure 3.5(b) (the two best strategies). In the latter case, the scoring robot takes more time to kick the ball past its adversary and subsequently escape, as indicated by the larger concentration of movements around the centre of the field.

3.3.2.3 Partial orderings

The results of Tables 3.2(a) and 3.2(b) lead to several observations on the effectiveness of the various strategies and ideas presented in this chapter. We summarise these findings in terms of **partial orderings** among templates. We use the notation $\mathcal{T}_1 \geq \mathcal{T}_2$ to denote that template $\mathcal{T}_1$ performs at least as well as template $\mathcal{T}_2$. Furthermore, let '$\cdot$'



(a) $\langle E,I\rangle$ (cyan) vs. $\langle N,N\rangle$ (pink) (b) $\langle E,I\rangle$ (cyan) vs. $\langle N,I\rangle$ (pink)

Figure 3.5: Trajectory traces from selected game episodes.

be the wildcard symbol, and $\neg A$ be any instantiation of a particular strategy or strategy template except A.

- $\langle \cdot, I\rangle \geq \langle \cdot, \neg I\rangle$: Regret minimisation seems to be by far the most prevalent strategy, both on its own and when combined with escape strategies, as indicated by both the overall scores and the statistics.

- $\langle E, \cdot\rangle \geq \neg\langle N,I\rangle$: The use of escape strategies is also highly beneficial when compared to the benchmark $\langle N,N\rangle$ that makes no assumptions about the adversary. Furthermore, strategy $\langle E,I\rangle$ achieves the highest scores in all but one of the time metrics.

- $\neg\langle N,N\rangle \geq \langle N,N\rangle$: All strategies using at least one heuristic out outperform the benchmark $\langle N,N\rangle$.

3.3.2.4 Convergence of regret minimisation

To verify that regret minimisation is indeed reliable, we tested it against a static adversary who does not move. The regret minimising robot is not aware of this, so it initialises all intent template weights uniformly. Figure 3.6 illustrates how regret minimisation helps converge to the "true" template distribution. Note that the PREDMOVE template is also representative of a static adversary's "strategy"; if that adversary is consistently observed not to move, then the distribution $\mathcal{RM}$ will assign high weights to null moves, thus also predicting a static reaction. Hence, the joint weights of STATIC and PREDMOVE (Static + PredMove) are correctly observed to converge to 1.

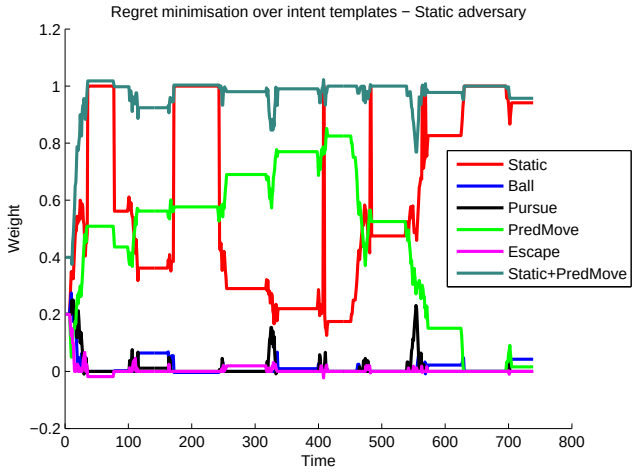


Figure 3.6: Regret minimisation over the intent templates of a static adversary. The weights of the intent filters are correctly observed to converge to the "true adversary" behaviour.

3.4 Conclusions

This chapter presents a strategic decision making framework for a relatively complex class of multi-robot games, characterised by both sensory and strategic uncertainty. The specific contributions of this chapter are twofold. On the one hand, we present a novel probabilistic adversarial state estimation algorithm, featuring both data-driven approximation and reasoning about dynamical constraints. Our evaluation shows a performance improvement, in simulation, compared to general purpose filtering algorithms, which supports our argument in favour of decoupling estimation of a noisy state from estimation of strategy in the noisy adversarial environment. On the other hand, we have adapted game theoretic concepts, which had previously been studied primarily in an abstract theoretical setting without physical constraints, into a unified intent inference framework for multi-robot games. We have tested several instantiations of our framework by evaluating their performance against varying unknown strategies. Our results favour the use of regret minimisation as an adaptive learning mechanism, while showing promise for careful use of escape strategies that exploit the adversary, as part of a larger decision making system with a diverse set of intent templates.

In relation to the interaction shaping problem considered in this thesis, this chapter introduces the use of techniques that can influence the beliefs of interacting agents, such as escape strategies. However, these strategies are handcrafted, and do not account for or adapt to different types of adversaries. In the ensuing chapters, we expand

on these concepts and situate them within a formal theoretical framework. First, we introduce a procedure through which templates for such strategies can be learned from human demonstration (Chapter 4), instead of being defined manually as in this chapter. This is a general procedure that can also be applied to other domains outside robotic soccer. Second, we formulate a Bayesian learning algorithm, through which basic strategic behaviours can be synthesised and learned (Chapter 5). This algorithm is similarly based on regret-minimisation and opponent modeling ideas, but allows for the generation of temporally extended action strategies, which are expected to lead the adversary to some target state. Thus, the strategies presented in this chapter can be essentially viewed as basic shaping behaviours, which we subsequently expand on. Third, we introduce human subjects directly into our experimental evaluation, by pitting them against various autonomously programmed robots. In these experiments, humans essentially act as intelligent, strategic adversaries, whose behaviour has not been scripted in any way. Thus, we improve on the sophistication of the adversaries presented in this chapter, and we test our models under more realistic and challenging conditions.

Chapter 4

Learning to interact with strategic agents from human demonstrations

4.1 Overview

Most compelling application scenarios for autonomous robots involve operation in environments inhabited by other decision-making agents, e.g. people. In many such applications, there is a need for non-trivial strategic interaction with these other agents. As with most forms of robot behaviour that must be adaptive, it would be of great interest to be able to *learn* to interact strategically. However, the problem of efficiently learning such strategic behaviours remains open.

In Chapter 3, we introduced a class of strategic behaviours that can be used to influence an interacting adversarial agent. One drawback of that approach was that these strategies were manually specified by the system designer, based on heuristics believed to be optimal for the experimental domain (in that case, robotic soccer). However, manual design of behaviours is often cumbersome and inflexible, while scaling poorly to different interaction domains where there may be a lack of clear, appropriate design heuristics. By contrast, most decision-making frameworks would considerably benefit from a *general, automated* strategy learning procedure, through which behaviour components can be extracted directly from provided human demonstrations.

The focus of this chapter is on learning such strategy templates that can facilitate strategic interaction with humans. In this setting, the open questions for us would be: how can robots effectively make strategic decisions that can influence and affect human behaviour during an interaction, and how can such strategies be learned from human demonstration?

With these questions in mind, we consider the robotic soccer penalty shooting problem between NAO humanoid robots, which was introduced in Section 1.3. In this domain, both types of agents have the same locomotion capabilities, so a robot cannot benefit by e.g. walking or kicking the ball faster. Thus, we can compare agents directly at the *behavioural level*, without however removing the underlying physical uncertainty that characterises most realistic human-robot environments.

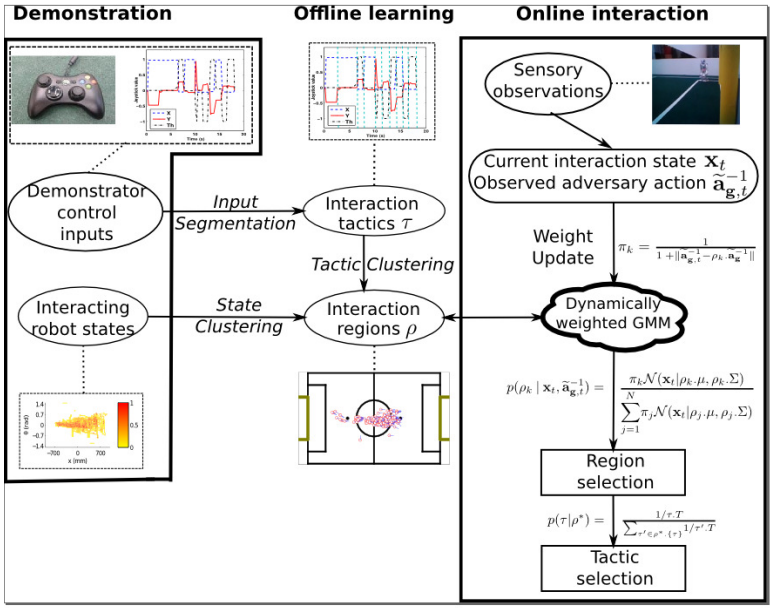


Figure 4.1: Overview of proposed approach. Human demonstrators teleoperate a robot to provide examples of the desired strategic behaviour, in the context of an interaction with a heuristic autonomous adversary. Demonstrations are organised into interaction tactics and regions, which form the basis of a Gaussian Mixture Model (GMM). The learned model is used to interact strategically with novel adversaries. The mixture weights are dynamically updated based on the observed actions of the adversary, which, combined with the current state of the interaction, informs action selection.

We present a semi-supervised learning procedure through which autonomous robots can learn mixtures of *interaction strategies* from human demonstration (Figure 4.1). We first record the control inputs of several subjects demonstrating the striker behaviour against a heuristic autonomous goalkeeper. Because of the nature of the interaction, part of the demonstrated examples may be *suboptimal* (e.g. the robot takes a lot of time to score a goal). Strategies are extracted directly from demonstrations, and represented as sequences of *tactics* that are used to transition between interaction *regions*. A region can be viewed as a group of states frequently visited by the com-

peting robots, whereas a tactic is a continual action intended to change the state of the game, which is selected in response to the observed behaviour of the adversary. Regions are represented as a dynamically weighted Gaussian Mixture Model, within which lie additional distributions for selecting tactics. Through this formulation, new strategies can be generated and synthesised probabilistically against a wide range of adversaries.

In the remainder of this chapter, we first describe the experimental setup (Section 4.2) for our problem, by illustrating the interaction rules and the capabilities of our robots. Then, we present our method for extracting strategies from human demonstration (Section 4.3). In Section 4.4, we evaluate our approach against several autonomous and human-controlled agents, and we show that the learned agent can successfully compete with different adversaries. Finally, we review the key contributions of this chapter, and describe its connection to the following chapters of this thesis (Section 4.5).

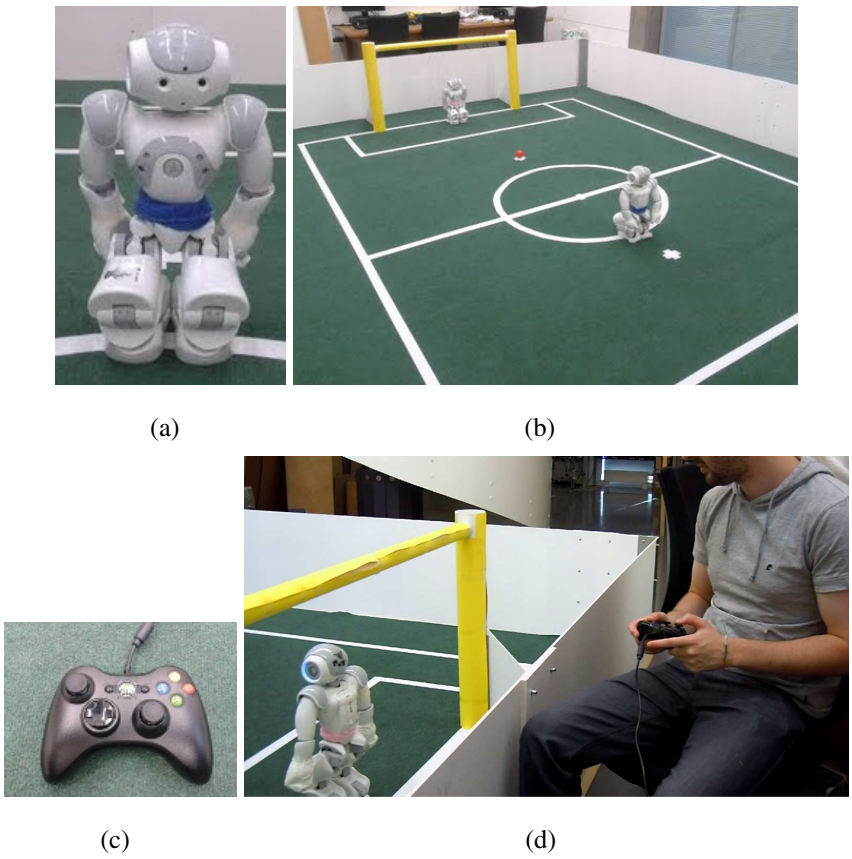
4.2 Experimental setup

4.2.1 Robot platform

We use the NAO robot (`http://www.aldebaran-robotics.com/en/Discover-NAO/nao-datasheet-h25.html`) (Figure 4.2(a)), a 58cm-tall humanoid with 21 degrees of freedom and two independently running on-board cameras. The NAO is the official robot of the RoboCup Standard Platform League (SPL) (`http://www.tzi.de/spl`). Our software framework is based on the B-Human team code release (Röfer et al., 2011), which provides modules for fast walking, vision, and self-localisation.

4.2.2 Field

Our soccer field (Figure 4.2(b)) is a 3:4 scaled-down version of the official SPL field (`http://www.tzi.de/spl/pub/Website/Downloads/Rules2012.pdf`), with length $fl = 4.5$m and width $fw = 3.0$m. The goals are 1.40m wide and are painted yellow, and the ball is also colour-coded orange. The field lines are white and are placed at known, specified positions.



(a) (b)

(c) (d)

Figure 4.2: Interaction strategy demonstration - experimental setup. (a): The NAO humanoid robot. (b): The soccer field, with an orange ball on the penalty cross mark. The initial poses of the striker (near side, blue waistband) and the goalkeeper (far side, pink waistband) are also shown. (c): The controller used to command the robot. (d): Bringing it all together: remote control of the goalkeeper by a user.

4.2.3 Self-localisation and adversary pose estimation

The robots not controlled by humans are *fully* autonomous, so no external information (e.g. positions from overhead cameras) is provided to them. The images captured by their camera are used to identify the relative positions of the ball and the relevant field landmarks. This information is passed to a self-localisation module, which computes the robot's absolute pose (position and orientation) using a particle filter.

Egocentric estimation of the pose of other robots is a challenging task for autonomous NAOs, because of the limited number of distinguishable visual features. To overcome this problem, we let robots wirelessly communicate their pose estimate to their adversaries. One drawback of this approach is that any delays or packet losses in the network will yield outdated information on the state of the adversary.

4.2.4 Comparison between human and robot perception

The self-localisation and pose estimation techniques (Röfer et al., 2011) used by the autonomous robots are state-of-the-art and have been successfully tested in international competitions, e.g. RoboCup 2012 (`http://www.robocup2012.org`). However, they occasionally lead to false estimates, e.g. when a penalty box line is mistaken for a field line. This issue, coupled with the wireless connectivity discussed above, leads to uncertainty in the estimation of the current state of the game. By contrast, humans have the benefit of *full observability* of both the field and the robots during the interaction. This perceptual handicap should be taken into consideration when comparing the behaviour of the autonomous and the human-controlled agents.

4.2.5 Interaction rules

Our interaction follows the official rules of the SPL penalty shootout (`http://www.tzi.de/spl/pub/Website/Downloads/Rules2012.pdf`). The robots are initially placed on the positions shown in Figure 4.2(b). The striker has one minute to score a goal and is allowed only a single kick per trial. The goalkeeper is not allowed to leave or touch the ball outside the penalty box; any such violation results to a goal awarded to the striker. Strikers have a single, straight left kick they may execute; thus, to shoot towards the corners of the goal, they must adjust their orientation to face towards that corner.

4.2.6 Human control of the robots

Teleoperated robots are controlled through an Xbox pad (Figure 4.2(c)). There are commands for controlling the translational (forward-back-sidestepping) and rotational (left-right turn) motion of the robot, kicking (for the striker), and "diving" to block the ball (for the goalkeeper).

4.3 Method

4.3.1 System formulation and notation

We consider a continuous-time, continuous-state, and continuous-action system of two interacting robots. The state of the system at time $t \in \Re^+$ is described by:

- The state of the striker, $\mathbf{s}_t = [x_s, y_s, \theta_s]^T$, and the goalkeeper, $\mathbf{g}_t = [x_g, y_g, \theta_g]^T$, where $\{x_s, x_g\} \in [-fl/2, +fl/2]$, $\{y_s, y_g\} \in [-fw/2, +fw/2]$ are the planar coordinates, and $\{\theta_s, \theta_g\} \in [-\pi, +\pi]$ are the orientations of the two robots.

- The ball position, $\mathbf{b}_t = [x_b, y_b]^T$, $x_b \in [-fl/2, +fl/2]$, $y_b \in [-fw/2, +fw/2]$.

- The actions $\vec{\mathbf{a}}_{\mathbf{s},t}$, $\vec{\mathbf{a}}_{\mathbf{g},t}$ requested and executed by the two players – each action $\vec{\mathbf{a}}$ is a tuple $[dx, dy, d\theta, kick, dive]^T$, where $\{dx, dy, d\theta\} \in [-1.0, +1.0]$ are the requested translation and rotation as fractions of the maximum speed of the robot[1] (so $\{0,0,0\}$ for no motion), and $kick \in \{none, kick\}$, $dive \in \{left, none, right\}$ are the kicking (for the striker) and diving (for the goalkeeper) requests.

For teleoperated robots, joystick commands and button presses are converted to action requests (e.g. pushing the left joystick forward issues a request of $dx = +1.0$).

For the remainder of this chapter, we will use the superscripts ^{H} and ^{A} to denote human-controlled and autonomous robots, respectively. For example, $\mathbf{s}_t^H$ is the state of a human-controlled striker at time t.

4.3.2 Autonomous goalkeeper behaviour during demonstrations

During demonstration of striker behaviours, the goalkeeper runs a simple heuristic algorithm: given the current estimate of the striker's orientation, θ_s, the expected ball trajectory is a straight line segment starting at the ball position, and following this angle. Then, the point where this segment intersects the goal line is selected as the best blocking position, so the goalkeeper moves to this point. The goalkeeper may also dive to prevent a goal from being scored if he visually detects the ball moving towards his own axis of motion.

We refer to this player as the **heuristic autonomous goalkeeper (HAG)**. This algorithm also serves as a simple baseline that can be contrasted with autonomous strikers, not only in the results of this chapter, but also in Chapter 5 where the main shaping algorithm is presented.

4.3.3 Human behaviour demonstration

We captured the control inputs of several users controlling the striker against the autonomous goalkeeper. Data were collected in our robotics lab (Figure 4.2(b)), and

[1] The positive directions are forward, left, and counter-clockwise rotation.

the demonstrators were members of our research group who had prior experience of the robots. For each trial, we recorded time-indexed sequences of the robot poses, as well as the control commands input by the users, at a rate of 10 frames/second. Each recorded sequence $\mathbf{q}$, $|\mathbf{q}| = M$, is of the form:

$$\{\{t_1, \mathbf{s}^H_{t_1}, \mathbf{g}^A_{t_1}, \mathbf{b}_{t_1}, \vec{\mathbf{a}}^H_{\mathbf{s},t_1}\}, ..., \{t_M, \mathbf{s}^H_{t_M}, \mathbf{g}^A_{t_M}, \mathbf{b}_{t_M}, \vec{\mathbf{a}}^H_{\mathbf{s},t_M}\}\} \tag{4.1}$$

where $\vec{\mathbf{a}}^H_{\mathbf{s},t}$ corresponds to an action command by the user. Trials were also annotated by the experimenter based on their outcome, i.e. whether they led to a goal or not. We then extracted the set of successful trials leading to a goal:

$$\mathbf{Q}^+ = \{\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_N\}. \tag{4.2}$$

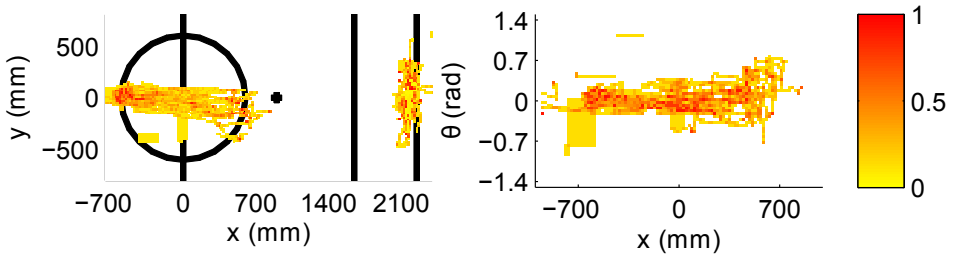
However, even if a demonstrated example is labelled as *successful*, it is not necessarily *optimal*. Several of the collected trials were ***suboptimal*** (e.g. the user took a long time to score a goal), ***imperfect*** (e.g. at least one player converged to an incorrect self-localisation estimate), or ***both*** (e.g. the goalkeeper stumbled and fell over, leaving an open goal for the demonstrator to score). Nevertheless, we did not discard such demonstrations from our learning procedure.

In total, 29 successful demonstrated trials were retained. Figure 4.4 shows two such demonstrations, one of which is successful. Furthermore, Figure 4.3 provides heat maps of all demonstrations from the lab capture. It can be seen that although the two sets of trajectories are similar, successful demonstrations are characterised by a higher intensity of motion (and particularly rotational motion) around the penalty mark, which indicates an attempt to perform finer adjustments of the striker's pose and deceive the autonomous goalkeeper. By contrast, simpler strategies such as walking directly to the ball and kicking (as in the first example of Figure 4.4) are less likely to outperform the goalkeeper and score a goal.

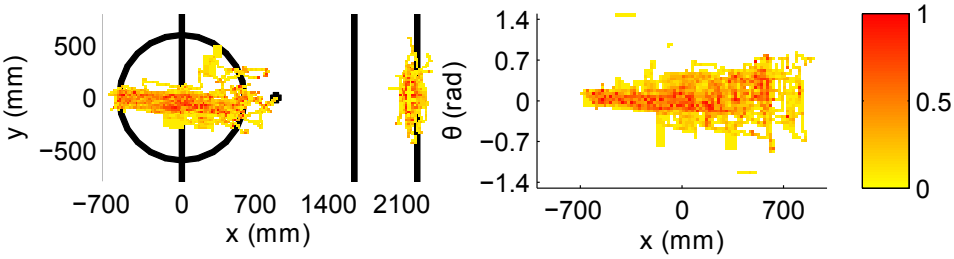
4.3.4 Learning strategy mixtures

4.3.4.1 Definitions

We represent striker strategies for the autonomous striker as sequences of **tactics** used to transition between different **interaction regions**. A tactic τ is an action continually evoked by a robot for a variable period of time, in response to the observed behaviour of the adversary:



(a) Unsuccessful demonstrations.



(b) Successful demonstrations.

Figure 4.3: Heat map representations of demonstrations. Colour indicates the percentage of trials in which a particular point was recorded. *Left subplots:* $x-y$ motion trajectory components - left blob corresponds to human-controlled striker, right blob to autonomous goalkeeper trajectories. *Right subplots:* $x-\theta$ motion for the striker.

$$\tau = \langle \check{\mathbf{s}}, \check{\mathbf{g}}, \vec{\mathbf{a}}_{\mathbf{s}}, dt, T, \widetilde{\mathbf{a}}_{\mathbf{g}}^{-1} \rangle, \tag{4.3}$$

where $\check{\mathbf{s}}, \check{\mathbf{g}}$, are the states of the players at the time the tactic is invoked, $\vec{\mathbf{a}}_{\mathbf{s}}$ is the action followed by the striker, dt is the time interval for which this action should be taken, T is the overall time of the trial from which τ was originally extracted, and $\widetilde{\mathbf{a}}_{\mathbf{g}}^{-1}$ is the action believed to have been followed by the goalkeeper at the time interval preceding τ.

An interaction region ρ is a distribution over related states frequently visited by the two robots during the interaction:

$$\rho = \langle \mu, \Sigma, \widetilde{\mathbf{a}}_{\mathbf{g}}^{-1}, \{\tau\} \rangle, \tag{4.4}$$

where $\mu = [\bar{\mathbf{s}}\,; \bar{\mathbf{g}}]$ is the mean of the region, represented as a joint striker-goalkeeper state vector, Σ is the covariance matrix, $\widetilde{\mathbf{a}}_{\mathbf{g}}^{-1}$ is the adversarial action associated with the region, and $\{\tau\}$ is the set of tactics that can be invoked from the region. Through this formulation, the interaction becomes a dynamic game between the two robots, where the striker must select the appropriate tactics in response to the state and inferred

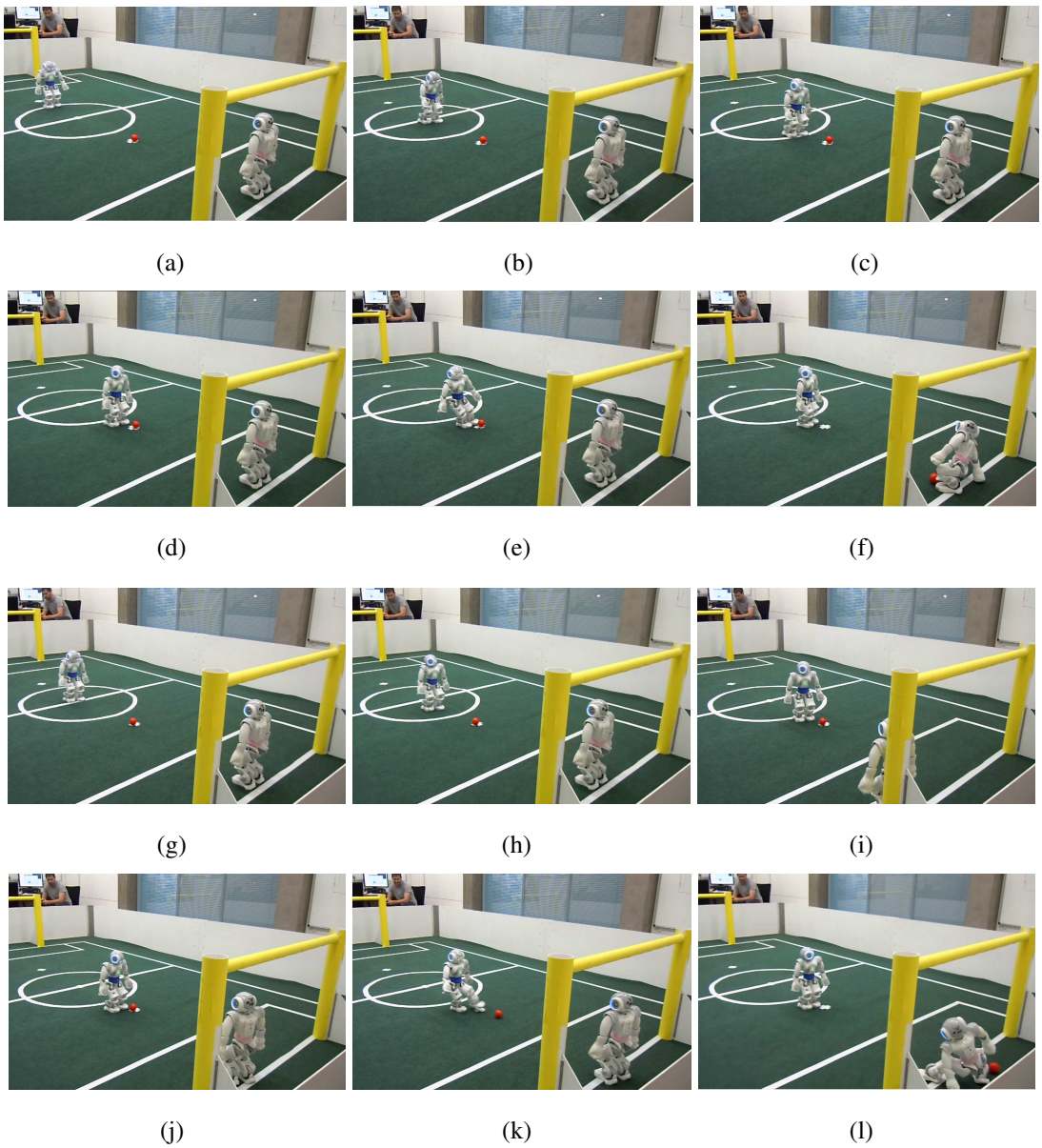


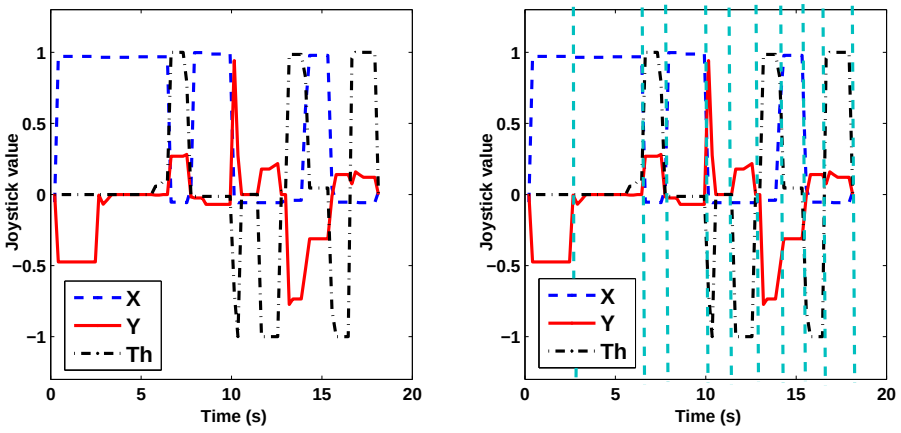
Figure 4.4: Examples of two demonstrations by a user (human-controlled striker, autonomous goalkeeper). (a)-(f): Unsuccessful trial. (g)-(l): Successful trial.

actions of its adversary.

4.3.4.2 Extracting tactics from human demonstration

Given the set $\mathbf{Q}^+$ of successful demonstrations, we extract tactics from the recorded commands, $\{\vec{\mathbf{a}}^H_{\mathbf{s},t_k} \mid k = 1..N\}$, of each demonstration $\mathbf{q}$, where $|\mathbf{q}| = N$. Figure 4.5(a) shows the raw input commands for the rotation and translation axes from one such trial. To account for possible joystick miscalibrations, we define the *activity thresholds*

$\psi = \pm 0.4$ for each motion axis; values with magnitude less than ψ are discarded as noise. The incorporation of this threshold is important as the input device is very sensitive to minor pressure on the joysticks, which however do not represent intentional motion commands. We process inputs as joint continuous signals to segment them into tactics.



(a) (b)

Figure 4.5: Control inputs and tactics from a successful demonstration. (a): The raw inputs from the translational and rotational joystick axes. (b): Segmentation of the input into tactics; a new tactic begins when either of the inputs crosses the activity threshold of ± 0.4 (not shown). The cyan vertical lines illustrate the boundaries between consecutive tactics.

A new tactic τ begins when at least one of the motion inputs, $\alpha^H \in \{dx, dy, d\theta\}$, crosses either $+\psi$ or $-\psi$, i.e. if

$$\vec{\mathbf{a}}^H_{t_k} > |\psi| > \vec{\mathbf{a}}^H_{t_{k-1}} \textbf{ or } \vec{\mathbf{a}}^H_{t_k} < |\psi| < \vec{\mathbf{a}}^H_{t_{k-1}}. \tag{4.5}$$

A tactic is in progress while all α^H remain on their current side of $|\psi|$. Figure 4.5(b) illustrates how the raw input is segmented into tactics using this heuristic. As successful trials always end with a "kick" command, we append an additional such tactic to our set[2].

For every extracted tactic τ, we record the start and end times at its boundaries, t_s and t_e. Then, the tactic time interval and input states (as defined in Section 4.3.4.1) are defined as

[2]In effect, the kick button can be viewed as an additional continuous input whose value is always 0, except for the end of the trial when it is 1.

$$\tau.dt \leftarrow t_e - t_s,\ \tau.\check{\mathbf{s}} \leftarrow \mathbf{s}^H_{t_s},\ \tau.\check{\mathbf{g}} \leftarrow \mathbf{g}^A_{t_s}. \tag{4.6}$$

The parameter $\tau.T$ is the overall duration of $\mathbf{q}$. The tactic action vector, $\vec{\mathbf{a}}_\mathbf{s}$, is the mean of the motion inputs (and *none* for kicks/dives) over the duration of τ,

$$\tau.\vec{\mathbf{a}}_\mathbf{s} \ \leftarrow\ [\,\frac{1}{t_e - t_s}\sum_{t=t_s}^{t_e}\alpha^H_t \,|\, \alpha^H \in \{dx, dy, d\theta\}\,]. \tag{4.7}$$

The last adversarial action, $\tau.\widetilde{\mathbf{a}}^{-1}_\mathbf{g}$, is obtained by evaluating the goalkeeper's motion during the previous tactic. For example, if during that interval a leftward translation was observed but no forward or rotational motion, we set $\widetilde{\mathbf{a}}^{-1}_\mathbf{g} = [0.0, 1.0, 0.0, \mathit{none}, \mathit{none}]^T$. We assume no motion at the start of the trial, so, for the first tactic, $\widetilde{\mathbf{a}}^{-1}_\mathbf{g} = [0.0, 0.0, 0.0, \mathit{none}, \mathit{none}]^T$.

4.3.4.3 Region computation

Given a set of extracted tactics, regions are generated in a two-step clustering process. In the first step, we form a set of tactic groups $TG = \{\tau g_1, \tau g_2, ..., \tau g_M\}$ based on input state similarity, so that any two tactics τ_i, τ_j within a tactic group $\tau g \in TG$ satisfy

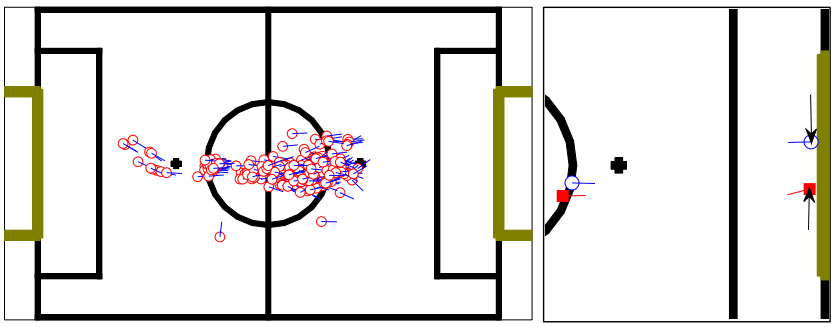
$$d([\tau_i.\check{\mathbf{s}}; \tau_i.\check{\mathbf{g}}], [\tau_j.\check{\mathbf{s}}; \tau_j.\check{\mathbf{g}}]) < \delta, \tag{4.8}$$

where $d(\cdot,\cdot)$ is the distance between two state pairs, and δ is a distance threshold. Then, tactics within the same group τg but with different adversarial actions are separated, to obtain a new set of tactic groups TG', where any two tactics τ'_i, τ'_j in a group $\tau g'$ also satisfy $\tau'_i.\widetilde{\mathbf{a}}^{-1}_\mathbf{g} = \tau'_j.\widetilde{\mathbf{a}}^{-1}_\mathbf{g}$.

Each resulting group $\tau g' \in TG'$ is converted to a new region ρ, with adversarial action $\rho.\widetilde{\mathbf{a}}^{-1}_\mathbf{g} = \tau.\widetilde{\mathbf{a}}^{-1}_\mathbf{g}$, and tactic set $\rho.\{\tau\} = \tau g'$. The parameters μ and Σ are the sample mean and covariance of the input states of all tactics in $\rho.\{\tau\}$.

The collected demonstrations yielded a total of 236 interaction regions, which are shown in Figure 4.6.

The aim of the region computation procedure is twofold. On the one hand, we group tactics with similar input states, so that the robot has a choice of actions when selecting that region. On the other hand, we separate tactics with different associated adversarial actions, so that the choice of region can be biased by the observed behaviour of the adversary.



(a) (b)

Figure 4.6: Means of interaction regions for a striker shooting towards the right goal, as computed from successful human demonstrations. (a): All regions - *only* the striker position (red circle) and orientation (blue line) component of the region mean is shown. (b): Full visualisation of the means of two regions with similar striker states - note the difference between the corresponding goalkeeper states and adversarial actions (indicated by the black arrows).

4.3.5 Strategic interaction with novel adversarial agents

Interaction with an adversarial agent is formulated as a sequential hierarchical region and tactic selection problem. The striker first selects an interaction region, and then samples a tactic τ from the tactic set $\{\tau\}$ of that region, which should be followed for $\tau.dt$. This process is repeated at the completion of the currently executed tactic.

4.3.5.1 Region selection

Given a set of N interaction regions, $R = \{\rho_1, \rho_2, \cdots, \rho_N\}$, we use their means and covariances to obtain a set of N multivariate Gaussian distributions,

$$G = \{\mathcal{N}(\mathbf{x} \,|\, \rho_1.\mu, \rho_1.\Sigma), \cdots, \mathcal{N}(\mathbf{x} \,|\, \rho_N.\mu, \rho_N.\Sigma)\}. \tag{4.9}$$

where $\mathbf{x} = [\mathbf{s}_t^A; \mathbf{g}_t^H]$ represents the current states of the players. The set G forms the basis of a dynamically weighted Gaussian Mixture Model (GMM), whose mixing weights are re-computed based on the current state of the game. Given the current estimate of the adversary's last followed action, $\widetilde{\mathbf{a}}_{\mathbf{g},t}^{-1}$, the weight of the k-th distribution, π_k, is given by

$$\pi_k = p(\rho_k.\widetilde{\mathbf{a}}_{\mathbf{g}}^{-1} \,|\, \widetilde{\mathbf{a}}_{\mathbf{g},t}^{-1}) = \frac{1}{1 + \parallel \widetilde{\mathbf{a}}_{\mathbf{g},t}^{-1} - \rho_k.\widetilde{\mathbf{a}}_{\mathbf{g}}^{-1} \parallel} = \frac{1}{1 + \sum_{i=1}^{3} |\widetilde{\mathbf{a}}_{\mathbf{g},t}^{-1}[i] - \rho_k.\widetilde{\mathbf{a}}_{\mathbf{g}}^{-1}[i]|}. \tag{4.10}$$

In other words, π_k reflects the similarity between $\widetilde{\mathbf{a}}_{\mathbf{g},t}^{-1}$ and the adversarial action of the k-th interaction region.

If $\mathbf{x}_t = [\mathbf{s}_t; \mathbf{g}_t]$ is the current estimate of the player states, the probability of selecting region ρ_k at time t is

$$\gamma(\rho_k) \equiv p(\rho_k \,|\, \mathbf{x}_t, \widetilde{\mathbf{a}}_{\mathbf{g},t}^{-1}) = \frac{p(\rho_k.\widetilde{\mathbf{a}}_{\mathbf{g}}^{-1} \,|\, \widetilde{\mathbf{a}}_{\mathbf{g},t}^{-1}) p(\mathbf{x}_t|\rho_k)}{\sum_{j=1}^{N} p(\rho_j.\widetilde{\mathbf{a}}_{\mathbf{g}}^{-1} \,|\, \widetilde{\mathbf{a}}_{\mathbf{g},t}^{-1}) p(\mathbf{x}_t|\rho_j)} = \frac{\pi_k p(\mathbf{x}_t|\rho_k)}{\sum_{j=1}^{N} \pi_j p(\mathbf{x}_t|\rho_j)} = \frac{\pi_k \mathcal{N}(\mathbf{x}_t|\rho_k.\mu, \rho_k.\Sigma)}{\sum_{j=1}^{N} \pi_j \mathcal{N}(\mathbf{x}_t|\rho_j.\mu, \rho_j.\Sigma)}. \tag{4.11}$$

The agent then selects the region with the highest probability,

$$\rho^* = \arg\max_{\rho \in R} \gamma(\rho). \tag{4.12}$$

By weighting a GMM through action similarity, the agent prefers regions that are both proximate to the current state of the robots, as well as containing tactics that are suited to the observed behaviour of the adversary.

4.3.5.2 Tactic selection

Tactics are sampled from the set $\{\tau\}$ of the selected region ρ^*. The probability of selecting a tactic $\tau \in \rho^*.\{\tau\}$ is inversely proportional to the overall time of the trial the tactic was extracted from, i.e.

$$p(\tau|\rho^*) = \frac{1/\tau.T}{\sum_{\tau' \in \rho^*.\{\tau\}} 1/\tau'.T} \tag{4.13}$$

This prioritisation was enforced to penalise users who took a long time to score a goal during demonstration. Thus, we seek to reward tactics extracted from fast demonstrations without many redundant movements. However, alternative heuristics would also

be plausible in this step, depending on the interaction domain and the nature of the recorded data; one simple approach would be to sample from the tactics of the selected region uniformly-randomly, without any additional weighting. In Chapter 5, we introduce an empirical learning mechanism through which the utility of different tactics can be updated interactively.

Once a tactic τ^* is selected, its action $\tau^*.\vec{\mathbf{a}}_\mathbf{s}$ is followed for $\tau^*.dt$, before another region/tactic is selected again.

4.3.5.3 Recovery actions

A robot may not always be sufficiently close to one of the sampled interaction regions. In such a case, rather than selecting a tactic from an unrepresentative region, the agent moves closer to the sampled regions using a *recovery action*.

To this end, we define a threshold β for the probability $\gamma(\rho^*)$ of the most likely region returned by the mixture model. If $\gamma(\rho^*) < \beta$, the agent computes the vector linking its current pose to the mean of ρ^*, and selects an appropriate recovery action to move towards it. For example, if the agent finds itself close to the left side line, a plausible recovery action would be a right sidestep towards the centre. Once the probability is over the threshold again, the agent returns to the normal region/tactic selection mode.

4.4 Experimental results

4.4.1 Structure of the experiments

The aim of our experimental evaluation is twofold. First, we seek to assess how well our method can reproduce and synthesise the demonstrated strategies, in order to compete against the same adversary (the heuristic autonomous goalkeeper HAG described in Section 4.3.2). Second, we seek to evaluate the robustness of our approach against novel, human-controlled adversaries, who were not part of the demonstrated traces.

To this end, our experimental analysis is divided in three parts. First, we evaluate the performance of 30 human-controlled strikers (HCSs), in 5 trials each, against the HAG. The experimental sample was varied, consisting of both male and female subjects, young children and adults, users with previous robotics experience and users who were interacting with robots for the first time. Second, we evaluate an autonomous striker programmed through the procedure of Section 4.3 in 150 trials (30 independent

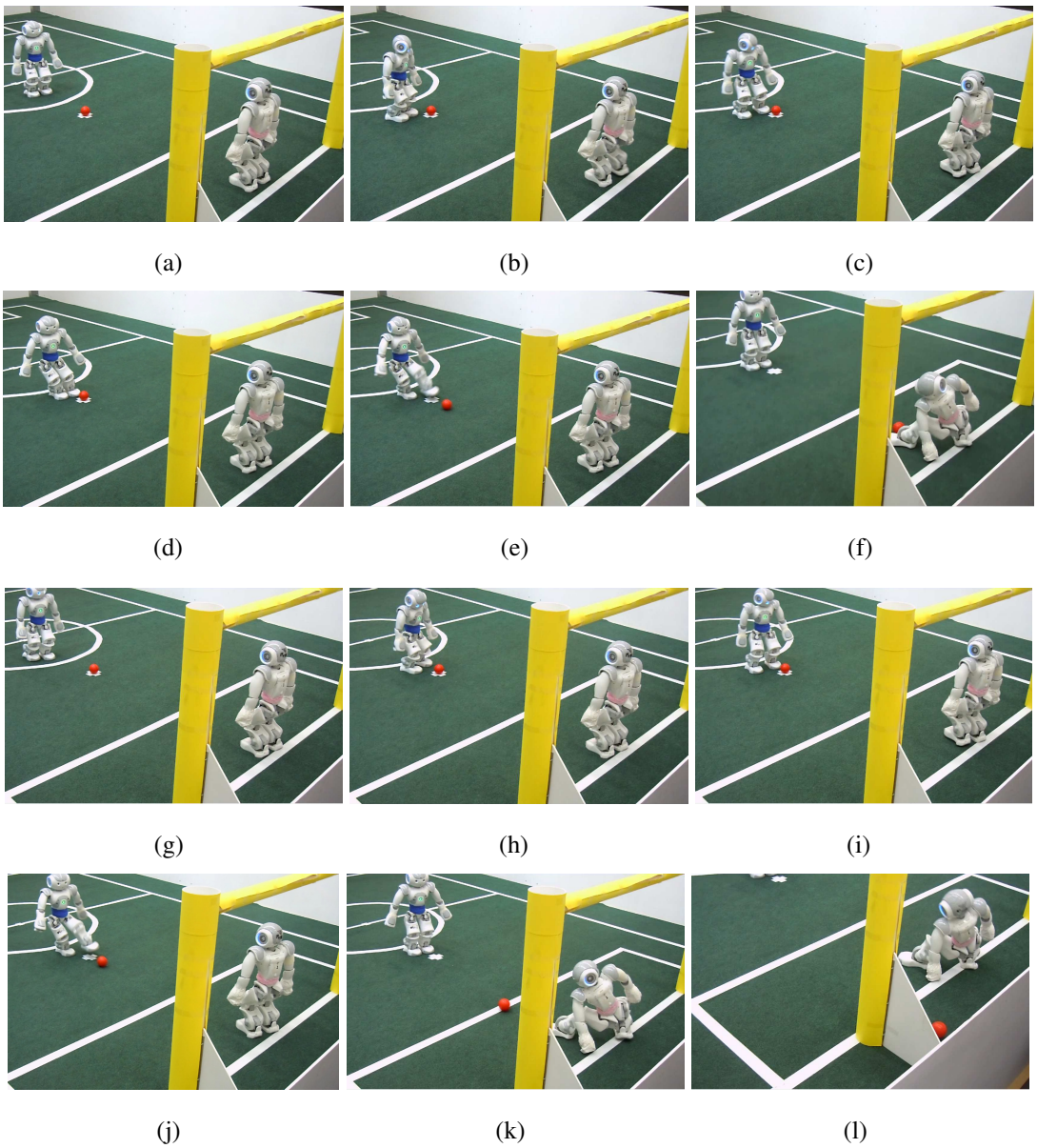


Figure 4.7: Examples of two trials by the strategy mixture striker (SMS) against a human-controlled goalkeeper (HCG). (a)-(f): Unsuccessful trial. (g)-(l): Successful trial.

sets of 5 trials) against the HAG. We refer to this autonomous agent as the Strategy Mixture Striker (SMS). Third, we evaluate the SMS against the same 30 subjects described above, who now operate as human-controlled goalkeepers (HCGs). Figure 4.7 shows snapshots from two trials involving the learned agent.

	HCSs vs HAG	SMS vs HAG	SMS vs HCGs
Total goals scored	61/150	77/150	71/150
Mean striker cuccess rate	40.6%	51.3%	47.2%
Standard deviation	$\pm$ 20.6%	$\pm$ 9.2%	$\pm$ 16.2%

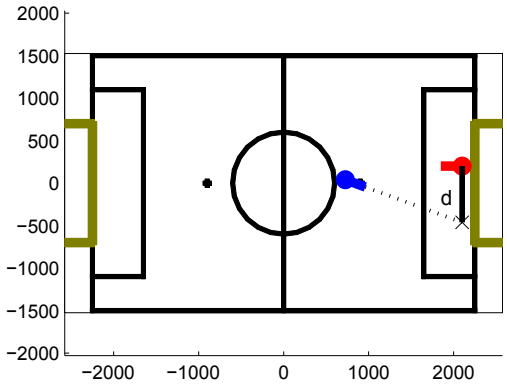
Table 4.1: Overall performance statistics for the three experiment phases. HCSs: Human-Controlled Strikers. HAG: Heuristic Autonomous Goalkeeper. SMS: Strategy Mixture Striker. HCGs: Human-Controlled Goalkeepers.

4.4.2 Performance evaluation

Table 4.1 lists the performance statistics for the three parts of the experiment. These overall results indicate a marginally better performance of the SMS compared to the average human-controlled striker against the HAG. When the standard deviation is taken into account, the SMS is seen to perform comparably to the best HCSs. These results indicate that our method can successfully reproduce the effectiveness of the demonstrated strategies versus the adversary they were demonstrated against. Similarly, when the goalkeeper changes to being human-controlled, the average performance of the SMS drops slightly, yet remains comparable to the success rate observed against the HAG. This indicates that the synthesised templates can be effective even against opponents who were not included in the demonstration process. However, the deviation of the success rate in this case also suggests that the SMS struggles against more able adversaries, whose responses are not fully captured by the learned templates.

Because of the dynamic nature of the interaction, the number of goals scored and conceded is not fully representative of an agent's performance. To address this issue, we assessed strikers on an additional performance metric: the distance of the goalkeeper from the optimal blocking position at the time of the striker's kick (Figure 4.8(a)). Through this metric, we model how well each goalkeeper was able to respond to the moves of the striker, and move to a position that will maximise the chances of a save.

As shown in Table 4.8(b), the strategy mixture striker is more successful at leading the HAG to a position from which it is harder to block a shot. This ability drops slightly against the human-controlled goalkeepers, but the SMS is still capable of outperforming the mean HCS in this metric. This suggests that the SMS is more robust at selecting action sequences that can impact the behaviour of the goalkeeper in the desired manner.



(a)

	HCSs vs HAG	SMS vs HAG	SMS vs HCGs
Mean distance of goalkeeper from optimal position (mm)	326.73	367.45	358.82
Standard deviation	139.37	108.29	196.66

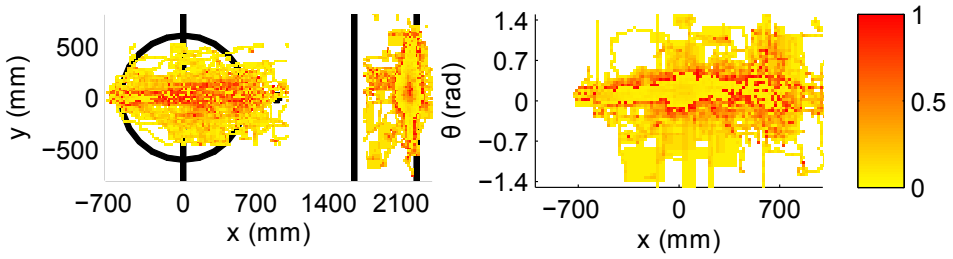
(b)

Figure 4.8: Performance metric: goalkeeper distance from optimal blocking position. (a): Explanation of metric. Poses of the striker and the goalkeeper at time of kick - optimal position for goalkeeper is the intersection of line formed by striker's orientation, and goalkeeper's line of motion. (b): Overall results for each of the three experiments.

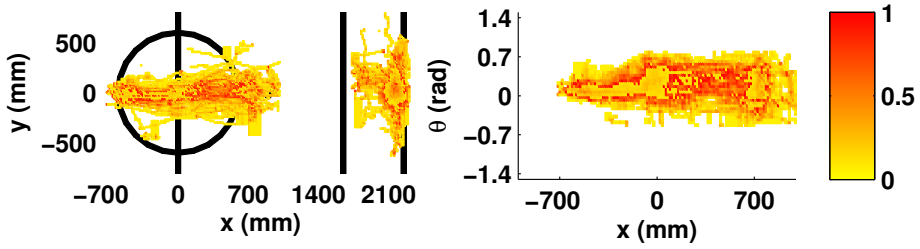
Furthermore, Figure 4.9 shows the heat maps for all trajectories recorded during the experiments. Variability is considerably greater in Figure 4.9(a), as the map collectively visualises attempts by different subjects with varying degrees of skill. Conversely, the heat maps of Figures 4.9(b) and 4.9(c) come closer to the successful demonstrations of Figure 4.3(b), indicating the ability of the SMS to *reproduce* the demonstrated strategies. However, these figures also shows trajectories and moves that were not directly captured by the demonstrations. This highlights the ability of our algorithm to *synthesise* and adapt the demonstrated strategies to novel adversaries.

4.5 Conclusions

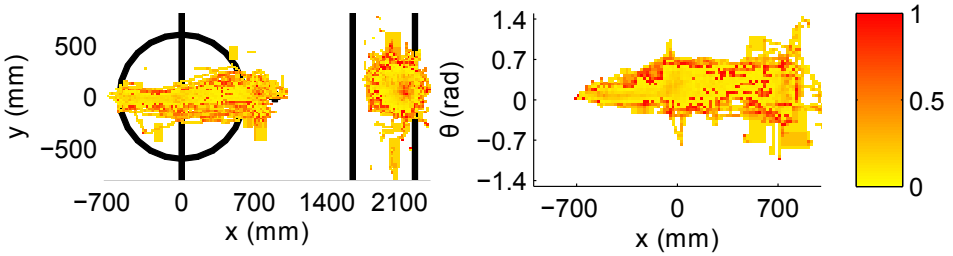
In this chapter, we present an algorithm for learning to interact with strategic human-controlled agents in adversarial environments. Our approach is novel in learning *strategic behaviours* from imperfect human demonstrations, which can be probabilistically synthesised to interact with previously unseen strategic adversaries. Results demonstrate that our procedure yields an autonomous agent that can consistently compete



(a) HCSs vs HAG.



(b) SMS vs HAG.



(c) SMS vs HCGs.

Figure 4.9: Heat map representations of all trajectories. Colour indicates the percentage of trials in which a particular point was recorded. *Left subplots:* $x-y$ motion trajectory components - left blob corresponds to striker, right blob to goalkeeper trajectories. *Right subplots:* $x-\theta$ motion for the striker.

with both autonomous and human-controlled robots, while outperforming most human subjects against a fixed, heuristic adversary.

A limitation of the current model is that it does not feature *online learning*, through which a robot could *interactively* adapt to a given adversary. Thus, the mixture of strategies introduced in this chapter may lead to overfitting, and fail against opponents who are not within the range of the demonstrated traces. In Chapter 5, we formulate a game-theoretic model of interaction control, where an autonomous robot can learn *from experience* to shape strategic interactions with other human-controlled robots. This constitutes an important step towards applications where robots can actively model and influence the behaviour of interacting strategic agents.

Chapter 5

Learning to shape and influence strategic interactions

5.1 Overview

One important issue for autonomous robots operating in interactive environments is that they must be able to adapt to a wide range of strategic adversaries. In the previous chapter, we described a procedure through which demonstrated interactive strategies can be *synthesised* against a given opponent. The selection of these templates is based on a dynamically weighted model, which accounts for the current state of the interaction and the observed recent behaviour of the adversary. However, one drawback of that approach is that the model cannot *learn* the utility of the various demonstrated strategies, based on observations from repeated interaction with that adversary. Thus, the learning component of that method is limited to the *offline* phase, where the local, composable strategies are extracted from the provided demonstrations.

In this chapter, we address this issue by introducing a Bayesian framework for learning, through *repeated interaction*, influencing behaviours for adversarial environments. Like the method of Chapter 4, the Bayesian framework also builds on behaviours that have been demonstrated by human operators. However, we now define probability distributions on these strategies, which are updated based on the observed effects of the executed actions. This probabilistic formulation extends the regret-based model introduced in Chapter 3, where the intent of the interacting adversary was inferred and updated over time. Moreover, we now consider *temporally extended sequences* of actions that are expected to change the state of the interaction at some future moment. Thus, unlike the approaches presented in Chapters 3 and 4, where

a single temporally extended action was chosen at every decision point, a robot can now plan a strategy that spans a greater time horizon. The resulting model constitutes our main contribution to the interaction shaping problem, which was presented in the introduction of this thesis (Section 1.1).

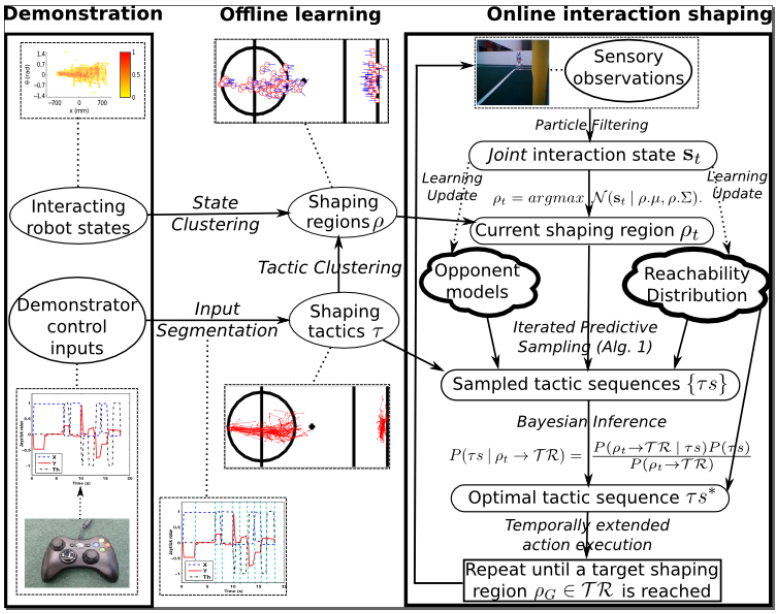


Figure 5.1: Our approach to strategic interaction shaping. Human demonstrators provide traces of the desired behaviour, which are converted into shaping regions and tactics, and used as composable templates by an autonomous agent. Online, the shaping agent attempts to reach a desired joint state by sampling tactic sequences through iterated prediction of its adversary's expected responses, and selecting optimal sequences through Bayesian inference. Opponent models and expected region reachability are updated through repeated interaction.

Our framework for strategic interaction shaping in adversarial mixed robotic environments (Figure 5.1) first learns offline a set of interaction templates from provided human demonstrations of the desired strategic behaviour. The demonstrations are encoded as ***shaping*** *regions* and *tactics*, which represent salient interactive modes of the state and action spaces. These templates are similar to the interaction regions and tactics introduced in Chapter 4, but are additionally characterised by local, empirically estimated opponent models. This allows the autonomous robot to track the responses of its adversary to individually executed tactics. Thus, opponent modeling is now conducted at the tactic and not at the region level.

In the online phase, the shaping robot seeks to lead the interaction to a target joint

state by chaining sampled sequences of tactics as an interactive strategy. To achieve this, the agent empirically updates a distribution over the *reachability* of the various shaping regions against the given adversary. The reachability metric measures the *compliance* of the adversary with executed actions, and thus indicates whether a given tactic is likely to lead to a desired state. Using this distribution, the interaction shaping problem is formulated as a two-step sampling and sequencing process. First, different tactic sequences are sampled through *iterated prediction* of the adversary's expected responses. Then, optimal sequences selected through *Bayesian inference* over the expected reachability of their traversed regions. Thus, the shaping robot learns, through repeated interaction, to choose temporally extended strategies that are likely to successfully shape an interaction with a given adversary.

In the remainder of this chapter, we first describe the interaction shaping method (Section 5.2), distinguishing between offline learning and online synthesis of interaction templates. We also describe the differences between the interaction regions and tactics introduced in the previous chapter, and the shaping templates described here. As before, our experimental scenario is the adversarial robotic soccer penalty shooting problem between NAO humanoid robots. However, we now focus on the ability of the resulting *shaping* autonomous robot to interact with and against human-controlled agents. Our results (Section 5.3) demonstrate an autonomous performance level comparable to that of the best human subjects when interacting with the same adversary, and an ability to improve shaping performance over time even against a challenging human-controlled adversary. Finally, we discuss the key contributions of this chapter and its connection to the rest of this thesis in Section 5.4. The methodology and results presented in this chapter also appear in (Valtazanos and Ramamoorthy, 2013a).

5.2 Method

5.2.1 Preliminaries and notation

We consider a system of two robots, R and R', interacting in a planar environment, where R is the *shaping* robot, and R' is the *shapeable* roboty. At time $t \in \Re^+$, the system is described by:

- The joint state of the two robots, $\mathbf{s}_t = \langle s_t, s'_t \rangle$, $s_t = [x_t, y_t, \theta_t]^T$, $s'_t = [x'_t, y'_t, \theta'_t]^T$, where $\{x_t, x'_t\} \in \Re$, $\{y_t, y'_t\} \in \Re$ are the positional coordinates, and $\{\theta_t, \theta'_t\} \in$

$[-\pi, +\pi]$ are the orientations of the robots[1].

- The action vectors, $\vec{a}_t$, $\vec{a}'_t$ available to the robots – each vector may consist of both discrete and continuous actions. For example, in a task involving navigation and manipulation, one choice for the action vector would be $[dx, dy, d\theta, grip]^T$, where $\{dx, dy, d\theta\} \in [-1.0, +1.0]$ are the requested translation and rotation as fractions of the maximum speed of the robot, and $grip \in \{open, close\}$ is a command for the robot's actuator.

The goal of R is to lead R' to one of several possible *target states* $\mathbf{z} \in \mathbf{Z}$, over a time horizon η, where each $\mathbf{z} = \langle s, s' \rangle$ represents a joint target configuration. In other words, R seeks to reach, at some time $t \leq \eta$, a joint state $\mathbf{s}_t \in \mathbf{Z}$.

In the remainder of this chapter, we will use the superscript ^{H} to denote states and actions of human-controlled robots. For example, s_t^H is the state of a human-controlled robot at time t.

5.2.2 Learning from human demonstration

As in the method introduced in Chapter 4, we learn basic templates of the desired interaction from demonstrations of human subjects performing the same task. As before, demonstrations are used to identify salient modes of the state and action spaces, which are used as "building blocks" for the learning algorithm of the shaping agent. However, we modify the original definition of interaction regions and tactics, so that different opponent models are now defined for every individual tactic. These models are formulated as the *inferred responses* of the adversary to a tactic, and are initially also learned from demonstration. Nevertheless, later in this section we show how these responses can be empirically updated during the interaction. We call the resulting action and state space templates ***shaping tactics*** and ***shaping regions***, respectively, in order to reflect their direct applicability to the interaction shaping problem.

5.2.2.1 Interaction shaping tactics

An interaction shaping tactic τ is a time-independent action continually evoked by R for a variable period of time:

$$\tau = \langle \mathbf{i\check{s}}, \mathbf{t\check{s}}, \check{a}, dt, \{\tilde{\mathbf{r}}\} \rangle, \tag{5.1}$$

[1] Positive directions: forward, left, counter-clockwise.

where $\mathbf{i\check{s}}, \mathbf{t\check{s}}$, are the joint input and target states of the tactic, $\vec{\mathbf{a}}_\mathbf{s}$ is the action followed by R, dt is the duration of this action, and $\{\tilde{\mathbf{r}}\}$ is a set of normalised *expected responses* of R' to τ. A response $\tilde{m} = \langle dx, dy, d\theta \rangle \in \{\tilde{\mathbf{r}}\}$ is a possible move by R', normalised over a time interval $\bar{\mathbf{n}}$, in response to τ. For the remainder of this chapter, we set $\bar{\mathbf{n}} = 1$ second.

The demonstration procedure is identical to the one described in the previous chapter. To recapitulate, we summarise the key features of this approach. Tactics are computed from recorded inputs of humans teleoperating the shaping agent R in the desired interaction. During this phase, the adversarial robot R' can be either also human-controlled, or an autonomous robot executing a hard-coded behaviour. The demonstrator-controlled robot R is teleoperated through a game controller, which maps inputs to action vectors $\vec{a}$.

For each demonstration, we also record the states of the two robots, s and s'. Thus, we obtain a time-indexed sequence of states and commands, $\mathbf{q} = \{\{t_1, s^H_{t_1}, s'_{t_1}, \vec{a}^H_{t_1}\}, ..., \{t_N, s^H_{t_N}, s'_{t_N}, \vec{a}^H_{t_N}\}\}$. Demonstrations are also annotated based on their outcome as *successful* or *unsuccessful* examples of shaping behaviour. We then retain the set of successful demonstrations, $\mathbf{Q}^+ = \{\mathbf{q}^+_1, ..., \mathbf{q}^+_M\}$.

For every extracted tactic τ, we record the start and end times at its boundaries, t_s and t_e. Then, the tactic time interval, input, and target states are $\tau.dt \leftarrow t_e - t_s$, $\tau.\mathbf{i\check{s}} \leftarrow \langle s^H_{t_s}, s'_{t_s} \rangle$, and $\tau.\mathbf{t\check{s}} \leftarrow \langle s^H_{t_e}, s'_{t_e} \rangle$. Similarly, the tactic action vector, $\tau.\check{a}$, is the mean of the inputs over the duration of τ.

The primary element of novelty in the new, shaping framework is the introduction of tactic-specific opponent models. These models are formulated as the expected responses of the adversary to τ, $\tau.\{\tilde{\mathbf{r}}\}$, which are initialised by dividing each tactic interval into $\lceil \tau.dt / \bar{\mathbf{n}} \rceil$ fixed-length segments. Each segment yields a candidate response by R', which is the change of the state of R' between the segment endpoints, averaged over its duration. We also set an upper bound $\boldsymbol{m}$ on the size of each $\{\tilde{\mathbf{r}}\}$, so if the number of tactic segments, $\boldsymbol{n}$, exceeds this bound, $\boldsymbol{n}$–$\boldsymbol{m}$ randomly selected candidate responses are discarded. We define this bound as

$$\boldsymbol{m} = \lceil c \cdot \tau.dt / \bar{\mathbf{n}} \rceil \tag{5.2}$$

where $c \geq 1$ is a small positive constant, such that the bound is only slightly greater than the duration of the tactic, $\tau.dt$, and the sampling interval, $\bar{\mathbf{n}}$.

Thus, each successful demonstration $\mathbf{q}^+$ is associated with a set of extracted tactics $\mathbf{q}.\{\tau\}$. By applying this procedure to all demonstrations in $\mathbf{Q}^+$, we obtain the set of all

shaping tactics, $\mathcal{ST} = \bigcup_{\mathbf{q} \in \mathbf{Q}^+} \mathbf{q}.\{\tau\}$.

5.2.2.2 Interaction shaping regions

A shaping region ρ is a normal distribution $\mathcal{N}$ over related states frequently visited by the robots during the interaction:

$$\rho = \langle \mu, \Sigma, \{\tau\} \rangle, \tag{5.3}$$

where μ is the mean joint state of ρ, Σ is the covariance matrix, and $\{\tau\}$ are the tactics that can be invoked from ρ.

The above definition constitutes a simplification of the interaction regions defined in Chapter 4, as opponent models are now migrated into individual tactics. However, the generation of shaping regions follows the same guidelines as before. In particular, shaping regions are computed by clustering the extracted tactics, $\mathcal{ST}$, based on their input states. We first form a set of tactic groups $TG = \{\tau g_1, ..., \tau g_M\}$ based on input state similarity, so that any two tactics τ_i, τ_j within a tactic group τg satisfy

$$d([\tau_i.\check{s}; \tau_i.\check{s}'], [\tau_j.\check{s}; \tau_j.\check{s}']) < \delta, \tag{5.4}$$

where $d(\cdot,\cdot)$ is the distance between two state pairs, and δ is a distance threshold defining the similarity between regions within the same group. Each group $\tau g \in TG$ is converted to a new region ρ, whose tactic set is $\rho.\{\tau\} = \tau g$. The parameters $\rho.\mu$ and $\rho.\Sigma$ are the mean and covariance of the input states of all tactics in $\rho.\{\tau\}$. Thus, we obtain a set of shaping regions, $\mathcal{SR}$.

In the problem we are considering, the shaping robot, R, seeks to lead an interaction with another robot, R', to one of several possible target states. To associate these states with specific regions, we first aggregate the last computed tactics of all successful demonstrations into the set $\mathcal{ST}_L = \{\mathbf{q}.\{\tau\}_{|\mathbf{q}.\{\tau\}|} \mid \mathbf{q} \in \mathbf{Q}^+\}$, where $|\mathbf{q}.\{\tau\}|$ is the number of tactics extracted from $\mathbf{q}$ (and hence also the index of the last tactic of the set $\mathbf{q}.\{\tau\}$). Thus, the resulting set $\mathcal{ST}_L$ is a subset of $\mathcal{ST}$. Then, we apply the above clustering and partitioning procedure on $\mathcal{ST}_L$ (as opposed to the whole of $\mathcal{ST}$ like above). This leads to the set of *target regions*, $\mathcal{TR} \subset \mathcal{SR}$, representing states the shaping agent eventually seeks to reach.

5.2.3 Bayesian interaction shaping

This section describes the main algorithmic component of the proposed interaction shaping framework. Given a set of computed shaping tactics and regions, we formulate our approach to this problem as a two-step *sampling* and *selection* process over these templates, which is followed by an *online learning* update. The goal for the shaping robot, R is to identify *sequences of tactics*, $\{\tau_1, ..., \tau_N\}$, that are likely to lead the adversarial agent, R', to a desired target joint state. To achieve this, R first samples multiple tactic sequences that are likely to reach a target shaping region. This procedure iteratively predicts the expected state of the interaction following the execution of a sampled tactic, by sampling from the set of opponent responses for that tactic. Thus, by chaining these estimates together, R can also predict the expected state of the interaction at the completion of a sequential execution of the tactics.

In the selection process, an optimal sequence is selected from the collected samples based on the posterior probability of reaching a target region. This probability is computed through Bayesian inference over the discounted expected reachability of a sequence's constituent tactics.

The final stage of the shaping framework is concerned with learning of the employed reachability and response models. We describe empirical learning mechanisms for these updates, which extend and follow on the regret minimisation techniques introduced in Chapter 3.

5.2.3.1 Empirical reachability likelihood

The success of an interaction shaping strategy depends on the *compliance* of R' with tactics selected by R. To model this effect, we define the *empirical reachability likelihood* distribution, $\mathcal{RD}$, which expresses the probability of reaching a region with a given tactic:

$$\mathcal{RD}(\rho_1, \tau, \rho_2, \rho_3) \doteq P(\rho_3 \mid \rho_1, \tau, \rho_2). \tag{5.5}$$

Thus, $\mathcal{RD}(\rho_1, \tau, \rho_2, \rho_3)$ gives the probability of reaching ρ_3, given that τ was invoked from ρ_1 with the intention of reaching ρ_2. As explained in Section 5.2.3.2, the correlation between intended and actually reached regions is the main bias in selecting robust shaping tactics. We initialise $\mathcal{RD}$ assuming "perfect control" over tactics, so that

$$P(\rho_3 \,|\, \rho_1, \tau, \rho_2) = \begin{cases} 1, & \rho_2 = \rho_3 \\ 0, & \rho_2 \neq \rho_3 \end{cases} . \tag{5.6}$$

In other words, R is initially assumed to always be able to successfully influence R', and complete the execution of a tactic by reaching the desired shaping region. However, we later show how these probabilities are continually updated during the interaction, based on the inferred effects of the executed tactics.

5.2.3.2 Tactic sampling and iterated prediction

The number of possible tactic sequences may vary depending on the quantity and quality of the provided demonstrations. Thus, exhaustive search over these templates may often be infeasible. To address the complexity of tactic sequence selection, we establish bounds for the maximum number of sequences, N_S, and the length of each sequence, L_S, sampled at every decision point. We set $N_S = \max_{\rho \in \mathcal{SR}} |\rho.\{\tau\}|$ (size of largest tactic set), and $L_S = (\max_{\mathbf{q} \in \mathbf{Q}^+} |\mathbf{q}|)$ (size of longest demonstration).

The robot R first estimates the joint state of the world, $\mathbf{s}_t$, at the current time t, based on its sensory observations. Then, the most likely current region, ρ_t, is selected as

$$\rho_t = \arg\max_{\rho \in \mathcal{SR}} \mathcal{N}(\mathbf{s}_t \,|\, \rho.\mu, \rho.\Sigma). \tag{5.7}$$

To generate a new tactic sequence, τs, we first select a tactic τ from $\rho_t.\{\tau\}$ with probability

$$P(\tau \sim \rho_t.\{\tau\}) = \frac{1}{|\mathcal{SR}|} \sum_{\acute{\rho}} P(\acute{\rho} \,|\, \rho_t, \tau, \acute{\rho}), \tag{5.8}$$

so as to reflect the overall expected *successful* reachability of regions from ρ_t using τ. In other words, the above probability reprsents the overall accordance between expected and actually reached regions for a given tactic τ. Then, we iteratively predict how the interaction may evolve if R follows τ. The expected state of R, $\tilde{s}$, upon completion of τ, is the target state $\tau.\mathbf{t\check{s}}$. For R', we sample $\lceil \tau.dt/\bar{\mathbf{n}} \rceil$ responses from $\tau.\{\tilde{\mathbf{r}}\}$. Starting from the current state of R', $\tilde{s}' = s'_t$, we iteratively apply each sampled response, $\tilde{m}$, to $\tilde{s}'$, i.e.

$$\tilde{s}' \leftarrow \tilde{s}' + \sum_{i=1}^{\lceil \tau.dt/\bar{\mathbf{n}} \rceil} \tilde{m}_i \tag{5.9}$$

Through this iterative application of the predicted responses of the adversary, we obtain the expected joint state, $\tilde{\mathbf{s}} = \langle \tilde{s}, \tilde{s}' \rangle$, at the end of τ.

We then repeat the procedure of Equation 5.7, in order to compute the most likely region of $\tilde{\mathbf{s}}$, $\tilde{\rho} = \arg\max_{\rho \in \mathcal{SR}} \mathcal{N}(\tilde{\mathbf{s}} \,|\, \rho.\mu, \rho.\Sigma)$. We call $\tilde{\rho}$ the *expected next region* of τ, denoted as $\tau.\rho_{+1}$. If $\tilde{\rho}$ is one of the target regions $\mathcal{TR}$, we stop and return the tactics sampled so far as a tactic sequence τs. Otherwise, we repeat the above iterated prediction process, until either a target region is found, or the maximum sequence length L_S is reached. We repeat the whole procedure to obtain N_S sequence samples. The overall tactic sequence sampling method is summarised in Algorithm 5.

The tactic sampling algorithm predicts the evolution of at most $N_S \cdot L_S$ tactics in the worst case. The ability to find sequences leading to a target region depends on the convergence of the sampled adversarial responses. If these samples are a good representation of the "true" behaviour of R' in response to a given tactic, the expected next regions of that tactic will tend to match the actual reached regions. In interactions with non-stationary adversaries, however, R may not be able to find sequences leading to a target region, owing to the discrepancy between expected and reached states. If no such tactic sequence is found, R attempts to transition to a different region from which better sequences may be retrieved. Thus, given the interactive nature of our problem domain, our objective is not an exhaustive search over tactics, but instead an efficient sampling procedure yielding bounded-length sequences that are likely to impact the interaction.

5.2.3.3 Tactic selection

In the sampling process, tactic sequences are generated with the intention of reaching a target region $\rho_G \in \mathcal{TR}$. In the selection process, we seek to identify the sequence that is most likely to achieve this terminal objective. We formulate this problem in a Bayesian setting, under the assumption that all ρ_G are equally desirable for R. In this context, we define the *posterior probability* of selecting a tactic sequence, τs, given that R wants to eventually reach a region $\rho_G \in \mathcal{TR}$ from the current shaping region ρ_t:

$$P(\tau s \,|\, \rho_t \to \mathcal{TR}) = \frac{P(\rho_t \to \mathcal{TR} \,|\, \tau s) P(\tau s)}{P(\rho_t \to \mathcal{TR})}. \tag{5.10}$$

The *prior probability* of selecting a sequence τs is inversely proportional to the overall expected duration of its constituent tactics. We refer to this duration as the *inverse total time* of the sequence, $\mathbf{T}^{-1}(\tau s) = 1/(\sum_{\tau \in \tau s} \tau.dt)$. Then, the prior probability,

Algorithm 5 Tactic Sequence Sampling

TACTICSEQUENCESAMPLING$(\mathbf{s}_t, \mathcal{SR}, \mathcal{TR}, \mathcal{RD}, L_S, N_S, \bar{\mathbf{n}})$
Input: Current joint state $\mathbf{s}_t$, shaping regions $\mathcal{SR}$, target regions $\mathcal{TR}$, reachability distribution $\mathcal{RD}$, maximum sequence length L_S, maximum sampling attempts N_S, response interval $\bar{\mathbf{n}}$
Output: Set of tactic sequences $\{\tau s\}$
$\{\tau s\} \leftarrow \{\{\}\}$
$\rho_t \leftarrow \arg\max_{\rho \in \mathcal{SR}} \mathcal{N}(\mathbf{s}_t \,|\, \rho.\mu, \rho.\Sigma)$ {find current region}
for $i = 1 \rightarrow N_S$ **do**
    $\tau s \leftarrow \{\}$ {initialise new tactic sequence}
    $j \leftarrow 1$
    $\tilde{\rho} \leftarrow \rho_t$
    $(\tilde{\mathbf{s}} \equiv \langle \tilde{s}, \tilde{s}' \rangle) \leftarrow \mathbf{s}_t$
    while $\tilde{\rho} \notin \mathcal{TR}$ **and** $j \leq L_S$ **do**
        $\tilde{\tau} \sim \tilde{\rho}.\{\tau\}$ {sample tactic using $\mathcal{RD}$ as in Eq. 5.8}
        $\tilde{s} \leftarrow \tilde{\tau}.\mathbf{t\check{s}}$ {own expected state $\leftarrow$ tactic target}
        for $j = 1 \rightarrow \lceil \tilde{\tau}.dt / \bar{\mathbf{n}} \rceil$ **do**
            $\tilde{m} \sim \tilde{\tau}.\{\tilde{\mathbf{r}}\}$ {sample from tactic responses}
            $\tilde{s}' \leftarrow \tilde{s}' + \tilde{m}$ {apply sample}
        end for
        $j \leftarrow j + 1$
        $\tilde{\mathbf{s}} \leftarrow \langle \tilde{s}, \tilde{s}' \rangle$
        $\tilde{\rho} \leftarrow \arg\max_{\rho \in \mathcal{SR}} \mathcal{N}(\tilde{\mathbf{s}} \,|\, \rho.\mu, \rho.\Sigma)$
        $\tilde{\tau}.\rho_{+1} \leftarrow \tilde{\rho}$ {assign $\tilde{\rho}$ as expected next region}
        τs.`insert`$(\tilde{\tau})$
    end while
    $\{\tau s\}$.`insert`(τs)
end for
return $\{\tau s\}$

$P(\tau s)$ is formulated as

$$P(\tau s) = \beta(\tau s) \cdot \frac{\mathbf{T}^{-1}(\tau s)}{\sum_{\tau s' \in \{\tau s\}} \mathbf{T}^{-1}(\tau s')}, \tag{5.11}$$

where $\beta(\tau s)$ is a coefficient penalising sequences whose last tactic, τ_N, is not expected

to reach a target region, i.e.:

$$\begin{cases} \beta(\tau s) = 1, & \tau_N.\rho_{+1} \in \mathcal{TR} \\ 0 < \beta(\tau s) < 1, & \tau_N.\rho_{+1} \notin \mathcal{TR} \end{cases}. \tag{5.12}$$

This factor is incorporated to account for sampled sequences that did not reach a desired state within the defined sampling bounds of Algorithm 5. Thus, through $P(\tau s)$, shorter sequences expected to reach a target shaping region are a priori more preferable.

The *likelihood* of reaching a target region given a sequence τs, $P(\rho_t \to \mathcal{TR} \,|\, \tau s)$, is computed as the discounted sum of the empirical reachability likelihoods of the constituent tactics of τs,

$$P(\rho_t \to \mathcal{TR} \,|\, \tau s) = \frac{\beta(\tau s)}{|\tau s|} \sum_{i=1}^{|\tau s|} \gamma^{i-1} \cdot P(\rho_{+1} \,|\, \rho_{-1}, \tau_i, \rho_{+1}) \tag{5.13}$$

where $\beta(\tau s)$ is defined as in Equation 5.12, $0 < \gamma \leq 1$ is a discount factor incrementally penalising the contribution of future tactics to the likelihood sum, τ_i is the i-th tactic of τs, ρ_{+1} is the expected next region of τ_i (line 21 of Algorithm 5), and ρ_{-1} is the previous region of a tactic τ_i,

$$\rho_{-1} = \begin{cases} \rho_t, & i = 1 \\ \tau_{i-1}.\rho_{+1}, & i > 1 \end{cases}. \tag{5.14}$$

Thus, the likelihood $P(\rho_t \to \mathcal{TR} \,|\, \tau s)$ provides a measure of the expected discounted *compliance* of the adversary with a tactic sequence.

Finally, the *normalisation term*, $P(\rho_t \to \mathcal{TR})$, is given by

$$P(\rho_t \to \mathcal{TR}) = \sum_{\tau s' \in \{\tau s\}} P(\rho_t \to \mathcal{TR} \,|\, \tau s') P(\tau s'). \tag{5.15}$$

We select the optimal tactic sequence τs^* through maximisation over the posterior distribution, i.e.

$$\tau s^* = \arg \max_{\tau s \in \{\tau s\}} P(\tau s \,|\, \rho_t \to \mathcal{TR}). \tag{5.16}$$

5.2.3.4 Belief updates

The robustness of the sampled and selected strategies depends on the ability to model the effects of the executed tactics, and the adaptation to the observed behaviour of a

given adversary. To this end, the shaping robot R learns to influence an adversary R' by updating the tactic expected responses and the region reachability distribution $\mathcal{RD}$. Through these learning updates, the sequence sampling and selection procedures are biased to favour shaping tactics that more closely account for the observed behaviour of R'.

-Learning adversary responses: When executing a tactic τ, R observes the responses of R', and uses them to update the set $\tau.\{\tilde{\mathbf{r}}\}$. The tactic time interval, $\tau.dt$, is divided into $\lceil \tau.dt/\bar{\mathbf{n}} \rceil$ segments, and the observed state change of R' is recorded for each segment. If t_1, t_2 are the times at the endpoints of a segment σ, the candidate response $\tilde{m}$ for σ is

$$\tilde{m} = s'_{t_2} - s'_{t_1}. \tag{5.17}$$

A candidate $\tilde{m}$ is added to the existing set of expected responses of a tactic so as not to violate the bound on the maximum number of these responses, $\boldsymbol{m}$. If $\tau.\{\tilde{\mathbf{r}}\}$ already has $\boldsymbol{m}$ samples, the oldest sample is replaced by $\tilde{m}$. Otherwise, $\tilde{m}$ is simply appended to the set. Through this update, $\tau.\{\tilde{\mathbf{r}}\}$ is biased to reflect the most recently observed reactions of R' to τ.

Adversarial responses model the *local* reactive behaviour of R', without making explicit assumptions about the long-term reasoning or strategic intentionality of that agent. These effects are implicitly addressed by the iterated predictions and expectations of shaping regions, which model the compliance of R' with a temporally extended sequence of actions.

-Learning region reachability: Upon completion of a tactic τ, R updates $\mathcal{RD}$ based on the resulting shaping region ρ_c. If τ was invoked from region ρ_i with the intention of reaching ρ', we update the probability $P(\rho|\rho_i,\tau,\rho')$ based on the rule

$$P(\rho|\rho_i,\tau,\rho') = \begin{cases} P(\rho|\rho_i,\tau,\rho') + w, & \rho = \rho_c \\ P(\rho|\rho_i,\tau,\rho') - \frac{w}{|\mathcal{SR}|-1}, & \forall \rho \neq \rho_c \end{cases} \tag{5.18}$$

where $0 < w < 1$ is the update weight. Probabilities are also normalised after each weight update so that they all lie between 0 and 1. Thus, the distribution progressively assigns higher weight to region-tactic-region transitions that have been empirically found to be reachable. These transitions are then favoured in subsequent tactic sampling and selection iterations, as moves that are likely to lead the adversary to a desired target joint state.

-Tactic sequence update frequency: The *update interval*, $\mathbf{u}$, is the number of tactics after which a new sequence should be selected, based on the updated beliefs. For $\mathbf{u} = 1$, a new τs will be selected upon completion of every executed tactic. If $\mathbf{u}$ is greater than 1, the shaping robot will execute multiple tactics from a sequence before replanning its strategy.

5.3 Experimental Results

We evaluate the Bayesian interaction shaping framework on the previously described penalty shooting problem between autonomous and teleoperated NAO robots (Sections 1.3, 4.2). However, in this section, we focus on the *learning* and *shaping* ability of the autonomous robot, in the context of interactions with an adversarial agent. Thus, we do not restrict our presentation to overall performance results, but we also give a qualitative and quantitative evaluation of the learning process.

We first describe how shaping regions and tactics are learned from human demonstration. The set of provided demonstrations is identical to the one described and used in Chapter 4. We then illustrate the learning performance of our algorithm in a variety of interactions with autonomous and human-controlled adversarial agents.

5.3.1 Shaping region and tactic computation

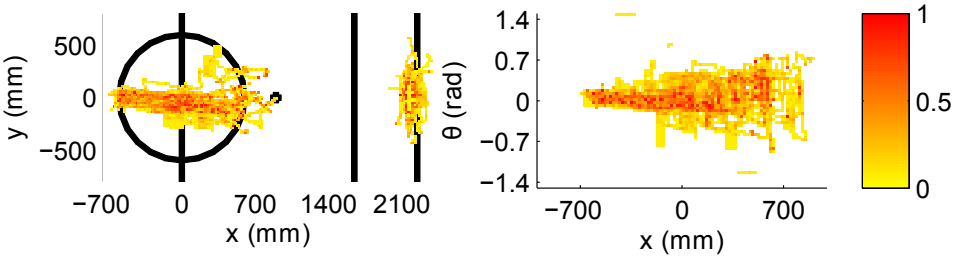
The shaping templates were learned from demonstration by subjects with prior experience of the NAO robots. Demonstrators provided traces of the desired behaviour by controlling the striker against a heuristic autonomous goalkeeper (HAG), which runs the algorithm described in Section 4.3.2. We review the key features of this algorithm: given the striker's current orientation, θ, the expected ball trajectory is a straight line segment starting at the ball position, and following this angle. Then, the HAG moves to the point where this segment intersects the goal line, as the expected best blocking position. The HAG may also dive to block the ball when it detects it to move towards the goal line.

For each trace, we recorded the demonstrator inputs (dx, dy, $d\theta$ motion and kick commands) and the poses of the robots. In total, 29 successful trials were retained (Figure 5.2(a)). By applying the procedure of Section 5.2.2, the collected data yielded a total of 134 shaping regions and 320 shaping tactics. Figure 5.2(b) shows the means of the computed regions, whereas Figure 5.2(c) provides a graph representation of the

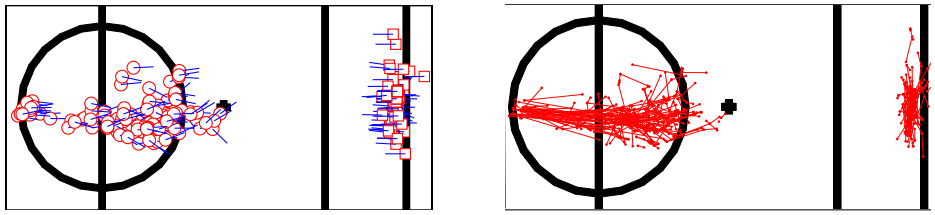
computed tactics, illustrating how they can be chained to form a sequence.

Target regions represent joint states from which the striker is likely to score. In these states, the striker (R) should be within the kicking distance of 190mm from the penalty mark, $pm = [px, py]$, where the ball is initially positioned, and the goalkeeper (R') should be on the goal side opposite the striker's orientation, so that either $\theta > 0$ and $y' < 0$, or $\theta < 0$ and $y' > 0$. In other words, R seeks to reach (one of) the regions with the properties:

$$\mathcal{TR} = \{\rho \mid d(\rho.\mu.[x:y], pm) < 190 \text{ and } ((\rho.\mu.\theta > 0 \text{ and } \rho.\mu.y' < 0) \text{ or } (\rho.\mu.\theta < 0 \text{ and } \rho.\mu.y' > 0))\}. \tag{5.19}$$



(a)



(b) (c)

Figure 5.2: Learning shaping templates. (a): Heat map representation of successful demonstrations – these are the traces recorded in the laboratory capture session described in Section 4.3.2. Colour indicates percentage of trials in which a point was recorded. *Left:* (x) - (y) motion, both players. *Right:* (x) - (θ) motion, striker. (b): Means of computed regions. Circles: striker states. Squares: goalkeeper states. Lines: orientations. A region mean comprises *both* a striker and a goalkeeper state. (c): Tactic graph – edges represent desired transitions between input and target regions.

5.3.2 Shaping agent evaluation

The evaluation of the autonomous interaction shaping striker (ISS) is twofold. First, we compare this agent ***with*** several human-controlled strikers (HCSs), in interactions

with the HAG. Our objective is to compare the performance of these two agent types when they compete against the same adversary. Then, we evaluate the ISS ***against*** a more challenging human-controlled adversarial goalkeeper. Here, we assess how the interaction shaping performance is affected when the adversary is a truly strategic, human-controlled adversary, whose exact behavioural model is not known a priori.

To this end, we conduct three different experiments. First, we evaluate the performance of 30 human subjects, in 5 trials each, acting as strikers against the HAG. This phase is identical to the first part of the experiments of Section 4.4, so only the relevant results are summarised here for comparison purposes. Second, we evaluate the ISS against the same adversary (HAG), in 10 independent sets of 25 trials each. Third, we repeat the procedure of the second experiment, but we now evaluate the ISS against an expert human-controlled goalkeeper (EHCG), who is teleoperated by an experienced member of our research group.

The EHCG is a considerably harder adversary for two reasons. First, the human operator has *full visibility* of the environment through his own eyes, as opposed to autonomous robots that rely on their noisy camera feed. Second, the operator can learn to anticipate adversarial moves over time, in contrast to the HAG which has a fixed, non-adaptive behaviour. Thus, against the EHCG, the ISS must learn to shape interactions with another learning adversarial agent.

In the last two experiments, the ISS updates adversarial responses and region reachabilities using the parameters $N_S = 20$, $L_S = 10$, $\beta = 0.1, \gamma = 0.7, w = 0.1, \mathbf{u} = 1$.

Interaction (Striker vs Goalkeeper)	**HCSs vs HAG**	**ISS vs HAG**	**ISS vs EHCG**
Total goals scored	61/150	138/250	92/250
Mean striker success rate	40.67%	55.20%	36.80%
Standard deviation	± 20.60%	±5.72%	±6.67%

Table 5.1: Overall results. HCSs: Human-Controlled Strikers. ISS: Autonomous Interaction Shaping Striker. HAG: Heuristic Autonomous Goalkeeper. EHCG: Expert Human-Controlled Goalkeeper.

The overall results are shown in Table 6.7. When competing against the HAG, the ISS performs considerably better than the *mean* HCS. Furthermore, when the standard deviation is taken into account, the success rate of the ISS is found to be comparable to the *best* instances of HCS (around 60%). This suggests that the shaping template

formulation and learning procedure can successfully generate strategic behaviours that match the sophistication of experienced human users. By contrast, the shaping ability of the ISS drops considerably against the more challenging EHCG, as indicated by the reduced success rate, which is however still comparable to the mean rates achieved by HCSs against the HAG.

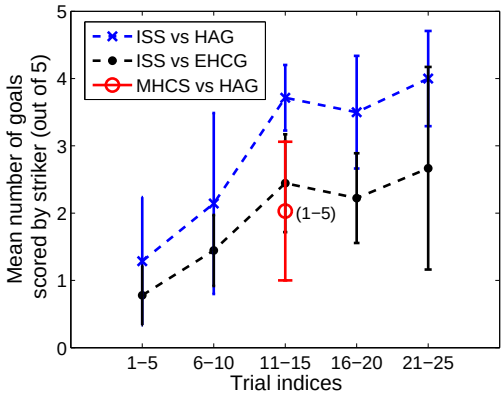
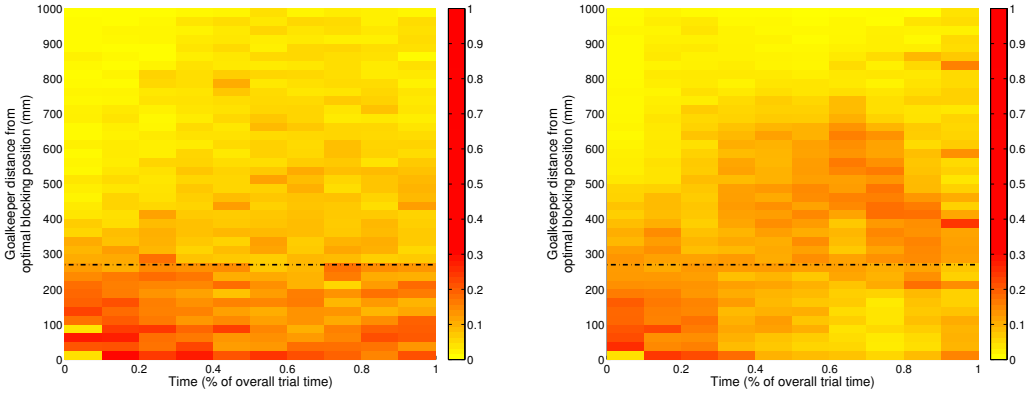


Figure 5.3: Inter-trial performance of the ISS. Each experimental run of 25 trials is split into blocks of 5, with results averaged over all 10 runs. The mean HCS success rate (MHCS), as averaged over the 5 trials taken by each of the 30 subjects, is also given.

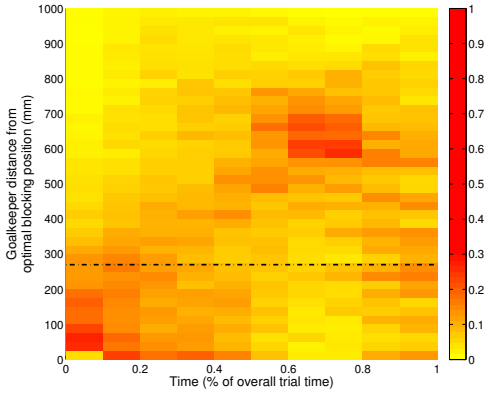
To better understand how the ISS learns to shape interactions over time, we divided the sets of 25 trials of the second and third experiments into blocks of 5, and we measured the mean number of goals scored in each block. Thus, we seek to assess how the performance of the ISS varies across these blocks. The resulting scores are shown in Figure 5.3. Despite the discrepancy in the number of goals scored against the two adversaries, we observe that the overall progression rate is similar. In both cases, the ISS begins with a low success rate, which improves as interaction progresses. This is an important result demonstrating that the learning rate of our algorithm is not affected by the strategic sophistication of the adversary. Thus, even when the ISS is pitted against an adversary controlled by an expert human operator, it can empirically learn strategies that improve its success rate.

Despite giving a strong indication of overall performance, goal-scoring rates do not show ***how*** the various strikers tried to influence their adversaries. To address this issue, we measured the *distance of the goalkeeper from the optimal blocking position*, d^*, which was introduced in the previous chapter (Figure 4.8(a)). However, we now measure this distance throughout an experimental trial, and not just at the end. Through this modification, we model how well each striker influenced goalkeepers into moving



(a) HCSs vs HAG

(b) ISS vs HAG



(c) ISS vs EHCG

Figure 5.4: Goalkeeper distances from optimal blocking position, d^* (see Figure 4.8(a) for an explanation of this metric, which was introduced in the previous chapter). Illustrations of time-indexed heat maps of distances - colour indicates percentage of trials in which a particular time-distance pair was recorded. The black dotted line (d = 270mm) shows the expected minimum distance required to score – this is the length covered by the goalkeeper's leg after a dive to save the ball. (a): HCSs vs HAG. (b): ISS vs HAG. (c): ISS vs EHCG.

to a suboptimal position over a temporally extended interval. Thus a good shaping behaviour should succeed in maximising d^* at the end of a trial.

As shown in Figures 5.4(a)-5.4(c), the ISS was more successful at maximising d^* than most HCSs, thus more explicitly trying to shape interactions. Moreover, in both ISS experiments, the dominant pattern is that d^* is initially small, reaching its maximum value around the midpoint of the trial and then dropping again. However, when competing against the EHCG, d^* drops more sharply towards the end. This

indicates that the expert user is more adept at recovering from deceptive moves by the striker than the HAG, thus preventing the interaction from being shaped at his expense.

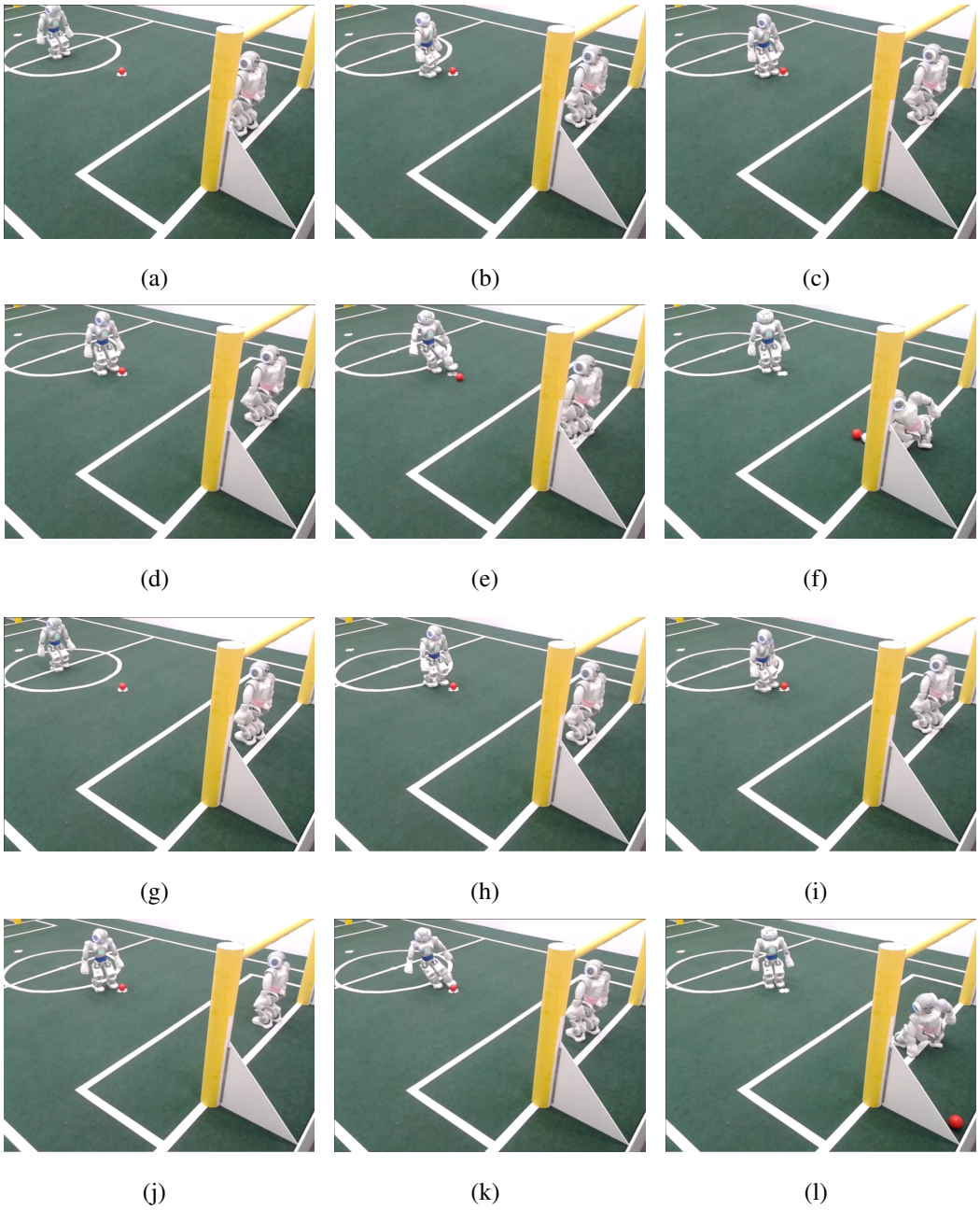


Figure 5.5: Six snapshots from two trials, ISS vs HAG. (a)-(f): Unsuccessful attempt. (g)-(l): Successful attempt. The two strategies are similar, but in the second trial, the ISS waits longer for the HAG to move towards the far side of the goal (3rd-4th snapshots), before turning to shoot towards the near side (5th-6th snapshots). Thus, the HAG is deceived into having less time to respond, and the interaction is shaped more effectively.

Furthermore, Figure 5.5 shows snapshots from two trials of the ISS against the

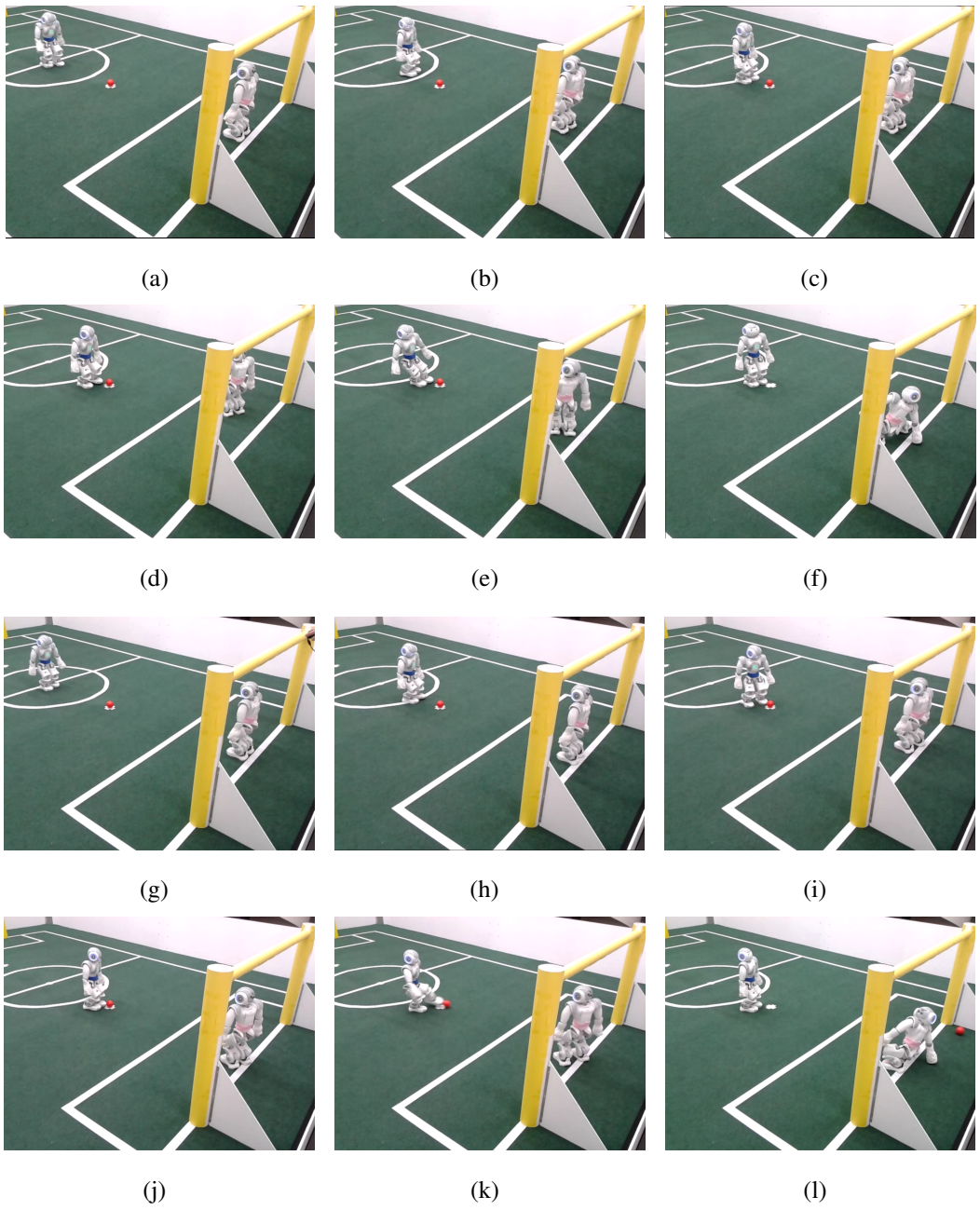


Figure 5.6: Six snapshots from two trials, ISS vs EHCG. (a)-(f): Unsuccessful attempt. (g)-(l): Successful attempt. In the successful trial, the strategy followed by the striker resembles the moves illustrated in the second half of Figure 5.5.

HAG. In both cases, the ISS first turns to face the far side of the goal, before turning to the near side and shooting. However, in the successful attempt, the ISS waits longer during the first turn, in order to make the HAG move closer to the far side and reduce its subsequent recovery time. Thus, d^* is greater at the end of the trial, and the ISS manages to shape the interaction more effectively.

A similar pattern is observed in Figure 5.6, which shows snapshots from trials of the ISS against the EHCG. In the first example, the striker follows a simple approach and kick, which is not sufficient to deceive the human controller of the goalkeeper. However, in the second example, the robot has learned a similar strategy to the second trial of Figure 5.5, where successive direction changes have been found to increase the likelihood of leading the goalkeeper to a desired state.

Further examples and illustrations from the above experiments are available in the supporting video of this chapter (Valtazanos, 2012a).

5.4 Conclusions

In this chapter, we present a framework for strategic interaction shaping in mixed robotic environments. Our approach combines offline learning of shaping regions and tactics from human demonstrations, and online learning and synthesis of these templates through Bayesian inference over the adversary's expected behaviour. This method extends the implementation of Chapter 4, where online interactive adaptation to a novel adversary is not supported by the framework. Experimental results demonstrate that the shaping agent can shape interactions with a given heuristic adversary comparably to the best human subjects, as identified from a diverse group of 30 individuals. Moreover, the shaping agent can successfully learn, through repeated interaction, to improve its performance against a challenging, human-controlled adversary, who is empirically shown to be less susceptible to deceptive behaviours. Thus, our work constitutes a novel, practical approach to online strategic learning in physical robotic systems, in interactions with unknown, potentially human-controlled, adversarial agents.

In the following chapter, we evaluate interactive human-robot decision making from a different perspective. In particular, we look at the ability of humans to interact with strategic agents who intend to influence their beliefs, such as the ones presented in this and in the previous chapter. To this end, our next experiment assesses the effects that factors such as limited visibility of the interaction environment have on human decisions, when the adversarial autonomous robot is acting strategically. Thus, we demonstrate why interaction shaping and related decision problems are challenging not only for autonomous robots, but also for human operators.

Chapter 6

Perceptual constraints in interactive teleoperation

6.1 Overview

In the preceding chapters, we described learnable autonomous behaviours that can shape and impact the beliefs of teleoperated agents. In this chapter, we study the related problem of how humans respond to such behaviours, and what are the factors that affect their responses. This is an important issue for many existing systems that depend on teleoperation, for example, rescue robot and de-mining teams in field applications. These systems often feature both fully autonomous robots and robots teleoperated by humans, or even physically present humans.

Interaction and coordination between such heterogeneous agents is a challenging task, largely due to their varied actions, perception, and cognitive capabilities. When looking at how humans (tele)operate in mixed domains, it is important to assess how these heterogeneous capabilities affect their ability to make robust *decisions*, in the presence of other, possibly adversarial, interacting agents. Humans are generally believed to have a superior grasp of context and situational awareness than autonomous robots. This is one reason why most deployed systems still depend quite heavily on the human user to control robots. However, this awareness also depends on the *perceptual* information made available to operators, which influences how they perceive their own robot's surroundings and the state of other interacting robots.

In many realistic situations, this information may be sparse or incomplete; for example, an operator controlling a rescue robot in a disaster site may only have access to the robot's noisy camera feed. Thus, a user having *full* visibility of the environment

may be able to fully understand how other agents are behaving, and plan the actions of the teleoperated robot accordingly. By contrast, if the same person has *limited* visibility of the environment, the decisions may be less informed and thus less effective. In the latter case, where users are effectively constrained to have the same perceptual capabilities as a robot, it is unclear whether their decisions would be able to exceed, or even match, the performance level of an autonomous agent (Figure 6.1). This is an important issue to be addressed in systems where the autonomous system can intervene to assist the human partner.

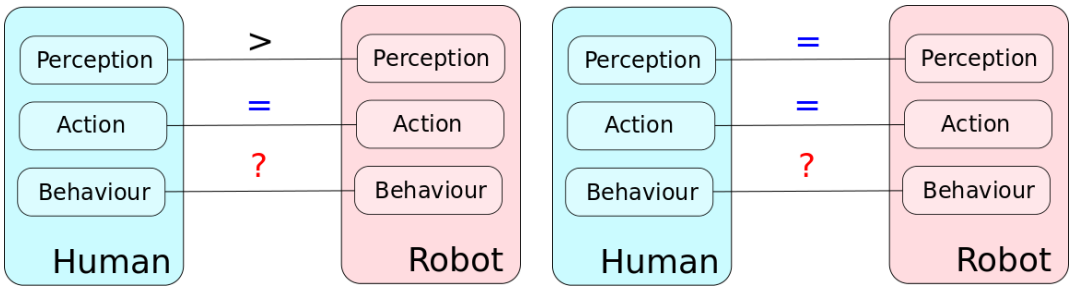


Figure 6.1: Sketch of different possible configurations in strategic human-robot interactions. (a): Approach followed in Chapters 4 and 5, in standard interactions between autonomous and teleoperated robots. Both robots have the same set of actions, however, humans still maintain the advantage of having full visibility of the interaction environment (unlike autonomous robots, which can only view it through a low-resolution camera). (b): The main question addressed in this chapter – when autonomous robots and human operators have not only the same set of actions, but also the same perceptual information (i.e. the operator is constrained to viewing only the robot's camera feed), how do their interactive decisions compare?

In this chapter, we once again consider interactions between autonomous and human-controlled robots in complex, physical environments. However, we introduce the restriction that these domains may be *perceptually constrained*, i.e. that the human operator may not always have full visibility of the interaction environment. With this consideration in mind, we present empirical data addressing the following questions:

- What is the effect of incompleteness and asymmetry of information on human teleoperation performance in interactive robotic tasks?
- Where should the boundary between human control and autonomy lie, and what is the correlation between the effects of perceptual limitations and the strategic

content of interactive tasks?

We view these issues as central not only to understanding the factors that influence interactive decisions, but also to designing mixed robotic systems that can successfully combine the relative merits of human control and autonomy.

In order to address the above questions, we evaluate the performance of several users in two different interactive tasks involving a teleoperated and an identical autonomous NAO humanoid robot. Both tasks share the following properties:

- The human users do not know a priori how the autonomous robot will behave, nor can they exchange any data with it during the task. Thus, they can only infer its decisions through *observation* and *repeated interaction*.

- The tasks are *fully interactive*, requiring users to make several decisions over a short time horizon and also to *respond* to the actions of the autonomous robot.

The first task is a *cooperative* target allocation task, where the two robots must reach two different targets without interfering with each other. The second is the strategic *adversarial* penalty shooting task studied in Chapters 4 and 5 – here we consider the case of an autonomous striker playing against a teleoperated goalkeeper. The penalty shooting task is considerably harder for two *interrelated* reasons:

- The autonomous robot is a ***strategic adversary*** who seeks to outperform the human through deceptive manoeuvres.

- The human subject must estimate and infer finer-grained information, e.g. the absolute states of the robots and the most likely kicking direction selected by the striker.

In both cases, we first evaluate subjects under full observability of the interaction environment, and we subsequently constrain them to viewing only a live feed from the robot's camera.

Main hypothesis: In light of the above constraints, our core hypothesis is that only a small proportion of subjects should perform worse in the cooperative task under restricted perception, whereas a greater fraction would be impacted in the adversarial task under these conditions. In other words, we hypothesise that the *combined challenge* of reasoning about *absolute states* and the *strategic behaviour* of the adversary will have an adverse effect on human performance under limited visibility in the second task, unlike the simpler interaction and inference requirements posed by the first task.

In the remainder of this chapter, we first describe the two interactive tasks, highlighting the challenges for human subject in each case (Section 6.2). We use the same experimental domain as in Chapters 4 and 5, i.e. interactions between autonomous and teleoperated NAO robots (see Section 4.2 for a summary of this setup). In Section 6.3, we present empirical results of our experimental evaluation on several subjects. Our results suggest that restricted visibility is more likely to impact participants in strategic interactions, where there is greater uncertainty over the autonomous adversary. We review the key contributions of this study in Section 6.4.

6.2 Interaction Scenarios

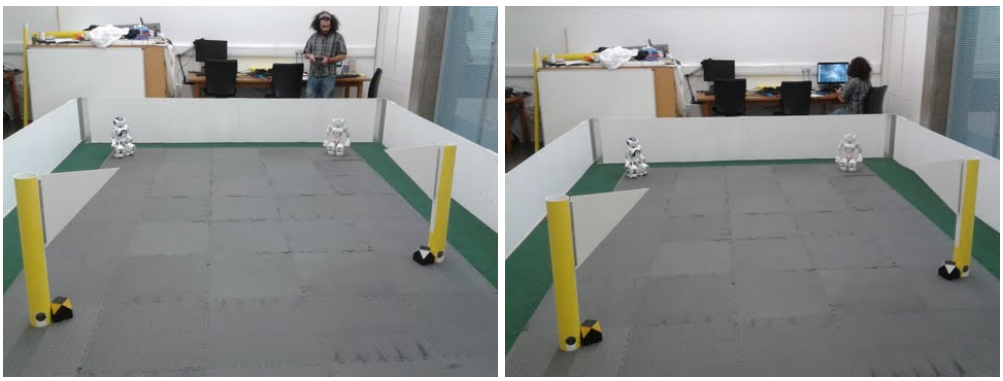
6.2.1 Cooperative task – Target allocation

In the cooperative target allocation task, the two robots are placed in an arena as shown in Figure 6.2(a). The task requires the robots to reach two different targets in the arena. The initial positions of the robots are fixed as in Figure 6.2(a), but the targets are moved around between trials. The autonomous robot initially selects a target at random, and begins moving towards it. The human user must then infer where the autonomous robot is heading, and steer his own robot to the other target as fast as possible. The user must also avoid collisions or interference with the autonomous robot.

The autonomous robot has no external information (e.g. positions from an overhead camera) and perceives the world only through its own perspective camera. There is also no communication between the robots, so there is no prior (or interactive) agreement on the allocation of the targets.

6.2.1.1 Autonomous robot behaviour

The autonomous robot navigates to its chosen target using a simple visual servoing routine. The targets are colour-coded so that they can be easily identified. The random selection of a target is enforced by having the robot initially look away from the arena (so that no targets are visible), and then randomly select whether it should start turning left or right. The robot then keeps turning until it locates a target, and then starts moving towards it.



(a) (b)

(c)

Figure 6.2: Experimental setup - cooperative task. (a): An autonomous (top left) and a teleoperated (top right, in front of human user) robot must reach two different targets (indicated by the flagpoles) without interfering with each other. The autonomous robot randomly selects a target to navigate to, which the user must infer during the interaction, in order to lead his robot to the other target. The initial positions of the robots are fixed but the locations of the targets change between trials. (b): Same task, but the user now has access only to the robot's noisy camera feed (shown in (c)).

6.2.1.2 Full vs. restricted perception

We consider the cooperative navigation task in two different situations. In the first case (Figure 6.2(a)), the user may view the entire arena, thus having full visibility of the environment. In the second case (Figure 6.2(b)), the user is restricted to viewing only the teleoperated robot's noisy camera feed (Figure 6.2(c)) on a computer screen. Thus, the user is constrained to have the same perceptual capabilities as the autonomous robot, so the two robots differ only at the behavioural level (autonomous vs teleoperated).

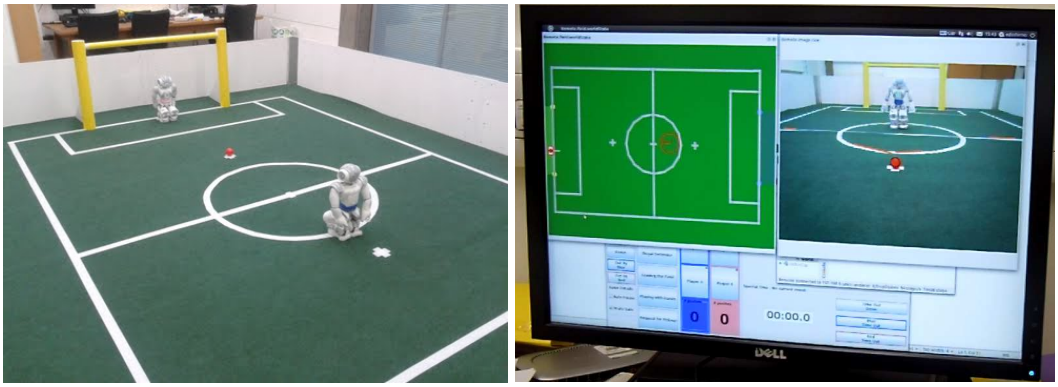
In the full visibility case, users have a clear view of both robots and both targets. Thus, it is relatively straightforward to identify where the autonomous robot is heading, and, assuming adequate familiarity with the joystick controller, lead the teleop-

erated robot to the appropriate destination. However, when perceptual information is restricted, recognising the autonomous robot's behaviour and steering the teleoperated robot becomes more challenging.

6.2.1.3 Teleoperation commands

The user is allowed to control the translational (forward-backward-side steps) and rotational (turn left-right) motion of the robot. In restricted visibility, there are additional inputs to control the robot's head movement and scan different parts of the world through its camera.

6.2.2 Adversarial task – Penalty shooting



(a) (b)

Figure 6.3: Experimental setup - adversarial soccer penalty shooting task. (a): Initial poses of the autonomous striker (near side, blue waistband) and the teleoperated goalkeeper (far side, pink waistband). (b): Restricted perceptual information. Left: Visualisation of the robots' self-localisation estimate (shown by the red markings on the field drawing). Right: Perspective view of the goalkeeper, looking at the ball and the approaching striker.

The adversarial task is the penalty shooting game studied in Chapters 4 and 5. The rules of the game are the same as those described in Section 4.2.5. The objective for the human is to guess which way the autonomous striker is going to shoot, and move the goalkeeper to a suitable shot-blocking position. This is considerably harder than cooperative navigation, as the autonomous robot now attempts to *outperform* the human, by strategically trying to score a goal. Thus, the human must also continuously reason about the absolute positions of the robots in the field.

In contrast to the cooperative task, where only relative distances to the targets are required, the autonomous robot now needs to know its absolute position in the field. To compute this information, the robot uses the particle-filter method for self-localisation described in Section 4.2.3.

6.2.2.1 Autonomous robot behaviour

The autonomous striker was programmed to run the Gaussian Mixture Model-based algorithm introduced in Chapter 4. This algorithm was chosen over the interactive learning algorithm of Chapter 5, in order to have a fixed behaviour against which all human subjects can be compared. In other words, our intention was to isolate any possible artefacts arising from an adaptive behaviour by the autonomous agent, and instead assess the effects of perceptual limitations on interactive human decisions, which is the core focus of our experiment.

6.2.2.2 Full vs. restricted perception

In the restricted visibility case, the user is provided with both the robot's live camera feed and a visualisation of the two robots' self-localisation estimates (Figure 6.3(b)). In the full visibility case, uncertainty in localisation presents the autonomous robot with an even greater perceptual handicap than in the cooperative task, as noisy or incorrect positional information is likely to lead the striker to misinformed decisions on its adversary. For humans, restricted visibility introduces the challenge of inferring the absolute positions of the robots, using only noisy sensory data.

6.2.2.3 Teleoperation commands

As in the cooperative task, users may control the translational and rotational motion of the goalkeeper. There are also inputs for spreading the robot's legs to block the ball. The goalkeeper is programmed to track the ball and the approaching striker automatically (as in Figure 6.3(b)), removing the need to control the robot's head separately.

6.3 Results

We evaluated the two tasks on 40 different subjects; 20 of these subjects were tested just on the adversarial task, 10 just on the cooperative task, and 10 participants on both tasks. The experimental sample was varied, consisting of both male and female

Figure 6.4: Snapshots from cooperative task trials. Navigation targets are now represented as orange balls. *Rows 1-2:* A subject controlling the robot (blue waistband, starting at the right) under full visibility. *Rows 3-4:* A trial as seen through the teleoperated robot's camera. *Rows 5-6:* The same subject controlling the robot under restricted visibility.

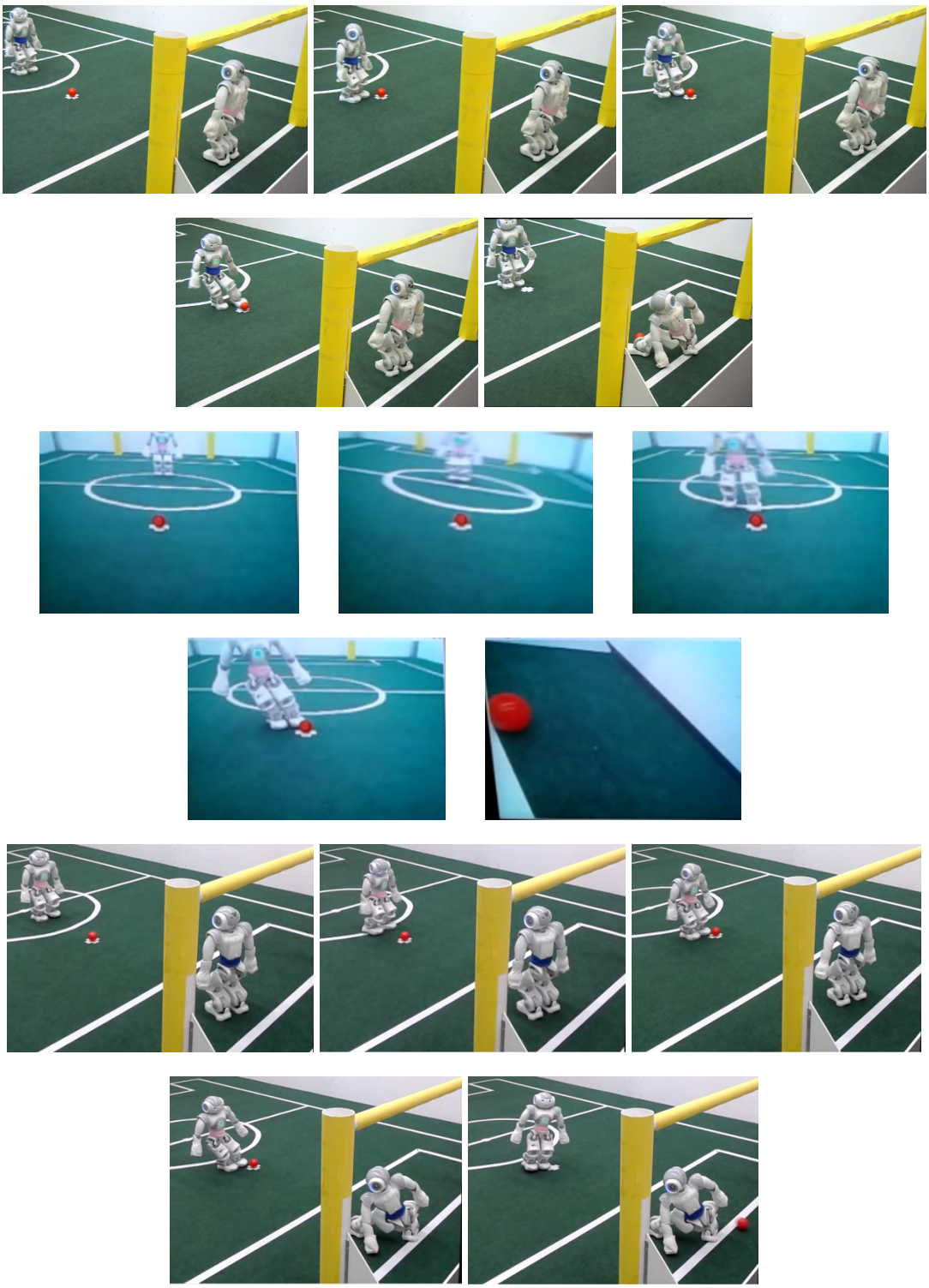


Figure 6.5: Adversarial task snapshots. *Rows 1-2:* Full visibility - a teleoperated goalkeeper (operator not shown) saves a shot. *Rows 3-4:* A trial as seen through the robot's camera. The last snapshot shows the view of the goalkeeper after an unsuccessful dive to save the ball. *Rows 5-6:* Limited visibility - a different subject conceding a goal.

subjects, young children and adults, users with previous robotics experience and users who were interacting with robots for the first time. Snapshots from recorded trials are given in Figures 6.4 and 6.5. Further examples are available in the supporting video of this chapter (Valtazanos, 2012b). The results presented in this section also appear in (Valtazanos and Ramamoorthy, 2013b).

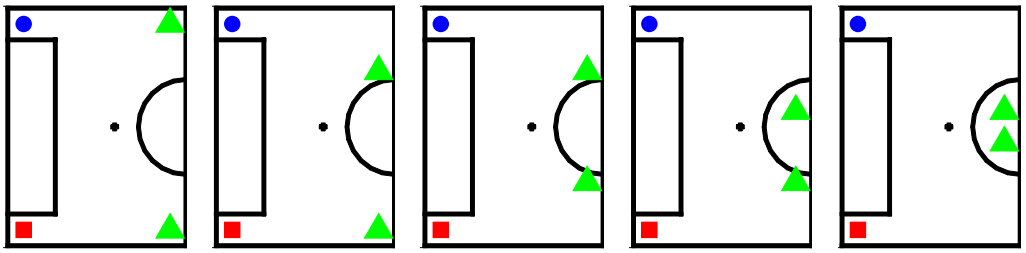


Figure 6.6: The five target configurations, cooperative task. *Blue circle*: Teleoperated robot initial position. *Red square*: Autonomous robot initial position. *Green triangles*: Target positions.

For the target allocation task, each subject was evaluated on 5 different target configurations, which are shown in Figure 6.6. Targets were progressively moved closer to increase the difficulty of the task. Subjects were initially tested on each configuration under full visibility, and then they were asked to repeat this procedure viewing only the robot's camera feed. In each trial, we recorded the targets selected by the robots, the time taken by the teleoperated robot to reach the selected target, whether or not there was a collision with the autonomous robot, and the user's joystick inputs. As target positions were known in each trial, we divided the distance to the selected target with the total time taken by the subject, to obtain the *average speed* as a normalised performance metric.

For penalty shooting, subjects controlled the goalkeeper for 5 trials under full visibility, and then for a further 5 trials under limited visibility. We recorded the outcome of each trial (goal/no goal), the control inputs of the user, and the self-localisation estimates of the two robots during the trial.

6.3.1 Overall performance

6.3.1.1 Performance metrics

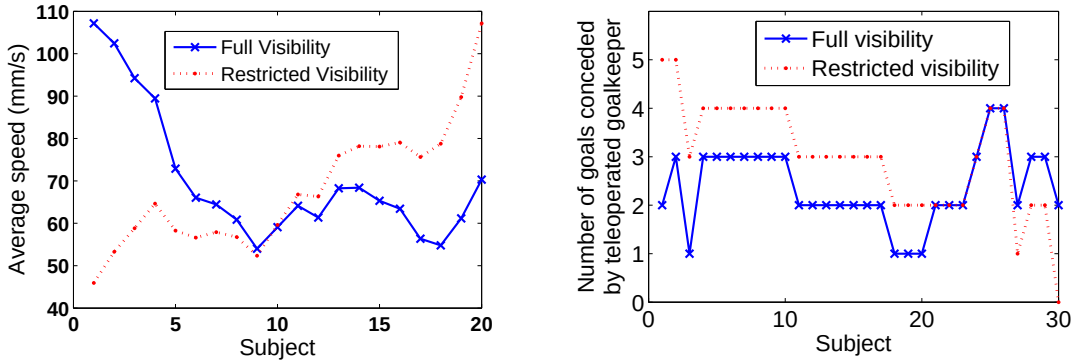
Results for the overall metrics (average speed for target allocation, goals conceded for penalty shooting) are shown in Figure 6.7. For target allocation, there was little differ-

Visibility	Full	Restricted
Mean average speed over all subjects (mm/s)	86.57	76.70
Minimum average speed	38.83	32.31
Maximum average speed	128.22	93.62
Standard deviation of avg. speed	20.05	13.66
Number of collisions with autonomous robot (out of 100 trials)	4	2

(a) Overall performance metrics – cooperative task.

Visibility	Full	Restricted
Total number of goals conceded	71/150	90/150
Mean goals conceded per subject	2.36/5	3/5
Standard deviation	0.81	1.14
Mean goal difference between visib. cases		0.663
Standard deviation		0.994

(b) Overall performance metrics – adversarial task.



(c) Performance metrics per subject - average speed in target allocation (left), goals conceded in penalty game (right). In each graph, values are sorted by the difference of the performance of the subject between full and restricted visibility. Values towards the left represent subjects most affected by restricted visibility, as indicated by the performance degradation.

Figure 6.7: Overall performance metrics – both tasks.

ence between visibility conditions, in both successful execution rate (collisions with autonomous robot) and performance rate (average speed). An interesting pattern is observed in the subject-specific illustration of the results (Figure 6.7(c) - left), where there is a roughly equal number of subjects with improved and deteriorated performance between the two visibility cases. This suggests that reduced visibility is not an impeding factor in this simple interactive setting.

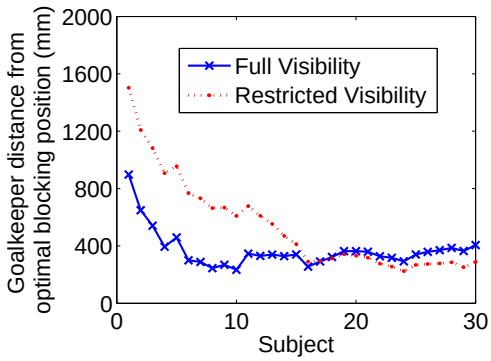


Figure 6.8: Alternative performance metric for adversarial task: distance from optimal blocking position (see Figure 4.8 for an explanation of the metric). Results per subject, sorted by difference between visibility cases.

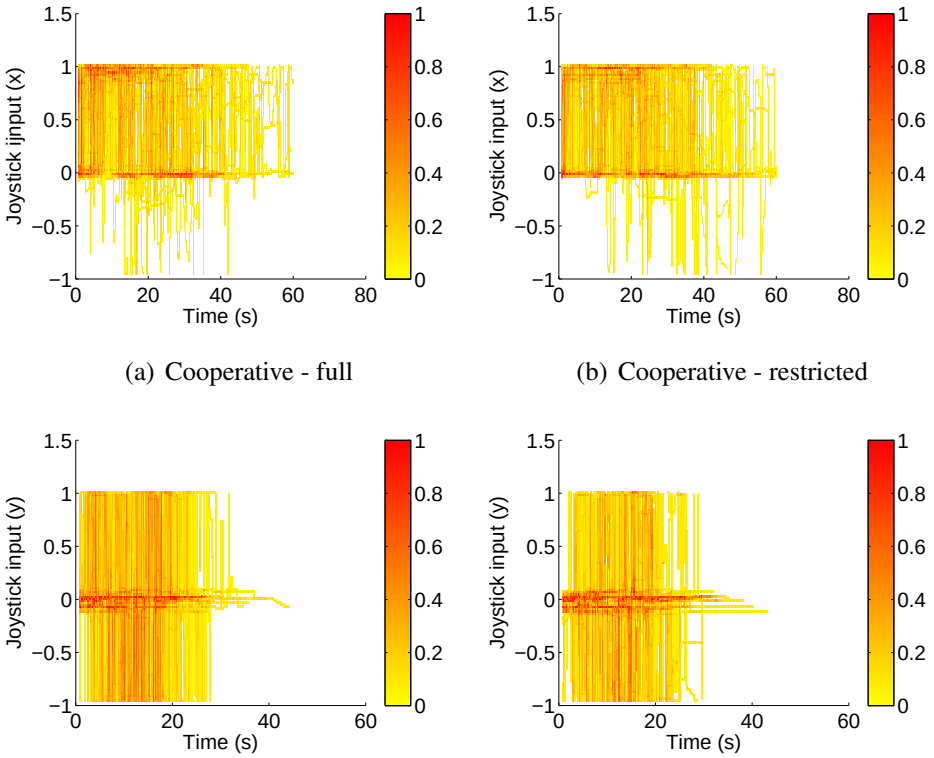
By contrast, most subjects appeared to struggle more under restricted visibility in the adversarial task. About two thirds of the subjects saved fewer goals when this restriction was applied, while only 4 out of 30 managed to save more (Figure 6.7(c)). For this task, we also recorded a different performance metric, the distance of the goalkeeper from the optimal blocking position at the time of the shot (see Figure 4.8 for a more detailed explanation). Through this metric, we model how well users were able to respond to the moves of the autonomous striker, and lead goalkeepers to a position that maximises the chances of a save. As shown in Figure 6.8, the recorded distance for almost half of the subjects increased considerably under restricted visibility.

6.3.1.2 Performance rate

Trial	**1**	**2**	**3**	**4**	**5**
Average speed (full visibility)	69.5	94.3	82.4	89.1	97.4
Average speed (restricted visibility)	67.6	76.1	77.9	78.1	83.8
Goals conceded (full visibility)	0.27	0.67	0.40	0.53	0.50
Goals conceded (restricted visibility)	0.60	0.80	0.40	0.50	0.70

Table 6.1: Time-indexed representation of overall results – mean values per trial

Table 6.1 shows a time-indexed representation of the overall results for the different presented experiments. Due to the small number of trials and the short duration of each trial (at most 1 minute for both tasks), subjects appear not to be affected by factors such as fatigue or stress, which could cause a visible performance degradation in longer



(a) Cooperative - full

(b) Cooperative - restricted

(c) Adversarial - full

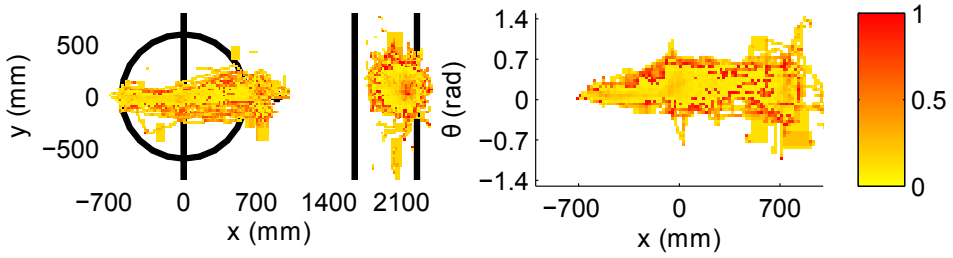
(d) Adversarial - restricted

Figure 6.9: Heat maps of recorded user inputs, all trials. Colour indicates the percentage of trials in which a particular control input/time pair was recorded. *Top row:* Cooperative task - forward motion (positive direction: front). *Bottom row:* Adversarial task - side motion (positive direction: left).

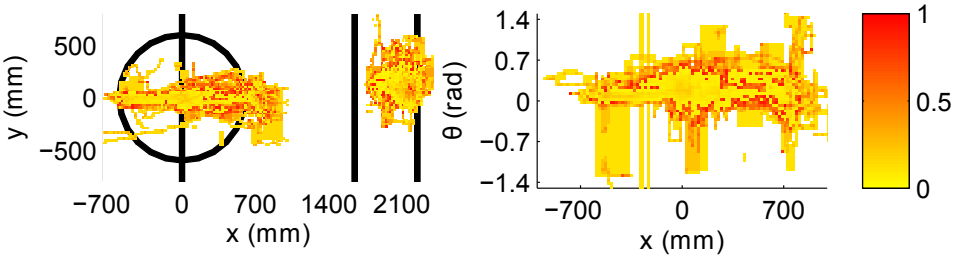
experiments. In the restricted visibility instance of target allocation, subjects are seen to improve their performance over time, without however reaching the average speeds attained in the full visibility case. By contrast, there is no conclusive evidence of time-induced learning in the other experiments, with the mean performance fluctuating across different trials.

6.3.2 User control inputs and trajectories

In addition to evaluating overall performance, we compared the variation of user control inputs under the different experimental conditions. Figure 6.9 provides a heat map representation of all recorded inputs for the two most frequently used axes of motion in the two tasks – the forward motion in target allocation and the goalkeeper's side motion in penalty shooting. In the cooperative task (Figures 6.9(a)-6.9(b)), we again



(a) Full visibility.



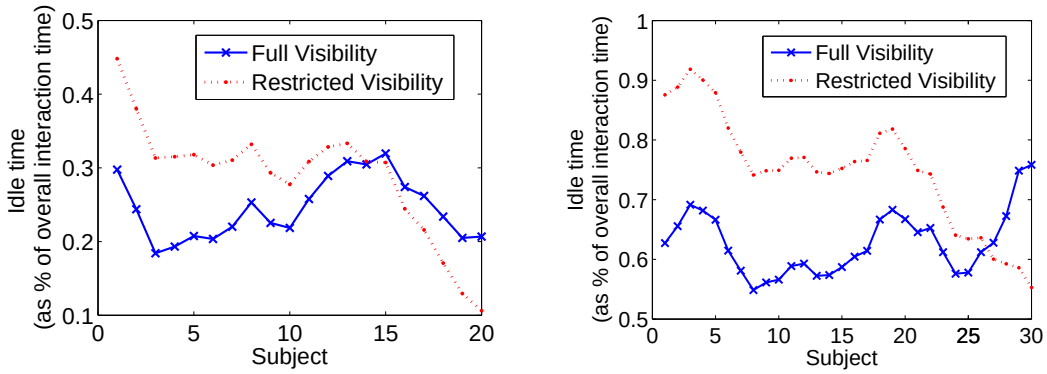
(b) Restricted visibility.

Figure 6.10: Heat maps of recorded striker and goalkeeper trajectories, all trials. Colour indicates the percentage of trials in which a particular point was recorded. *Left subplots:* heat maps for forward (x) - side (y) motion trajectory components - left blob corresponds to autonomous striker, right blob to teleoperated goalkeeper trajectories. *Right subplots:* heat maps for forward (x) - rotational (θ) motion components for the striker.

observe little variation between full and restricted visibility. However, in the adversarial task (Figures 6.9(c)-6.9(d)), the intensity of commanded motion is stronger in the full visibility case.

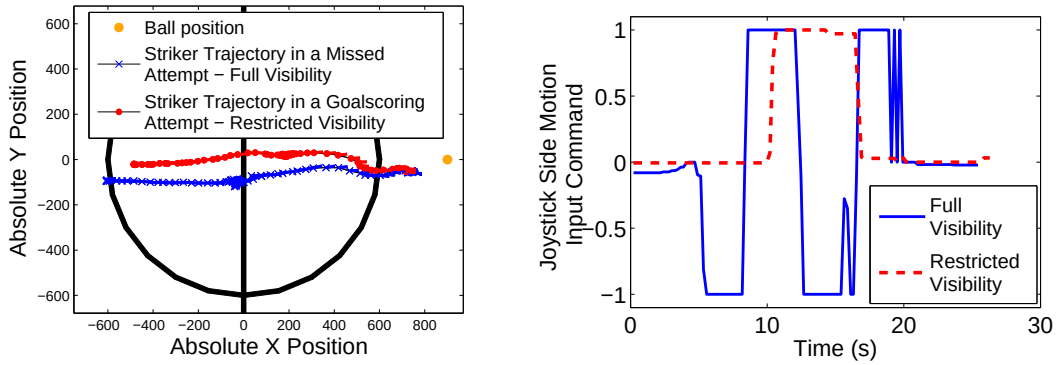
To further quantify this discrepancy, Figure 6.10 shows heat maps for all striker and goalkeeper trajectories in the adversarial task. It can be seen that although the trajectories of the autonomous striker are similar in both cases, goalkeepers move towards the edges of the goal less frequently in the second case. This partly explains the higher number of goals conceded by teleoperated robots under restricted visibility.

Moreover, we looked at how control inputs varied between tasks on a subject-to-subject basis. To this end, we measured the average *idle time*, i.e. the percentage of time during which no command was input by a subject (Figure 6.11). Idle time is considerably higher in penalty shooting, where subjects spend more time observing the autonomous robot's approach before they move their own robot. However, we also note that both the percentage of subjects whose idle time increases when visibility is restricted, as well as the average rate of this increase, are considerably higher in the



(a) Cooperative task. (b) Adversarial task.

Figure 6.11: Idle times per subject. The idle time is the percentage of the overall time during which no command was sent from the user to the robot.



(a) Striker trajectories. (b) Goalkeeper control inputs. Positive direction: left.

Figure 6.12: Effects of idle time on performance of a specific subject. (a): Two similar trajectories by the striker against this subject, one per visibility case. Only the full visibility attempt was saved by the teleoperated goalkeeper. (b): Illustration of the variation of the subject's side motion between these trials.

adversarial task.

Restricted visibility was also found to impact the *response time* of subjects in the adversarial task. To illustrate this effect, Figure 6.12(a) two similar autonomous striker trajectories (one for each visibility case) against a subject, and the corresponding user inputs. Although the trajectories are similar, only the full visibility one was saved by the teleoperated goalkeeper. As seen in Figure 6.12(b), this discrepancy is partly explained by the more delayed response in the restricted visibility case, where the subject needs more time to make sense of the interaction.

6.3.3 Statistical significance

6.3.3.1 Main hypothesis

In order to assess the statistical significance of our overall results, we tested for the *contradiction* of the main experimental hypothesis as stated in Section 6.1. In other words, our null hypotheses are that a worse performance would be observed for a majority of subjects (greater than 75%) in the simple cooperative task, and for a minority (less than 25%) of subjects in the more complex strategic task. To assess these null hypotheses, we conducted a t-test for the overall performance indicators – the average speed in target allocation, and the number of goals conceded in the penalty game. We measured the percentage of subjects for which performance deteriorated in each case, and computed

$$t = \frac{\bar{x} - \mu_0}{s} \cdot \sqrt{n}, \tag{6.1}$$

where $\bar{x}$ is the sampled percentage, μ_0 is the hypothesised percentage (75% in the cooperative task, 25% in the adversarial one), s is the sample standard deviation, and n is the sample size. Based on a two-tailed t-test for the two null hypotheses, we obtain p-values of 0.013 and 0.03, respectively. So, at a 5% significance level, we reject the null hypotheses, and conclude that limited perception does not have a significant impact in the cooperative task, while having a non-negligible effect in the adversarial one which features more severe constraints on perception and action – this was our original main hypothesis.

6.3.3.2 Explaining factors

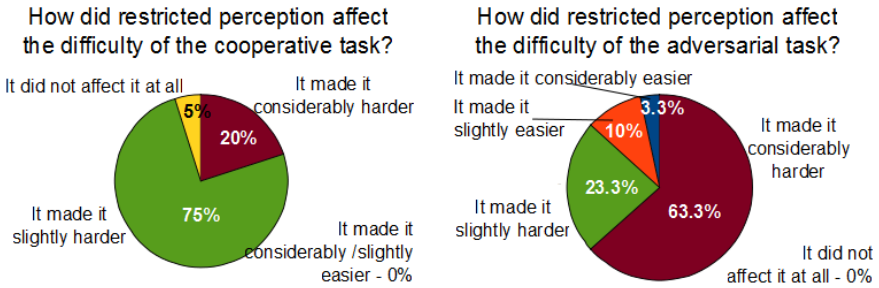
The difficulty in the adversarial task lies in the determination of the autonomous adversary's strategy, and the estimation of the absolute states of the interacting robots. As these challenges are *not* independent of each other, they cannot be explicitly decoupled in order to assess their individual contribution to the overall difficulty of the task. However, we can assess the correlation between the two secondary metrics, the idle time and the distance from the optimal position, and the overall subject performance. Our hypothesis is that increases in each metric between visibility cases are linked to performance degradation, i.e. that the overall mean goal difference (Table 6.7(b)), and the corresponding difference for subjects impacted by each metric should be comparable (i.e. not differ by more than 1 goal, which is approximately equal to the computed standard deviation).

For each metric, we conducted a two-sample pooled t-test to determine its effect on the overall performance degradation. In particular, we measured statistics for the sets of subjects for which idle time/distance from optimal position increased under limited perception, and computed

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - d_0}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, \quad s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2} \tag{6.2}$$

where $\bar{x}_1, s_1, n_1$ are the overall mean goal difference, standard deviation, and sample size, $\bar{x}_2, s_2, n_2$ are the corresponding values for the subset of subjects for which the value of the metric increased, and d_0 is the hypothesised mean difference. We tested for the contradictory null hypotheses that the two metrics cannot be used to explain performance degradation, i.e. that the difference between each $\bar{x}_2$ and the overall mean $\bar{x}_1$ will be more than 1 goal. For these tests, we obtained p-values of 0.010 for idle times and 0.008 for optimal positions. At a 5% significance level, we reject the null hypotheses, and conclude that an increase in the idle time or the distance from the optimal position is likely to be matched with an increase in the number of conceded goals under restricted visibility.

6.3.4 User experiences



(a) Cooperative task. (b) Adversarial task.

Figure 6.13: User experiences on restricted visibility.

After each experiment, we asked subjects to give us their opinion on the impact of restricted visibility on their behaviour (Figure 6.13). In both tasks, most subjects stated that limited perception impacted their performance. However, the dominant response in the first case was that restricted visibility made the task only "slightly harder", whereas most users found the adversarial task "considerably harder". Another interesting result was that some subjects found the penalty game easier under limited

perception, with one subject labelling it "considerably easier"; not surprisingly, this is the (only) subject who in Figure 6.7(c)-right saved all 5 shots under restricted visibility.

6.4 Conclusions

Our experimental analysis suggests that limited visibility is more likely to affect teleoperation performance in challenging, adversarial tasks, which require continuous inference of the absolute state and strategy of an interacting robot. Furthermore, restricted perception appears to affect the ability of humans to (inter)act *strategically*, with several subjects being deceived by the autonomous adversarial robot more easily. By contrast, when the task is not particularly challenging and requires only very basic modeling of robot states and strategies, most subjects are likely to be unaffected by this restriction.

Mixed robotic environments are becoming increasingly important in human-robot interaction, as several applications demand an interplay between autonomous and teleoperated agents in complex physical settings. In many such domains (e.g. rescue robotics), perceptual information is inherently limited, so it is important to identify situations where humans might fall short in teleoperating a robot, and how autonomous robots can compensate for these weaknesses. In this respect, the work presented in this chapter contributes an empirical evaluation which highlights interaction scenarios and visibility conditions where human control is likely to be problematic. Our experiment also informs decisions about when to assist human decision makers in teleoperation, and how to structure the balance between human command and robot autonomy. We believe that our methodology can be applied in the design of *mixed robotic teams*, where there is a need to empirically determine both the optimal composition (how many autonomous/how many teleoperated?) of a team, and the roles (what should each autonomous/teleoperated robot do?) of its constituent members.

One remaining open question in the strategic environments we consider is how can humans be directly introduced in the interaction loop (instead of being just operators of humanoid robots). In Chapter 7, we propose an algorithm through which inertial sensing and optical motion capture systems can be jointly used to learn a motion model for human subjects. We discuss how this model can be used in direct strategic human-robot interactions, by continuously providing autonomous robots with rich information on the state of interacting human partners.

Chapter 7

Towards direct strategic human-robot interactions

7.1 Overview

The algorithms and experiments presented so far in this thesis are primarily concerned with interactions between autonomous and teleoperated robots. One reason behind this choice is that we want to directly compare and contrast human and robot decision making, while minimising the influence of external factors, e.g. the fact that humans can walk faster and see better than most robots. However, direct human-robot interactions are also challenging because it is difficult to provide robots continuously and reliable with information on the *state* of the interacting human partner(s).

In several domains and applications, robots need to know both the absolute *position* of these interacting subjects, as well as finer-grained information on their body *posture*, such as arm movements or gait patterns. Furthermore, it is often required that motion be captured in challenging, *unconstrained* environments, for example, a home or an office spanning a large area with multiple rooms and corridors, or an open outdoor environment. As an illustrative example, consider an interaction between a human and a personal robot in a domestic environment, where the two sides need to collaborate in order to prepare a meal. The robot will need to know both where the human is (e.g. is the subject in the kitchen, or has he/she moved to a different room to retrieve a required object/ingredient?), as well as what type of motion that subject is performing (e.g. is the subject using a utensil, opening the fridge, or simply waiting idly for the robot to do something?). Such information is important if the robot is to make repeated, robust interactive decisions, and potentially shape and influence the behaviour of the human

participant during the interaction.

Despite recent advances in motion capture and sensing technologies, fulfilling all the above requirements in a robust manner is a challenging task. For example, optical motion capture systems can retrieve both positional and postural data, but only within contained environments limited to a small volume of capture. Inertial sensing systems allow for greater flexibility in the capture environment, as they are not restricted by line-of-sight constraints between the sensing devices and the tracked subject. However, they do so at the expense of not yielding absolute positions, as their calculations are based on relative sensory estimates, e.g. gyroscope and accelerometer readings. Similarly, global positioning system (GPS) sensors can compute absolute spatial positions, but at a coarse level of precision and without supplying postural information, while also having limited applicability in indoor environments. Other motion capture technologies, e.g. magnetic systems, also suffer from one or multiple of the above limitations.

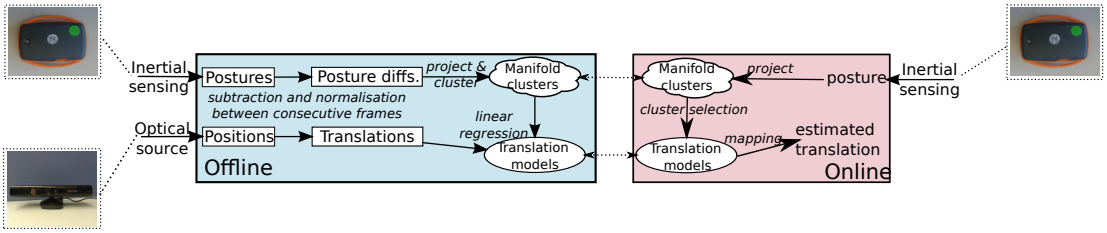


Figure 7.1: Overall structure of the proposed system. An inertial sensing and an optical source are synchronised and jointly used to learn generative models of whole-body translations in an offline phase. These translations are encoded as linear regression-based mappings from projected latent representations of *posture differences*, as detected from the inertial source, and *positional variations*, as detected from the optical source. Online, the optical source is removed, and the learned model is used to predict local translations for the tracked subject.

In order to address the challenges of simultaneous posture and position capture in unconstrained environments, one would need to combine the relative strengths of these heterogeneous systems in a principled manner. In this chapter, we propose a hybrid position and posture tracking algorithm (Figure 7.1), which jointly uses an inertial and an optical motion capture system to learn local *models of translation* for a given human subject. The algorithm consists of an *offline* learning and *online* generation phase. In the offline phase, body posture data collected from the inertial sensors are synchronised with position data captured from the optical source and aggregated into a single dataset.

Due to their high dimensionality, the posture data are projected and clustered on a low-dimensional *manifold*, which captures the salient kinematic structure of the dataset. For each cluster, the projected data are used to learn a mapping from local *posture differences* to *whole-body translations* through linear regression. In the online phase of the algorithm, the optical source is removed, and the learned models are used instead to generate translation vectors from estimated posture differences. By iteratively applying these translations, the proposed system can track both the position and the posture of a subject, thus overcoming the main limitation of inertial systems discussed above. Moreover, due to the removal of the optical source in the online phase, the system is not affected by the morphology or area of the capture environment.

In the remainder of this chapter, we first describe our method for hybrid posture and position tracking, distinguishing between the translation learning and generation phases (Section 7.2). In Section 7.3, our approach is evaluated both in simulations and experiments, on a selection of systems and motions; first, on data from the Carnegie Mellon Motion Capture Database (`http://mocap.cs.cmu.edu`), which are annotated with ground-truth absolute positions, and then, on a physical motion capture environment, where we use the Microsoft Kinect (Figure 2.6(a)) as an optical source and the Orient platform (Figure 2.6(b)) as the inertial system. In both cases, our algorithm is shown to yield a lower overall position error than the related established tracking method of acceleration integration in a wide range of motions. Furthermore, in the physical motion capture case, we demonstrate examples of successful position tracking in a challenging office environment, in which existing motion capture technologies cannot be applied in isolation. We conclude by reviewing possible extensions to our work, and discussing its potential applications in strategic human-robot interactions (Section 7.4). The methodology and results presented in this chapter also appear in (Valtazanos et al., 2013b).

7.2 Method

This section discusses our method for learning translation models from synchronised optical and inertial sensing data. We begin by outlining the output produced by each tracking source, and then explain the details of our mathematical model.

7.2.1 Sensory device outputs

7.2.1.1 Kinect

Figure 7.2: Body contour tracking using the Kinect. The tracking software automatically detects the outline of a human body, and tracks it as a cloud of points (shown as a blue blob).

We use the OpenNI body tracking interface (`http://www.openni.org`) to detect and track the position of human subjects (Figure 7.2). The software automatically detects the outline of a human body, and tracks it as a collection of N_I image point coordinates, $\vec{B} = \{(x_1, y_1), \ldots, (x_{N_I}, y_{N_I})\}$. The absolute position of the tracked body is approximated as the centroid of these points as computed through *image moments*.

Let W, H be the width and height (in pixels) of the image captured by the cameras, and let $\mathbf{I}$ be a two-dimensional array, such that

$$\mathbf{I}(a,b) = \begin{cases} 1, & (x_a, y_b) \in \vec{B} \\ 0, & (x_a, y_b) \notin \vec{B} \end{cases}, \tag{7.1}$$

where $1 \leq a \leq W$, $1 \leq b \leq H$. The raw image moments, M_{ij} are defined as

$$M_{ij} = \sum_{a=1}^{W} \sum_{b=1}^{H} x_a^i \cdot y_b^j \cdot \mathbf{I}(a,b). \tag{7.2}$$

Based on these definitions, the image coordinates of the centroid of the tracked body, $C \doteq (\bar{x}, \bar{y})$ are given by

$$C = (\,\mathtt{round}(M_{10}/M_{00}), \mathtt{round}(M_{01}/M_{00})\,). \tag{7.3}$$

The depth of each image pixel (with respect to the device) is measured by the range-finding sensor of the Kinect. This information is used to convert the computed image

centroid, $(\bar{x},\bar{y})$, to the centroid of the body surface that is visible to the Kinect. These coordinates approximate to the absolute positional coordinates of the tracked body,

$$p = (x^B, y^B, z^B). \tag{7.4}$$

7.2.1.2 Orient inertial measurement units

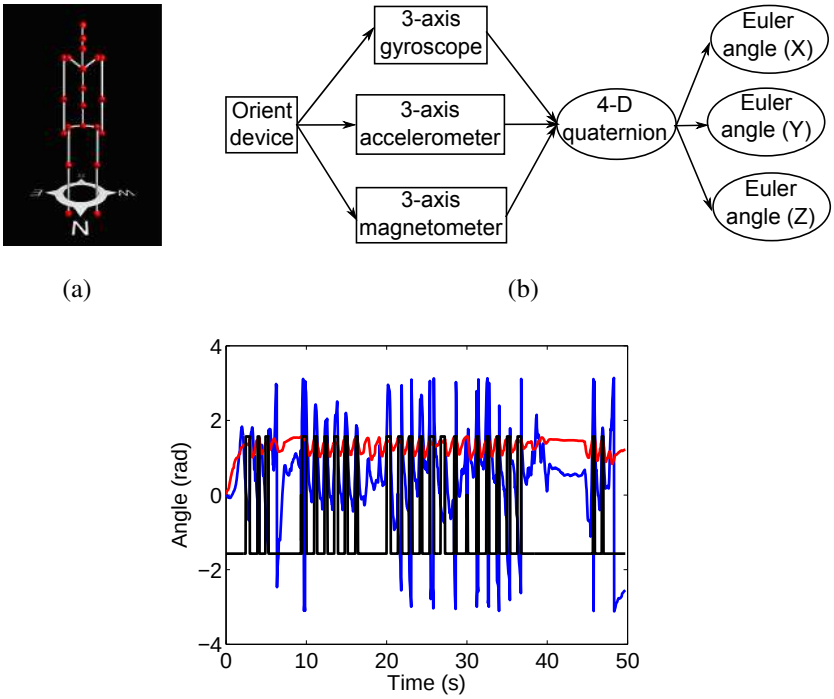


Figure 7.3: Posture estimation using the Orient inertial measurement units. (a): 3-D model of the tracked body – each device is placed and mapped onto a limb of this model (represented by white lines). (b): Orientation estimation procedure – raw data from a device's sensors computes a quaternion representing the orientation of a limb, which is in turn converted to three-dimensional Euler angles. (c): Example of Euler angles produced by one Orient device over a 50-second capture.

The posture of the tracked subject is computed by the Orient devices. Each device is placed on a limb of the subject's body (Figure 7.3(a)). The raw data from the device's sensors (triaxial gyroscope, accelerometer, and magnetometer) computes a quaternion representing the orientation at that limb, which is in turn converted into three-dimensional Euler angles (Figure 7.3(b)) based on a pre-specified rotation order. Due to this convention, an Euler angle can represent the orientation more succinctly

than the corresponding quaternion, thus reducing the size of our feature set. An example of the angles output by an Orient is given in Figure 7.3(c). By aggregating the angles computed by all devices placed on the subject's body, we obtain the ***posture vector***

$$\pi = \{(\theta_1^x, \theta_1^y, \theta_1^z), \ldots, (\theta_{N_D}^x, \theta_{N_D}^y, \theta_{N_D}^z)\}, \tag{7.5}$$

where N_D is the number of deployed units, and $(\theta_i^x, \theta_i^y, \theta_i^z)$ are the angles computed by the i-th unit.

7.2.2 Learning translation manifolds

7.2.2.1 Offline learning phase

In the offline learning phase, Kinect positions are time-synchronised with data from the Orient inertial measurement units. From this hybrid data, a mapping from ***posture variations***, as computed by the Orient devices, to ***translations***, as computed by the Kinect, is learned through local linear regression.

Let $\{(p_1, \pi_1, t_1), \ldots, (p_{\tau+1}, \pi_{\tau+1}, t_{\tau+1})\}$ be a set of recorded synchronised training data, comprising $(\tau+1)$ absolute position and posture pairs, along with the times t at which each pair was recorded. By taking the difference of successive instances, we obtain a training data set of τ *unnormalised* translations (i.e. position differences), posture variations[1], and time differences,

$$\begin{aligned}\widetilde{\mathbf{D}} = \{(\widetilde{dp}_1, \widetilde{d\pi}_1, dt_1), \ldots, (\widetilde{dp}_\tau, \widetilde{d\pi}_\tau, dt_\tau)\} = \{(p_2 - p_1,\\ \pi_2 - \pi_1, t_2 - t_1), \ldots, (p_{\tau+1} - p_\tau, \pi_{\tau+1} - \pi_\tau, t_{\tau+1} - t_\tau)\}.\end{aligned} \tag{7.6}$$

At this stage, translations $\widetilde{dp} = (\widetilde{dx}, \widetilde{dy}, \widetilde{dz})$ do not account for the absolute orientation of the subject's body. To address this problem, we assume that at least one inertial measurement unit, $\bar{u}$, is placed on a point where it can measure the subject's absolute orientation, $\bar{\theta}$, with respect to the transverse plane of motion. We focus on this single angle (instead of computing three-dimensional absolute orientations) because it is closely correlated with most turning movements that occur during walking motion sequences. Thus, by normalising with respect to $\bar{\theta}$, we can compensate for turns and changes of direction in the motion of the subject. We take $\bar{u}$ to be the device placed on the subject's waist or hips as a represantative location for this purpose. The required angle $\bar{\theta}$ is computed through $\bar{u}$'s magnetometers, which measure absolute orientations

[1] Angle differences are constrained to lie in $[-\pi, +pi)$.

using Earth's magnetic field. Thus, the unnormalised translation components on the transverse plane, $(\widetilde{dx}, \widetilde{dy})$, can be normalised through the rotation

$$\begin{pmatrix} \widetilde{dx} \\ \widetilde{dy} \end{pmatrix} = \begin{pmatrix} \cos(-\bar{\theta}) & -\sin(-\bar{\theta}) \\ \sin(-\bar{\theta}) & \cos(-\bar{\theta}) \end{pmatrix} \cdot \begin{pmatrix} \widetilde{dx} \\ \widetilde{dy} \end{pmatrix}. \tag{7.7}$$

Furthermore, we normalise translations and posture differences with respect to their recorded time intervals, so that they represent uniform-time variations. Thus, the *normalised* training data set is given by

$$\begin{gathered} \mathbf{D} = \{(dp_1, d\pi_1), \ldots, (dp_\tau, d\pi_\tau)\} = \\ \{(\widetilde{dp}_1/dt_1, \widetilde{d\pi}_1/dt_1), \ldots, (\widetilde{dp}_\tau/dt_\tau, \widetilde{d\pi}_\tau/dt_\tau)\}. \end{gathered} \tag{7.8}$$

-Dimensionality reduction: The size of each posture variation vector, $d\pi$, is $D = 3 \cdot N_D$, where N_D is the number of deployed devices. Even if N_D is not particularly large, it may be difficult to learn a direct mapping from posture variations to translations, due to the different modalities of the posture data. To overcome this problem, we project posture variation vectors to a *latent space*, from which a mapping can be learned more efficiently. We use Principal Component Analysis (PCA), which embeds data into a low-dimensional *linear manifold* by maximising their variance (Bishop, 2006). Thus, this method aims to preserve the high-dimensional structure of the data points in the projected latent space. We summarise the key features of PCA below.

Let $\{d\pi_i\}$, $1 \leq i \leq \tau$ be the set of posture variation vectors, each having dimensionality D. The mean, $\overline{d\pi}$, and covariance matrix, $\mathbf{S}$, of these vectors are given by

$$\overline{d\pi} = \frac{1}{\tau} \sum_{i=1}^{\tau} d\pi_i, \tag{7.9}$$

$$\mathbf{S} = \frac{1}{\tau} \sum_{i=1}^{\tau} (d\pi_i - \overline{d\pi})(d\pi_i - \overline{d\pi})^T, \tag{7.10}$$

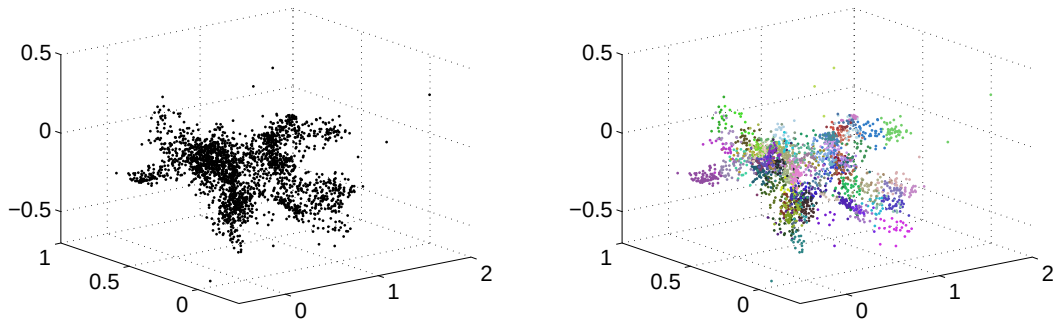
respectively. Now let d be the target dimensionality of the low-dimensional latent space, where $d < D$. We obtain the d *eigenvectors* (or principal components) of $\mathbf{S}$, $u_1, \ldots, u_d$, each of dimensionality D, corresponding to the d largest eigenvalues, $\lambda_1, \ldots, \lambda_d$ of this matrix. These vectors are set as the columns of a $D \times d$ matrix

$$\mathbf{M} = \begin{pmatrix} u_{1,1} & \cdots & u_{d,1} \\ \vdots & \ddots & \vdots \\ u_{1,D} & \cdots & u_{d,D} \end{pmatrix} \tag{7.11}$$

The latent representation of a D-dimensional posture variation vector $d\pi$ is given by

$$\phi = d\pi \cdot \mathbf{M}. \tag{7.12}$$

We refer to the manifold projections ϕ as the *feature vectors* of our translation learning algorithm. In both simulated and physical experiments (Section 7.3), we set the target subspace dimensionality to $d = 3$.



(a) (b)

Figure 7.4: Feature vector clustering example. (a): Projected feature vector points. (b): Division of the points in (a) in 100 distinct clusters, each represented by a different colour.

-Feature vector clustering: In a given set of training examples, there may be groups of similar posture variations leading to related translation vectors. To exploit this similarity, we group the projected feature vectors into clusters of related data points, and learn a separate translation mapping for each cluster (instead of a single mapping for the whole dataset). We use the k-means clustering algorithm, which groups input points into a specified number of k distinct clusters (Bishop, 2006). As we use clustering in a learning context, we use small (with respect to the size of the dataset) values of k to avoid overfitting the training data; in our experiments, this value does not exceed 2% of the overall number of training points. Despite the need for this manually specified parameter, k-means clustering has the advantage that it does not make assumptions about cluster structure (whereas distribution methods such as expectation-maximisation (Bishop, 2006) assume a Gaussian form), while also favouring clusters of approximately equal sizes.

Figure 7.4 illustrates an example of clustering on a set of three-dimensional points. When applied on a dataset of τ feature vectors, the algorithm returns the set of clusters $\mathbf{C} = \{c_1, \ldots, c_k\}$, with centres $\vec{\mu} = \{c_1.\mu_1, \ldots, c_k.\mu_k\}$, where each cluster c_i, $1 \leq i \leq k$, consists of a set of N_i feature vectors

$$c_i = \{\phi_1^i, \ldots, \phi_{N_i}^i\} \tag{7.13}$$

of size N_i.

-Translation mapping learning: For each data cluster $c_i = \{\phi^i_1, \ldots, \phi^i_{N_i}\}$, we learn a mapping from its constituent feature vectors to their corresponding measured translations, $T^c = \{dp^i_1, \ldots, dp^i_{N_i}\}$. We learn a separate mapping for each direction of motion (x, y, z) through *linear regression* on the training points. In other words, we represent each translation component as a *linear function* of the projected feature vectors.

To learn these mappings, we collect all feature vectors of a cluster as a $N_i \times d$ *design matrix*,

$$\mathbf{X} = \begin{pmatrix} \phi^i_{1,1} & \cdots & \phi^i_{1,d} \\ \vdots & \ddots & \vdots \\ \phi^i_{N_i,1} & \cdots & \phi^i_{N_i,d} \end{pmatrix}. \tag{7.14}$$

Furthermore, we define three *observation vectors*, one for each of the directions of motion, such that

$$\mathbf{x} = \begin{pmatrix} dx^i_1 \\ \vdots \\ dx^i_{N_i} \end{pmatrix}, \mathbf{y} = \begin{pmatrix} dy^i_1 \\ \vdots \\ dy^i_{N_i} \end{pmatrix}, \mathbf{z} = \begin{pmatrix} dz^i_1 \\ \vdots \\ dz^i_{N_i} \end{pmatrix}. \tag{7.15}$$

For each observation vector $\mathbf{v}$, we learn a linear mapping from the design matrix $\mathbf{X}$ using least squares approximation. This mapping is represented by a set of d weights $\mathbf{w}$, computed as

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{v}. \tag{7.16}$$

By applying this procedure to all three observation vectors, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, we obtain the linear mapping weights for cluster c_i, $\mathbf{w}^i_x, \mathbf{w}^i_y, \mathbf{w}^i_z$, respectively. These can be collectively represented as the *cluster translation mapping*

$$\mathbf{W}^i = \begin{pmatrix} \mathbf{w}^i_{x,1} & \cdots & \mathbf{w}^i_{x,d} \\ \mathbf{w}^i_{y,1} & \ddots & \mathbf{w}^i_{y,d} \\ \mathbf{w}^i_{z,1} & \cdots & \mathbf{w}^i_{z,d} \end{pmatrix}, \tag{7.17}$$

with a different $\mathbf{W}^i$ computed for each cluster. Thus, the latent space becomes a ***translation manifold*** that can be used to generate translations from given feature vectors.

7.2.2.2 Online translation generation

Learned translation mappings can be applied to novel instances of projected posture variation vectors to predict whole-body translations. Assuming a known estimate of a

tracked subject's initial position, (x_0, y_0, z_0) and orientation, θ_0, the predicted translations can be chained together to track absolute positions over time.

Let $\breve{d\pi}_t$ be the subject's estimated posture variation at time t, and let $\bar{\theta}_t$ be the subject's absolute orientation at that time. Furthermore, let dt_t be the length of the time interval over which $\breve{d\pi}_t$ was recorded. The projection of $\breve{d\pi}_t$ on the translation manifold, $\breve{\phi}_t$, is computed as $\breve{\phi}_t = \breve{d\pi}_t \cdot \mathbf{M}$, where $\mathbf{M}$ is the learned projection mapping from the high-dimensional to the latent low-dimensional space. The cluster nearest to $\breve{\phi}_t$ is given by

$$c^* = \arg\min_{c_i \in \mathbf{C}} \delta(\breve{\phi}_t, c_i.\mu_i) \tag{7.18}$$

where $\delta(\cdot,\cdot)$ is the Euclidean distance between two points. Then, if $\mathbf{W}^*$ is the cluster translation mapping for c^*, our model predicts a normalised translation for $\breve{\phi}_t$ as

$$\widehat{dp_t} \doteq (\widehat{dx}, \widehat{dy}, \widehat{dz}) = \mathbf{W}^* \cdot \breve{\phi}_t^T \tag{7.19}$$

The updated predicted position at time t, $\tilde{x}_t, \tilde{y}_t, \tilde{z}_t$, is obtained by applying the orientation $\bar{\theta}_t$ to $\widehat{dp_t}$, scaling it by dt_t to reflect the length of the current time interval, and adding it to the previously estimated position, $(\tilde{x}_{t-1}, \tilde{y}_{t-1}, \tilde{z}_{t-1})$. In other words,

$$\begin{pmatrix} \tilde{x}_t \\ \tilde{y}_t \\ \tilde{z}_t \end{pmatrix} = \begin{pmatrix} \tilde{x}_{t-1} \\ \tilde{y}_{t-1} \\ \tilde{z}_{t-1} \end{pmatrix} + dt_t \begin{pmatrix} \cos(\bar{\theta}_t) & -\sin(\bar{\theta}_t) & 0 \\ \sin(\bar{\theta}_t) & \cos(\bar{\theta}_t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \widehat{dx} \\ \widehat{dx} \\ \widehat{dz} \end{pmatrix}, \tag{7.20}$$

starting at the position $(\tilde{x}_{t-1}, \tilde{y}_{t-1}, \tilde{z}_{t-1}) = (x_0, y_0, z_0)$.

Through this approach, our method can generate translations from novel instances of feature vectors, and track the position of a subject without an optical source. This property is important in complex unconstrained environments, where optical systems cannot be directly applied. As joint angles are inherently supplied by the inertial devices, our approach can simultaneously track both position and posture from a ***single*** set of sensors.

7.3 Results

7.3.1 Simulation results

Our learning framework was first evaluated on recorded sequences from the Carnegie Mellon University (CMU) Motion Capture Database (`http://mocap.cs.cmu.edu`). Motions in this dataset were captured using an optical system that tracks reflective

markers on the subject's body. Posture vectors were formed by aggregating the detected marker positions for joints on the lower body parts (thighs, shins, ankles, feet). Note that this is a slightly different representation to what was presented in Section 7.2, where posture vectors consisted of joint angles, not joint positions. However, this differentiation does not impact the applicability of our algorithm, which has no internal model of the nature of the supplied feature vectors.

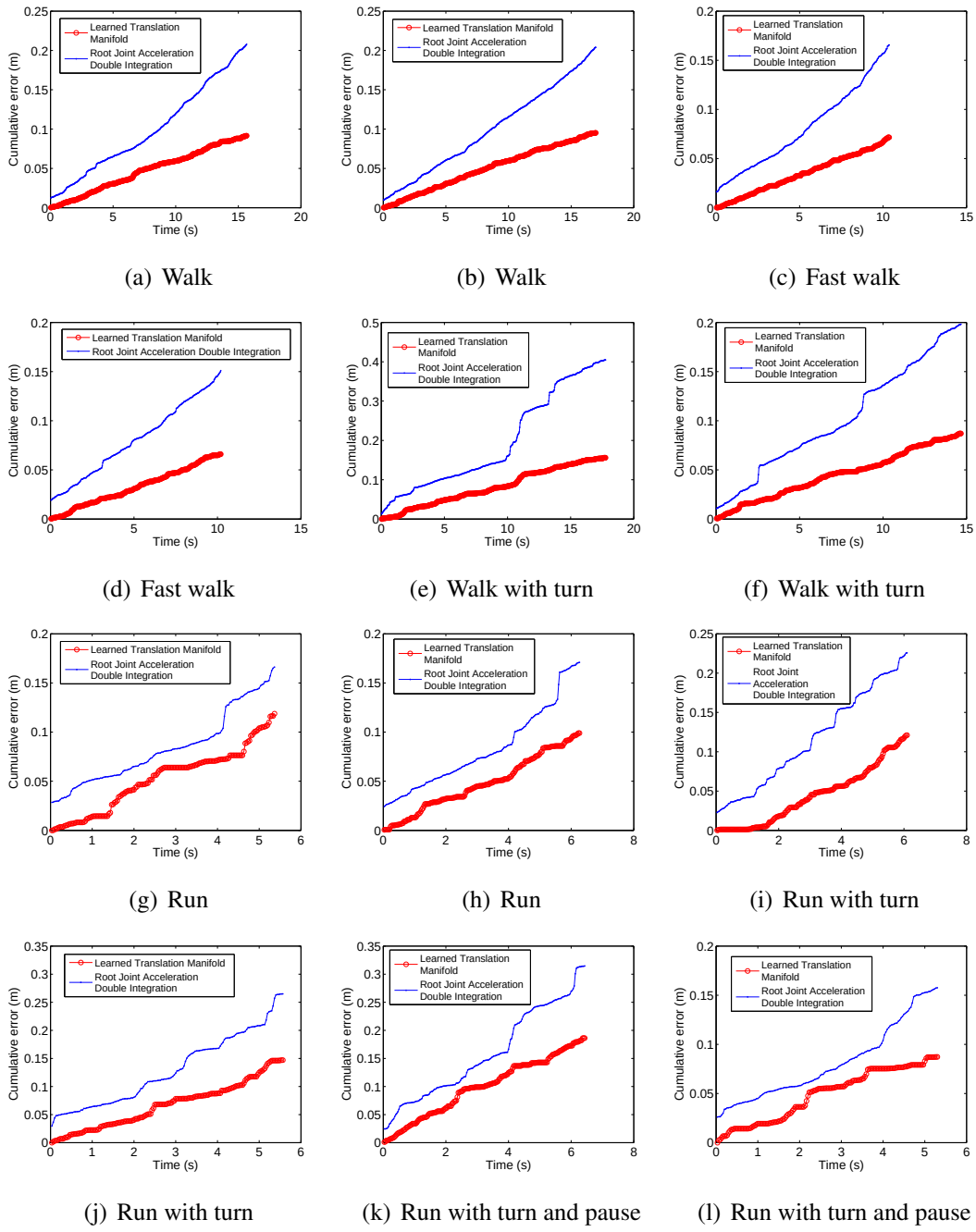
The position of the *root joint*, at the subject's hips, was taken as the absolute position of the body. This was used as a ground-truth benchmark, against which the iteratively predicted positions could be checked.

We compared our algorithm with a related open-loop, model-free position generation technique, the *double integration* of acceleration. Through this calculation, this method similarly generates local translations that can be chained together to compute positions. As the CMU dataset does not explicitly provide acceleration data, we simulated this information by extracting accelerations from successive positions at the subject's root joint, and integrating them twice to estimate translations.

We first demonstrate the ability of learned translation manifolds to correctly reproduce translations on the datasets they are trained on. Towards this end, we obtained several motion sequences and trained the learning algorithm on each of them individually. We assessed the similarity of the generated translations with the ground-truth translations, as estimated by differences of consecutive root joint positions. Our metric is the *cumulative translation error*, obtained by iteratively summing the Euclidean distance of each generated translation from the corresponding ground-truth translation.

The results for 12 distinct motion sequences, ranging from simple straight walking to running with turns, are shown in Figure 7.5. In all cases, the translations generated by the learned manifold yield a lower cumulative error than the corresponding double integration ones. For the simpler walking motions, the discrepancy between the two methods is shown to increase over time, thus suggesting that the translation manifold is more effective at capturing the dynamics of these motions.

The true potential of learned translation manifolds can be fully assessed when applied on *novel* instances of previously unseen motions. In our second simulated experiment, we trained our model on a dataset consisting of 11 different motions by the same subject: a straight walk, a straight walk followed by a 90° left turn, a straight walk followed by a 90° right turn, a walk with a left veer, a walk with a right veer, a fast straight walk, a straight run, a run followed by a 90° left turn, a straight run followed by a 90° right turn, a run with a left veer, and a run with a right veer. The total duration of these



(a) Walk (b) Walk (c) Fast walk

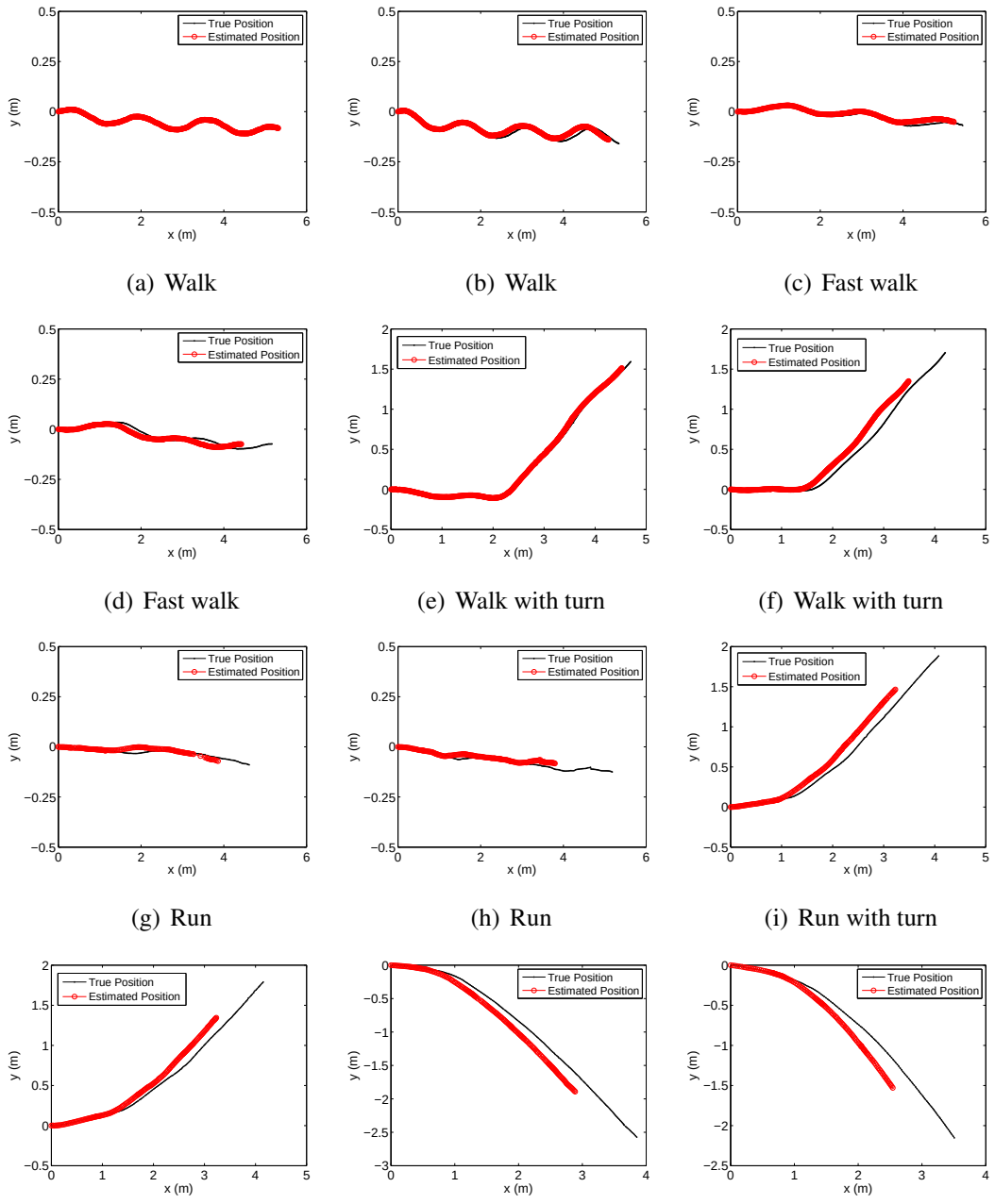
(d) Fast walk (e) Walk with turn (f) Walk with turn

(g) Run (h) Run (i) Run with turn

(j) Run with turn (k) Run with turn and pause (l) Run with turn and pause

Figure 7.5: Cumulative generated translation errors on 12 different motions from the CMU database. *Red:* Learned translation manifold that has been trained on the given motion sequence. *Blue:* Double integration of the acceleration of the root joint.

captures is 114 seconds, with walking-type and running-type motions accounting for 86 and 28 seconds, respectively. By including different types of motions, our aim was to model a wide range of posture-translation pairs, and maximise the likelihood that novel motion instances will be captured by our training set.



(a) Walk (b) Walk (c) Fast walk

(d) Fast walk (e) Walk with turn (f) Walk with turn

(g) Run (h) Run (i) Run with turn

(j) Run with turn (k) Run with turn and pause (l) Run with turn and pause

Figure 7.6: Overall position error estimation for 12 novel instances of unseen motion sequences. *Red:* Trajectories estimated by learned translation manifold. *Black:* Ground-truth trajectories.

The learned mapping was applied on 12 new motion sequences of various types. For these motions, we measured the discrepancy between the trajectories predicted by the learned manifold, and the ground-truth trajectories. The resulting trajectories are demonstrated in Figure 7.6. As previously, our algorithm is shown to reproduce ac-

curate positions for normal walks, with the error increasing for running-type motions. This increase is partly explained by the larger number of walking motion data points in the training set, which biases the manifold towards translations of smaller magnitude.

7.3.2 Experimental results

In our second set of experiments, we evaluated the translation learning algorithm on sensory data obtained from physical devices, using the Kinect as an optical source and the Orient platform as the inertial sensing source. For the training phase of the algorithm, data produced by these two sources were synchronised to learn a translation manifold, as described in Section 7.2. An important restriction in this case was the small capture volume of the Kinect (approximately 15m^3), which limited the variety of motions that could be performed by the subject. Thus, our framework is evaluated mainly on walking-type motions that require less physical space for training purposes.

7.3.2.1 Constrained environment experiments

The learning algorithm was first compared with the acceleration integration method against ground-truth positions estimated by the Kinect. Unlike simulation experiments, accelerations were now directly supplied by the accelerometers of Orient devices, so translations were generated through double integration of this data.

We captured 18 motion sequences of variable length, ranging from 20 to 180 seconds. We used a total of 4 Orient devices, placed on the subject's waist (root joint), right thigh, left thigh, and left ankle. Motions were captured in an office building environment, which impacted the quality of the sensory readings, especially magnetometers, due to metal in the building structure. For each capture, the subject was allowed to perform any sequence and combination of walking and standing, provided s/he remained within the capture area of the Kinect.

We selected 12 of the captured sequences as the training set, and we used the remaining 6 as novel instances for evaluation. As with simulation experiments, we first compared the cumulative error of translations generated by the learned manifold, and translations from double integration of root joint accelerations.

Figure 7.7 illustrates this comparison, along with the corresponding ground-truth positions captured by the Kinect. In all 6 trials, the subject was observed to repeatedly move around the capture area in a loop. Although in some cases the cumulative error generated by the double integration method was initially lower, in all trials the learn-

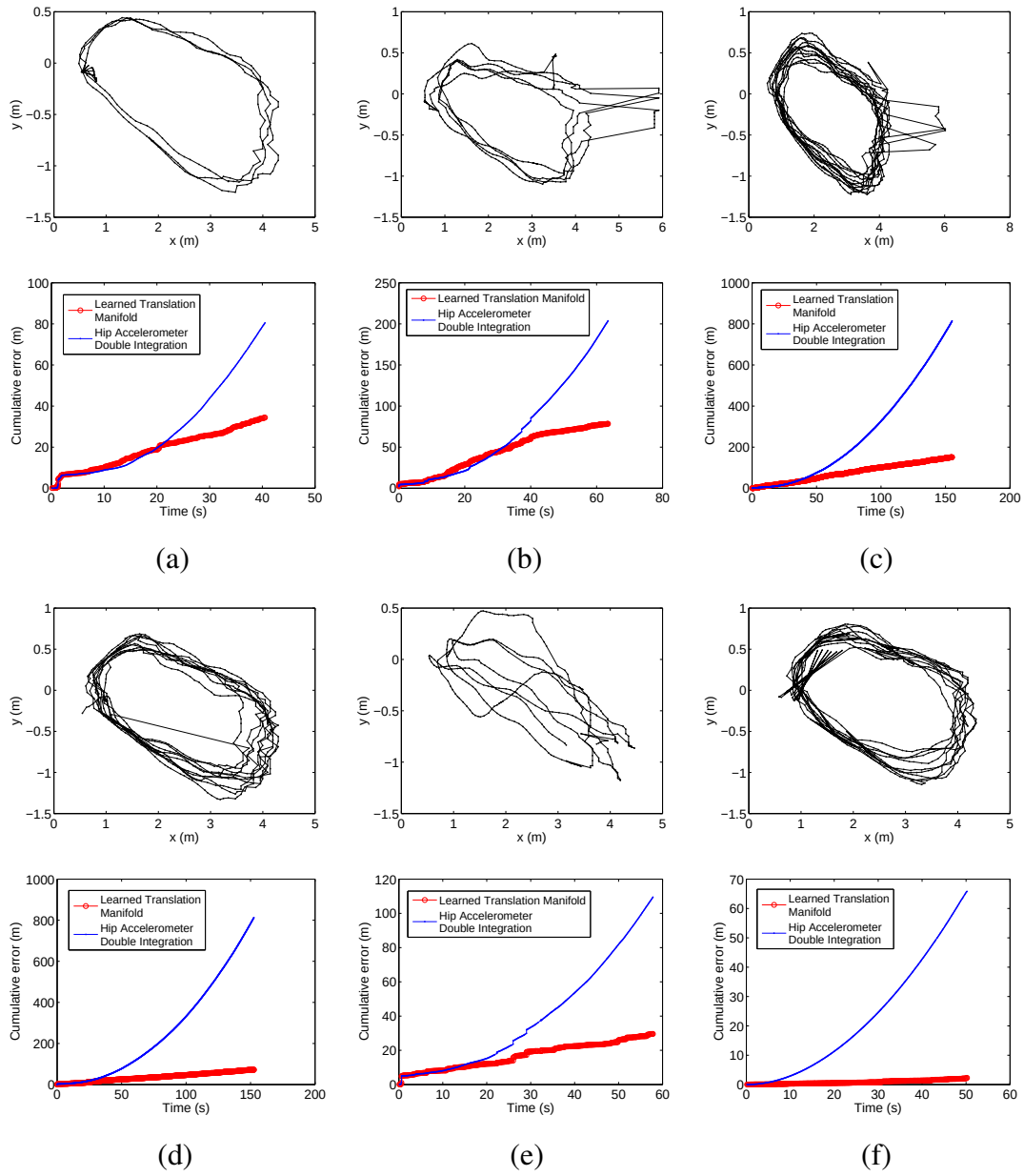
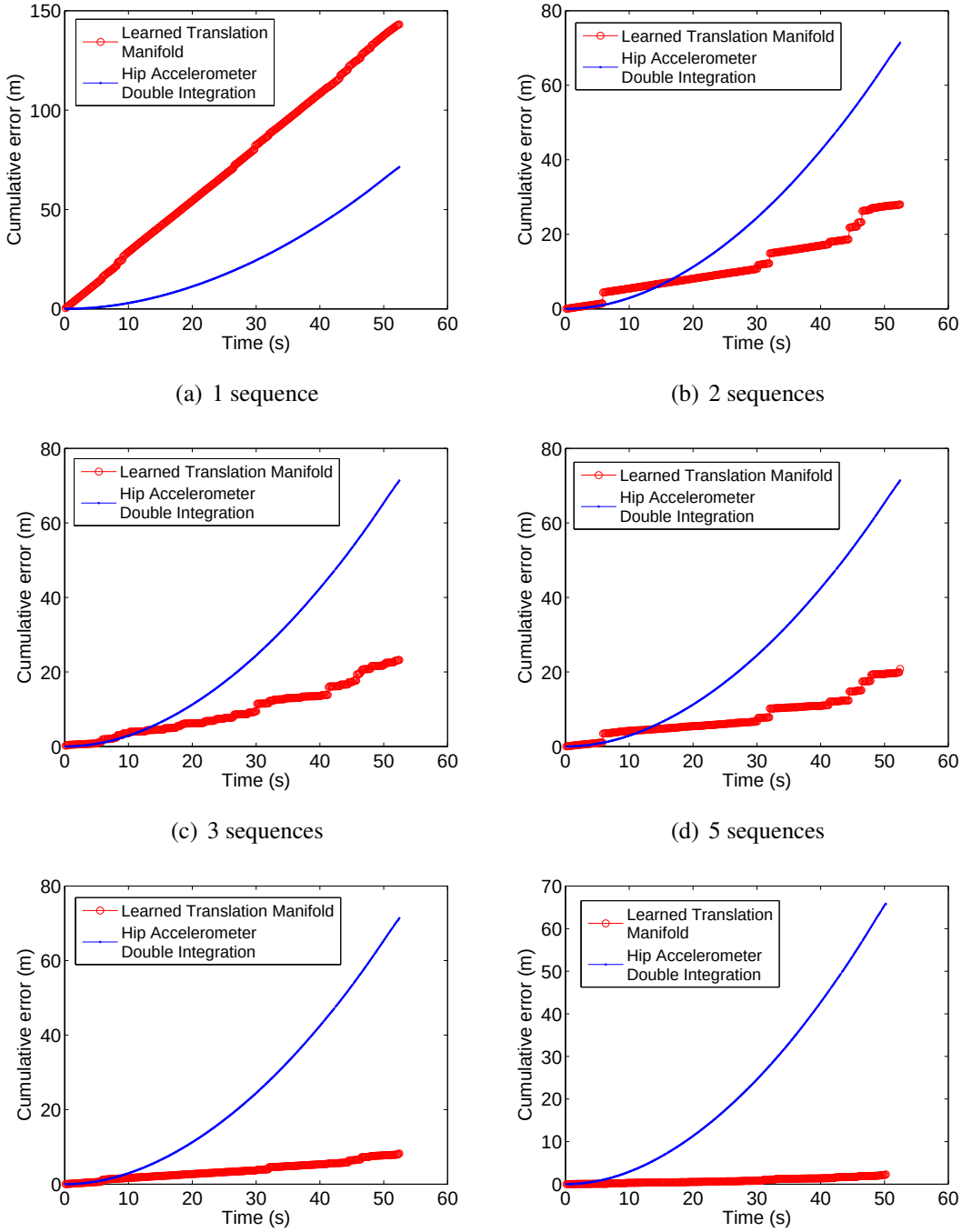


Figure 7.7: Cumulative generated translation errors for 6 novel motion sequences. The learning algorithm was trained on 12 sequences of varying length and motion composition. *Top of each subfigure:* Ground-truth positions computed by the body tracking interface. *Bottom:* Cumulative errors. *Red:* Learned translation manifold. *Blue:* Double integration of the acceleration of the root joint.

ing method had a considerably lower error at the end of the sequence. This superior performance was achieved despite some irregularities in the captured positional data, as, for example, in Figures 7.7(b) and 7.7(c). This is an important result demonstrating that our approach can learn a robust translation model on top of potentially noisy sen-



(a) 1 sequence (b) 2 sequences

(c) 3 sequences (d) 5 sequences

(e) 10 sequences (f) 12 sequences

Figure 7.8: Effect of training data set size on generated translations. The cumulative error is shown to decrease as the number of training motion sequences increases.

sory data, which can yield more accurate translations than methods operating directly on raw data.

The error of the translation manifold algorithm inevitably depends on the size and

quality of the training data set. To better understand this effect, we assessed the performance of the algorithm on the last trial of Figure 7.7 under varying training sets. The results are shown in Figure 7.8, where we start with just one training sequence, comprising only a few data points, and progressively increase this number. It can be seen that when only one short sequence is supplied, the performance is considerably worse than the double integration method. However, as more motion instances are added to the training set, the error is shown to decrease significantly over time. This indicates that the learning model relies on a good coverage of the posture and translation space, in order to be able to generalise effectively to novel instances. Thus, when recording data, it is important to ensure that the tracked subjects perform a wide range of motions, including various motion combinations (e.g. straight walks and turns).

Another related constraint on the performance of our algorithm is that motions captured during training must be similar to those executed in the online generation phase. For example, if a manifold is learned only from walking motions, it is highly unlikely to yield accurate translations on novel running motions. Thus, it is essential to capture not only a significant *quantity* of data (as shown in Figure 7.8), but also representative sequences that will be *qualitatively* similar to the motions the system will be tested on when deployed.

(a) (b) (c)

Figure 7.9: Illustration of unconstrained office environment. (a): The room and corridor in which our method was tested. (b): Approximate trajectory followed by the subject, starting and ending at the same point. (c): Approximate dimensions of the trajectory.

7.3.2.2 Unconstrained environment experiments

In the second set of physical experiments, we evaluated the performance of the learning in an unconstrained office environment (Figure 7.9). This environment represents a

setting in which an optical source cannot be used to track subjects, due to its larger area and morphology (doors, corridors). The subject was asked to follow a trajectory consisting of several landmark points, located inside an office room and in an adjacent corridor. There was no restriction on the time given to follow this trajectory, so the subject was allowed to pause for arbitrary periods of time.

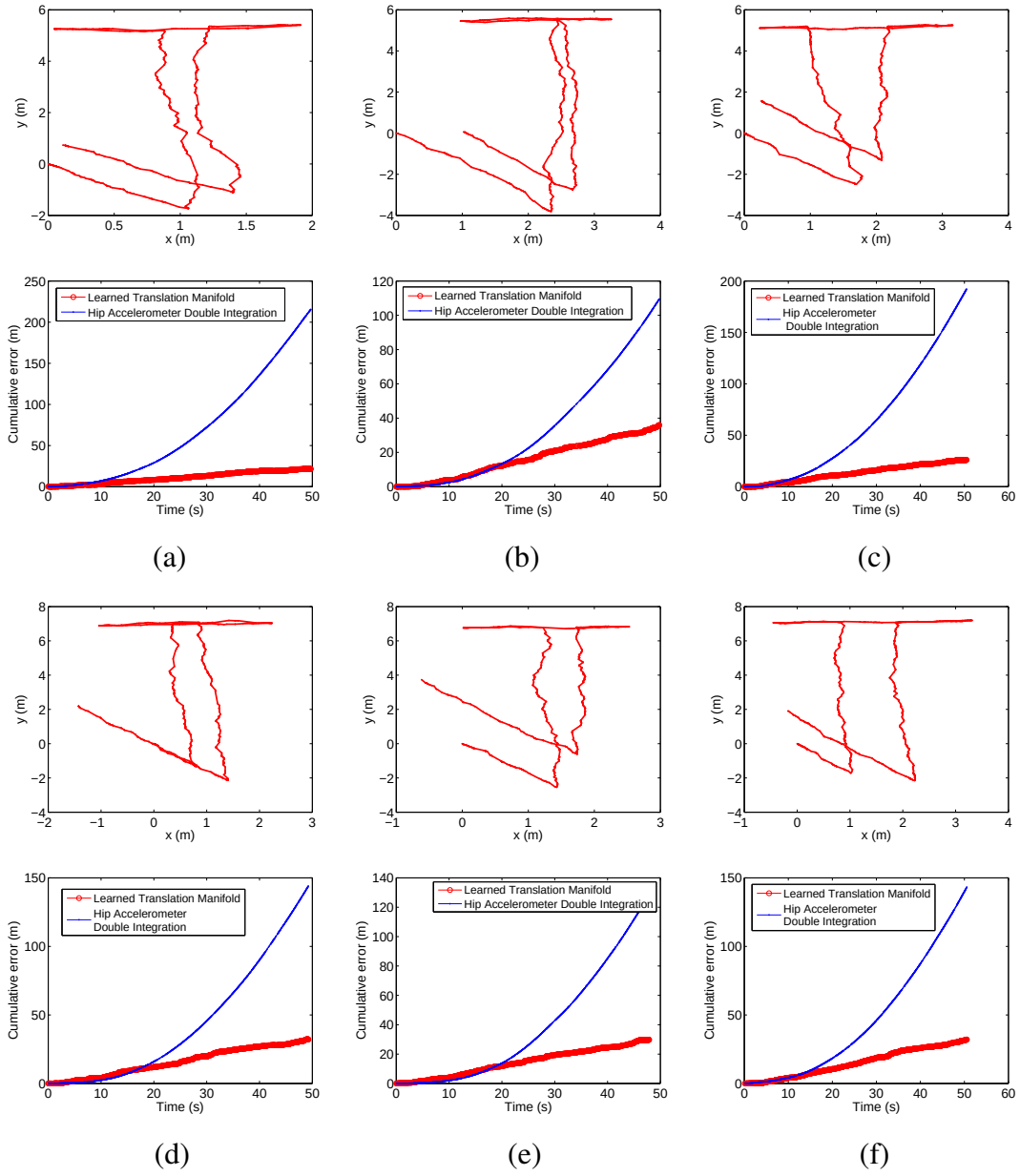


Figure 7.10: Generated positions for a subject following the trajectory shown in Figure 7.9, 6 trials. *Top subfigures:* Generated positions. *Bottom:* Cumulative errors and comparison with double integration.

We used the same set of training sequences as in the first set of physical exper-

iments to learn a translation manifold. Figure 7.10 shows the generated trajectories for six distinct trials of the subject moving along the prescribed path, along with the corresponding error comparison with the double integration method. The true precise trajectory followed in each case was not known, however, the subject always ended his path at the same point where he started. Thus, by comparing the difference between the start and end points of each generated trajectory, we can get an estimate of the resulting error.

Despite not being aware of the duration and nature of the motions performed by the subject, the learning algorithm is observed to produce translations that closely follow the true trajectory. The computed mean error for the final position was 1.783m, with the overall sum of distances between landmark points being about 30m. Furthermore, as shown in the bottom subfigures of Figure 7.10, the learning algorithm maintains the superior performance level over the double integration method. A common trait of both sets of physical experiments is that they feature several alternations between straight walking and turning motions, which are characterised by repeated variations in the velocity profile of the tracked subject. In this context, the double integration method initially produces an error comparable with the learning algorithm, but in both cases the margin increases exponentially over time. The learning method is therefore successful in identifying the salient structure of the high-dimensional data, and using it to learn a mapping that can be applied to novel motions.

7.4 Conclusions

We have presented a method for simultaneous posture and position tracking in unconstrained environments, based on learned generative translation manifolds. In an offline learning phase, two heterogeneous tracking sources, an inertial sensing (Orient-4) and an optical (Kinect) platform, are jointly used to learn a mapping from posture variations, as estimated by the former, to whole-body translations, as estimated by the latter. This mapping is learned through linear regression on clustered latent representations of posture variation vectors. Online, the optical source is removed, and the learned translation manifold is used to generate translations for unknown, novel motion instances. The generative method is experimentally shown to outperform the related model-free, dead-reckoning method of acceleration integration, and to correctly reproduce the structure of previously unseen, complex trajectories in unconstrained environments.

One drawback of our approach is that a different mapping must be learned whenever the system is tested on a new user. This characteristic is due to skeletal morphology and limb dimension constraints, which vary among different subjects. Thus, motion sequences captured on a specific subject may not adequately cover the posture difference and translation space for a different subject, thus leading to incomplete mappings. Nevertheless, one interesting extension to our work would be to learn translation manifolds from datasets which contain motions from various subjects with different characteristics (e.g. short/tall). This extension would be well-suited to the feature vector clustering procedure described in Section 7.2.2.1. In this context, we would employ a *hierarchical* clustering approach, where the evaluated subject would first be matched to the nearest (in terms of body morphology) user in the training data set, and then a translation would be generated based on the learned mappings for the matched subject.

A major strength of our approach is that it does not make any assumptions on the nature of motion being performed, the number and placement of inertial measurement units, or the morphology of the tracked subject's body. This property is advantageous for two reasons. First, our method can be applied to complex motions spanning all three dimensions (e.g. forward jumps), where traditional model-based approaches tracking gait events and foot contacts would fail. Second, for simpler, planar motion types (e.g. walking sequences), our method can be used as a predictive step for model-based filtering approaches, to yield superior tracking results and lower positional errors. Extending our work in these directions would further emphasise the benefits of using machine learning techniques to exploit the structure of high-dimensional data produced by physical sensor networks.

The applicability of our method can be extended to direct interactions between humans and robots. In this context, our algorithm can be used to track the positions and postures of human subjects, and provide this information in real time to interacting robots. Through this extension, our method could lead to the development of novel forms of human-robot interaction, in domains such as – but not limited to – unconstrained office and home environments, which depend heavily on the quality and quantity of available sensory information. As demonstrated by the results presented in this thesis, this type of information is a prerequisite for robust, strategic decision making in physical robotic environments. Thus, the method presented in this chapter contributes to this direction by bridging the gap between sensing technologies and information processing algorithms in complex human-robot systems.

Chapter 8

Conclusions and future work

8.1 Main contributions

This thesis introduces and proposes an integrated approach to the problem of interaction shaping between heterogeneous physical robots. The novel challenge in this problem lies in indirectly influencing (but not directly forcing) an interacting non-cooperative agent into moving to a target state. In this context, the goal is to find a sequence of strategic actions that can incite the desired responses from the interacting agent, and *shape* the evolution of the interaction accordingly. Our primary theoretical contribution is a framework for autonomous shaping of interactions with non-cooperative robots, which is based on a combination of offline learning from human demonstrations, and online empirical learning during an interaction. Our primary experimental contribution is a demonstration of the benefits of our approach on NAO humanoid robots, in a variety of interactions with other autonomous and human-controlled agents.

In Chapter 3, we introduce the different challenges an autonomous robot must face in an interactive, adversarial environment. We distinguish between sensing uncertainty, arising from the need to estimate the state of an interacting agent, and strategic uncertainty, arising from the need to infer the intent and strategy of that agent. For the first part, our contribution is the Reachable Set Particle Filter, a state estimation algorithm combining analytical dynamical constraints and empirical observations. For the second part, we propose a decision-making algorithm that is based on the ideas of regret minimisation and exploitation of the adversary's sensing capabilities. We experimentally validate our approach in a simulated robotic soccer domain, where we demonstrate the benefits of negotiating the interaction environment in this principled, modular way.

In Chapter 4, we lift the complexity of our experimental domain to address the challenges of physical robotic environments. We also consider the more difficult case where behaviours cannot be handcrafted from domain knowledge, and must instead be learned through a more general procedure, such as from provided human demonstrations. Our contribution is a method for probabilistically synthesising demonstrated traces of interactions into adaptive behaviours, which selects appropriate actions through a dynamically weighted Gaussian Mixture Model. Unlike many existing approaches to learning from demonstration, the provided demonstrations need not be optimal, in the sense of being executed by "expert" operators. The experimental evaluation of this method illustrates its ability to adapt to novel adversaries and generate strategies that are not directly encoded in the demonstrated traces.

In Chapter 5, we present our main contribution to the interaction shaping problem, which is a Bayesian framework that can interactively adapt to a given, unknown adversary. This framework also builds on interaction traces demonstrated by human subjects, but uses them as a basis for an empirical learning algorithm that progressively updates the expected utility of different actions and strategies. One of the novel concepts in this approach is an interactive formulation of reachability of different state space regions, based on opponent modeling techniques that account for the observed actions of the adversary. This formulation is then used to identify, through Bayesian inference, actions and strategies that are likely to influence the adversary in the desired way. Our experiments demonstrate an autonomous agent that can successfully learn to improve its influence over robots teleoperated by experienced human users, thus attaining interactively learned shaping behaviours.

In Chapter 6, we address the factors that affect human decisions in these complex, physical, strategic interactive environments. Our main contribution is a user study assessing the performance of subjects in multiple interactive teleoperation tasks, under different experimental conditions. Our focus is on factors that constrain the perception of the evaluated subjects, which effectively reduces their sensorimotor capabilities to those of an autonomous robot and allows for a direct comparison of their interactive decisions. This makes our study one of the first to contrast human and robot decision-making in a realistic experimental setup, while also evaluating human responses to shaping strategies and behaviours like the ones described above. Our experiments demonstrate that restricted perception has an adverse effect on user performance in complex interactive tasks. These findings have implications for other applications of human-robot interaction, where there is a need to appropriately distribute the roles

between the interacting parties.

In Chapter 7, we present a sensing algorithm towards addressing the needs of decision shaping in direct – and not teleoperation-mediated – strategic human-robot interactions. The novelty in this algorithm lies in the combined use of inertial sensing (Orient platform) and optical (Kinect) motion capture systems in order to learn a motion model for a given tracked human subject. This approach combines the relative advantages of the two systems and allows for human posture and position to be tracked in unconstrained environments. On the one hand, this is an important contribution to motion capture systems, where heterogeneous tracking systems and learning models have not been previously used to detect motion in such a way. On the other hand, our method can be directly applied to non-trivial interactions between humans and robots, where there is currently a lack of practical techniques for capturing and continuously supplying human motion data to an autonomous robot.

8.2 Evaluation and lessons learned

One recurring theme in most chapters of this thesis has been experimentation with real robotic systems, where autonomous robots are pitted directly against human-controlled and other autonomous adversaries. Given the interactive and human-centric nature of several aspects of our algorithms, this physical setting provides a more natural interface for human operators (than, for example, having users interacted with simulated implementations of robots). Furthermore, real-robot evaluation also allows us to fully test the robustness of our algorithms, and determine the influence of exogenous factors (e.g. uncertainty in sensing, locomotion, and localisation). However, experimentation with physical robots is not without its flaws, especially in experiments involving learning like our own. Indeed, there is a hard limit on the number of trials that can be executed, before a robot runs into hardware issues such as motor overheating and low battery power. Moreover, testing for theoretical performance guarantees becomes harder on physical platforms, which is why most of our results were empirical. As algorithmic and multi-agent tools are gradually gaining traction in the field of human-robot interaction, we believe that it is important to establish a solid interface between simulated and physical experiments, in order to develop robust interactive systems with performance standards.

Another defining characteristic of this thesis was that most of our experimental scenarios were inspired from our participation in the RoboCup Standard Platform League

(SPL). Although RoboCup soccer has a long-standing association with multi-agent systems and reinforcement learning research, we believe that the "human" aspect of the competition, as investigated in this thesis, is relatively novel. Indeed, much of the success in RoboCup SPL still depends on fundamental low-level components (vision, locomotion, localisation) and less so on decision-making and multi-robot interaction. However, by participating in this competition and observing the types of problems autonomous robots run into, we gained valuable insights on possible interactive experiments that could incorporate human factors. Thus, even though events like RoboCup are primarily of a competitive nature, we believe that there is a lot to learn by observing autonomous robots exhibiting what they can, and, most importantly, what they cannot do.

8.3 Future directions

The problems and methods discussed in this thesis begin to address several significant challenges in autonomous robotics and human-robot interaction. Many of these issues, such as indirect influence and implicit persuasion, are becoming increasingly important in robotic systems, where there is a need for more robust and seamless interaction between humans and autonomous agents. This thesis presents a concrete theoretical framework towards this goal, as well as a wide range of experiments in the popular robotic soccer domain. However, these ideas can be further refined and extended to robotic applications of broader interest. In the remainder of this section, we discuss some of these possible extensions, where we believe our thesis can have an impact in the near future.

8.3.1 Direct strategic human-robot interactions

One of the remaining big open questions is that of direct strategic interaction between humans and robotic systems. The teleoperation approach followed in the largest part of the thesis was chosen for two reasons: first, because it allows for better comparison between human and robot decision making (by making the two sides interact using identical "bodies"), and second, because of the challenges involved in tracking the state of the interacting human partner. The method proposed in Chapter 7 offers a practical solution to the second issue, however, it has so far been applied only on single-subject human motion capture examples.

An immediately attainable and interesting extension is to use our methods in complex direct human-robot interactions, for example, in scenarios taking place in domestic or office environments. These are environments that can benefit from the application of our approach, as they involve human motion in unconstrained space (e.g. moving in and out of rooms, navigation along corridors). Furthermore, these domains present several interesting applications where humans and robots are required to affect each other's decisions. For example, in a home environment, a human and a robot may be tasked with collaborating in order to prepare a meal; in this context, it would be desirable for the robot not only to receive instructions from the interacting human subject, but also to influence that person into performing certain tasks or retrieving certain objects. In such applications, our ideas and techniques can address several open issues, ranging from low-level state estimation to high-level decision making. By tackling these problems, our methods can lead to effective decision shaping and interaction with physically present human subjects, in a wide range of challenging experimental domains.

8.3.2 Integration with path planning algorithms

The decision methods presented in this thesis would also benefit from extensions incorporating autonomous path planning techniques. The path planning domain has seen several important developments in recent years, centred around widely adopted techniques such as *rapidly exploring random trees* (RRTs) (LaValle, 2006) and *probabilistic roadmaps* (PRMs) (Kavraki et al., 1996). These techniques and their derivatives have been successfully deployed in several important robotic domains, such as grasping and obstacle avoidance in cluttered environments. However, extensions incorporating elements of decision-theoretic planning (Boutilier et al., 1999) and interaction with adversarial agents have been less studied. One noteworthy recent line of work investigated the combined use of belief-space constraints and RRT methods, as a means with dealing with uncertainty in the state of the environment (Bry and Roy, 2011).

Our approach would be to extend and situate our methods within RRT-style frameworks, in order to provide theoretical guarantees on the execution of interaction shaping behaviours. To achieve this goal, the existing RRT formalisms would be extended to incorporate constraints on interacting adversarial agents, which would reflect the ability to successfully execute different strategic action sequences. In effect, we would seek to refine the obstacle space constraints that currently form the basis of most RRT

algorithms, in order to address the challenges of impacting the state of interacting strategic agents. The primary benefit of this approach would be that our techniques would be augmented with bounds on the feasibility of shaping behaviours against a given adversary. Thus, it would be possible to determine, given a set of demonstrated interaction traces and sampled opponent responses, whether a robust interaction shaping strategy exists, or if more demonstrations are needed to achieve this goal. This extension could also introduce an opportunity to integrate the offline and online learning components of our method more tightly, and assist in determining regions of the state and action spaces where demonstrations should be focused.

8.3.3 Extension to domains with more robots

A limitation of many of the algorithms developed in this thesis (particularly those introduced in Chapters 4 and 5) is that they have been defined for and tested in systems of two robots. Inevitably, any increase in this number would impact the complexity and flexibility of the proposed approaches. When interacting with more than one robot, one possible extension for our framework would be to consider modeling the behaviour of the interacting team as an ensemble, rather than updating and reasoning about the responses of individual opponents. This could be effected, for example, by defining team-level action vectors, representing the actions of a group of robots, which could be directly plugged in into our existing formulation. A related issue would be to look at the impact of different team members on an interaction, in order to determine which of those would be most likely to comply with a given shaping behaviour. In this context, an important open question would be whether an interaction with a team of robots can be successfully shaped even if only a fraction of those robots act in the desired manner. This property could be specified as an additional level in the system hierarchy, i.e. determining which robot(s) the shaping agent should focus interacting with, in order to maximise the effectiveness of a shaping strategy.

8.3.4 Improving teleoperation mechanisms in mixed-initiative systems

This thesis has primarily considered the problem of interaction shaping in *multi-robot* environments, where a robot is tasked with influencing another non-cooperative agent. However, our approach can also be extended to *single-agent* teleoperation applications with mixed-initiative agents combining elements of human control and autonomy.

Here, the goal for the agent would be to influence its *own* operator, and not a different robot. By doing so, the agent would learn to improve the quality of the decisions made by its operator over time.

This extension comes close to the idea of sliding autonomy (Heger and Singh, 2006), where a human operator can intervene to assist an autonomous robot whenever necessary. However, we seek to use our shaping model to address the opposite question, i.e. how a robot can autonomously assist an operator in tasks where human control may be error-prone. In this context, the shaping agent would be tasked with modifying human inputs in a way that not only improves their current effect, but also influences the operator towards better decisions in the future. As demonstrated in Chapter 5, this outcome can be attained by modeling the interaction with the operator as a shaping problem, where the goal is to determine action strategies that are likely to attain a desired joint future state.

The implications of such an extension would be important for several teleoperation applications in field and rescue robotics, where the data provided to the operator from the robot is often sparse or incomplete (e.g. the limited field of view case discussed in Chapter 6), and where communication may be limited. In this context, models of autonomous influence can assist in overcoming this uncertainty, and improving the quality of interactive teleoperation decisions in challenging environments.

Bibliography

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*.

Agmon, N. and Stone, P. (2012). Leading ad hoc agents in joint action settings with multiple teammates. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Aler, R., Garcia, O., and Valls, J. M. (2005). Correcting and improving imitation models of humans for robosoccer agents. In *Congress on Evolutionary Computation*, pages 2402–2409.

Argall, B., Gu, Y., Browning, B., and Veloso, M. M. (2006). The first segway soccer experience: towards peer-to-peer human-robot teams. In *International Conference on Human Robot Interaction (HRI)*, pages 321–322.

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

Arvind, D. K. and Valtazanos, A. (2009). Speckled tango dancers: Real-time motion capture of two-body interactions using on-body wireless sensor networks. In *IEEE International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pages 312–317.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2003). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77.

Avrahami-Zilberbrand, D. and Kaminka, G. A. (2005). Fast and complete symbolic plan recognition. In *International Joint conference on Artificial intelligence (IJCAI)*, pages 653–658.

Bainbridge, W. A., Hart, J., Kim, E. S., and Scassellati, B. (2008). The effect of presence on human-robot interaction. In *Robot and Human Interactive Communication,*

2008. RO-MAN 2008. The 17th IEEE International Symposium on, pages 701–706. IEEE.

Baker, C. L., Saxe, R., and Tenenbaum, J. B. (2009). Action Understanding as Inverse Planning. *Cognition*, 113(3):329–349.

Bard, N. and Bowling, M. (2007). Particle filtering for dynamic agent modelling in simplified poker. In *National conference on Artificial intelligence (AAAI)*, pages 515–521.

Barrett, S. and Stone, P. (2012). An analysis framework for ad hoc teamwork tasks. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Belta, C., Bicchi, A., Egerstedt, M., Frazzoli, E., Klavins, E., and Pappas, G. J. (2007). Symbolic planning and control of robot motion [grand challenges of robotics]. *IEEE Robotics & Automation Magazine*, 14(1):61–70.

Bernstein, D. S., Zilberstein, S., and Immerman, N. (2000). The complexity of decentralized control of Markov decision processes. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

Bhattacharya, S. and Hutchinson, S. (2010). On the existence of Nash equilibrium for a two-player pursuitevasion game with visibility constraints. *International Journal of Robotics Research (IJRR)*, 29(7):831–839.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Robot programming by demonstration. In *Springer Handbook of Robotics*.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bobick, A. F. and Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267.

Boutilier, C., Dean, T., and Hanks, S. (1999). Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94.

Bouton, M. (2007). *Learning and Behavior: A Contemporary Synthesis*. Sinauer Associates.

Bradtke, S. J. and Duff, M. O. (1994). Reinforcement learning methods for continuous-time markov decision problems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 393–400.

Browning, B., Xu, L., and Veloso, M. (2004). Skill acquisition and use for a dynamically-balancing soccer robot. In *National Conference on Artificial Intelligence (AAAI)*.

Broz, F., Nourbakhsh, I., and Simmons, R. (2008). Planning for human-robot interaction using time-state aggregated pomdps. In *National Conference on Artificial intelligence (AAAI)*, pages 1339–1344.

Bry, A. and Roy, N. (2011). Rapidly-exploring random belief trees for motion planning under uncertainty. In *International Conference on Robotics and Automation (ICRA)*.

Bui, H. H. (2003). A general model for online probabilistic plan recognition. In *International Joint Conference on Artificial intelligence (IJCAI)*, pages 1309–1315.

Burridge, R. R., Rizzi, A. A., and Koditschek, D. E. (1999). Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research (IJRR)*, 18(6):534–555.

Calinon, S., D'halluin, F., Sauser, E. L., Caldwell, D. G., and Billard, A. (2010). Learning and reproduction of gestures by imitation. *IEEE Robotics and Automation Magazine*, 17(2):44–54.

Calinon, S., Guenter, F., and Billard, A. (2006). On learning the statistical representation of a task and generalizing it to various contexts. In *International Conference on Robotics and Automation (ICRA)*.

Carberry, S. (2001). Techniques for plan recognition. *User Modeling and User-Adapted Interaction*, 11(1-2):31–48.

Cassandra, A. R. (1998). A survey of POMDP applications. In *Working Notes of AAAI 1998 Fall Symposium on Planning with Partially Observable Markov Decision Processes*, pages 17–24.

Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., and Warmuth, M. K. (1997). How to use expert advice. *Journal of the ACM*, 44(3):427–485.

Cesa-Bianchi, N. and Lugosi, G. (2012). Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422.

Chalodhorn, R., Grimes, D. B., Grochow, K., and Rao, R. P. N. (2007). Learning to walk through imitation. In *International Joint Conference on Artifical intelligence*, pages 2084–2090.

Charniak, E. and Goldman, R. P. (1993). A bayesian model of plan recognition. *Artificial Intelligence*, 64(1):53–79.

Chatzis, S. P., Korkinof, D., and Demiris, Y. (2012). A quantum-statistical approach toward robot learning by demonstration. *IEEE Transactions on Robotics*, PP(99):1 –11.

Chernova, S. and Veloso, M. (2007). Confidence-based policy learning from demonstration using gaussian mixture models. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 233:1–233:8.

Conner, D. C., Choset, H., and Rizzi, A. (2006). Integrated planning and control for convex-bodied nonholonomic systems using local feedback. In *Robotics: Science and Systems (RSS)*, pages 57–64.

Corrales, J. A., Candelas, F. A., and Torres, F. (2008). Hybrid tracking of human operators using imu/uwb data fusion by a kalman filter. In *International Conference on Human Robot Interaction (HRI)*, pages 193–200.

Crick, C., Osentoski, S., Jay, G., and Jenkins, O. C. (2011). Human and robot perception in large-scale learning from demonstration. In *International Conference on Human Robot Interaction (HRI)*, pages 339–346.

de Farias, D. P. and Megiddo, N. (2004). Exploration-exploitation tradeoffs for experts algorithms in reactive environments. In *Neural Information Processing Systems (NIPS)*.

Demiris, Y. (2007). Prediction of intent in robotics and multi-agent systems. *Cognitive Processing*, 8(3):151–158.

Dominey, P., Metta, G., Nori, F., and Natale, L. (2008). Anticipation and initiative in human-humanoid interaction. In *IEEE-RAS International Conference on Humanoid Robots*, pages 693 –699.

Doshi, P. and Gmytrasiewicz, P. J. (2009). Monte carlo sampling methods for approximating interactive POMDPs. *Journal of Artificial Intelligence Research*, 34:297–337.

Dragan, A. and Srinivasa, S. (2012). Formalizing assistive teleoperation. In *Robotics: Science and Systems (RSS)*.

Duchaine, V. and Gosselin, C. M. (2007). General model of human-robot cooperation using a novel velocity based variable impedance control. In *EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems.*

Edsinger, A. and Kemp, C. (2007). Human-robot interaction for cooperative manipulation: Handing objects to one another. In *IEEE International Symposium on Robot and Human interactive Communication.*

Feliz, R., Zalama, E., and Garcia-Bermejo, J. G. (2009). Pedestrian tracking using inertial sensors. *Journal of Physical Agents*, 3(1):35–42.

Flemisch, O., Adams, A., Conway, S. R., Goodrich, K. H., Palmer, M. T., and Schutte, P. C. (2003). The H-metaphor as a guideline for vehicle automation and interaction. In *NASA/TM2003-212672*.

Fletcher, L., Teller, S., Olson, E., Moore, D., Kuwata, Y., How, J., Leonard, J., Miller, I., Campbell, M., Huttenlocher, D., Nathan, A., and Kline, F.-R. (2008). The MIT-cornell collision and why it happened. *Journal of Field Robotics*, 25(10):775–807.

Foxlin, E. (2005). Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6):38 –46.

Geib, C. and Harp, S. (2004). Empirical analysis of a probabilistic task tracking algorithm. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS) - Workshop on Agent Tracking*.

Geib, C. W. and Goldman, R. P. (2009). A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence*, 173(11):1101–1132.

Genter, K., Agmon, N., and Stone, P. (2011). Role-based ad hoc teamwork. In *Plan, Activity, and Intent Recognition Workshop at the Twenty-Fifth Conference on Artificial Intelligence (PAIR)*.

Genter, K., Agmon, N., and Stone, P. (2013). Ad hoc teamwork for leading a flock. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Gerkey, B. P., Thrun, S., and Gordon, G. (2004). Visibility-based pursuit-evasion with limited field of view. In *International Journal of Robotics Research (IJRR)*, pages 20–27.

Gmytrasiewicz, P. J. and Doshi, P. (2005). A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24:24–49.

Goodrich, M. A. and Schultz, A. C. (2007). Human-robot interaction: a survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing*, 140(2):107–113.

Grollman, D. H. and Billard, A. (2011). Donut as i do: Learning from failed demonstrations. In *International Conference on Robotics and Automation (ICRA)*.

Grollman, D. H. and Jenkins, O. C. (2007). Learning robot soccer from demonstration: Ball grasping. In *Robotics: Science and Systems (RSS) - Robot Manipulation: Sensing and Adapting to the Real World*.

Havoutis, I. (2012). Motion planning and reactive control on learnt skill manifolds. In *PhD Thesis, University of Edinburgh*.

Heger, F. and Singh, S. (2006). Sliding autonomy for complex coordinated multi-robot tasks: Analysis and experiments. In *Robotics: Science and Systems (RSS)*.

Hester, T., Quinlan, M., and Stone, P. (2010). Generalized model learning for reinforcement learning on a humanoid robot. In *International Conference on Robotics and Automation (ICRA)*, pages 2369–2374.

Howard, R. A. (1971). *Dynamic Probabilistic Systems: Semi-Markov and Decision Processes*. New York: Wiley.

Hsiao, K., Kaelbling, L. P., and Lozano-Perez, T. (2007). Grasping pomdps. In *International Conference on Robotics and Automation (ICRA)*, pages 4685–4692.

Johnson, M. and Demiris, Y. (2005). Perceptual perspective taking and Action Recognition. *International Journal of Advanced Robotic Systems*, 2(4):301–308.

Jordan, P. R. and Wellman, M. P. (2009). Generalization risk minimization in empirical game models. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134.

Karaman, S. and Frazzoli, E. (2010). Sampling-based algorithms for a class of pursuit-evasion games. In *Workshop on Algorithmic Foundations of Robotics (WAFR)*.

Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566 –580.

Klingspor, V., Demiris, J., and Kaiser, M. (1997). Human-robot communication and machine learning. *Applied Artificial Intelligence*, 11(7):719–746.

Knox, W. B. and Stone, P. (2009). Interactively shaping agents via human reinforcement: The TAMER framework. In *The Fifth International Conference on Knowledge Capture*.

Kobayashi, Y. and Kuno, Y. (2010). People tracking using integrated sensors for human robot interaction. In *IEEE International Conference on Industrial Technology*, pages 1617 –1622.

Konidaris, G. and Barto, A. (2006). Autonomous shaping: knowledge transfer in reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 489–496.

Kurniawati, H., Bandyopadhyay, T., and Patrikalakis, N. M. (2011). Global motion planning under uncertain motion, sensing, and environment map. In *Robotics: Science and Systems (RSS)*.

Kurniawati, H., Hsu, D., and Lee, W. S. (2008). SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems (RSS)*.

Lallée, S., Lemaignan, S., Lenz, A., Melhuish, C., Natale, L., Skachek, S., van der Zant, T., Warneken, F., and Dominey, P. F. (2010a). Towards a platform-independent cooperative human-robot interaction system: I. perception. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 4444–4451.

Lallée, S., Yoshida, E., Mallet, A., Nori, F., Natale, L., Metta, G., Warneken, F., and Dominey, P. F. (2010b). Human-robot cooperation based on interaction learning. In *From Motor Learning to Interaction Learning in Robots*, pages 491–536.

Lathan, C. E. and Tracey, M. (2002). The effects of operator spatial perception and sensory feedback on human-robot teleoperation performance. *Presence: Teleoper. Virtual Environ.*, 11(4):368–377.

LaValle, S. M. (2006). *Planning algorithms*. Cambridge University Press.

Lee, S. H., Kim, H. K., and Suh, I. H. (2011). Incremental learning of primitive skills from demonstration of a task. In *International Conference on Human-Robot Interaction (HRI)*, pages 185–186.

Lusena, C., Goldsmith, J., and Mundhenk, M. (2001). Nonapproximability results for partially observable markov decision processes. *Journal of Artificial Intelligence Research*, 14:2001.

MacDorman, K. F., Chalodhorn, R., and Asada, M. (2004). Periodic nonlinear principal component neural networks for humanoid motion segmentation, generalization, and generation. In *International Conference on Pattern Recognition (ICPR)*, pages 537–540.

Mahadevan, S., Marchalleck, N., Das, T. K., and Gosavi, A. (1997). Self-improving factory simulation using continuous-time average-reward reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 202–210. Morgan Kaufmann.

Mavridis, N., Giakoumidis, N., and Machado, E. (2012). A novel evaluation framework for teleoperation and a case study on natural human-arm-imitation through motion capture. *International Journal of Social Robotics*, 4:5–18.

Messing, R., Pal, C., and Kautz, H. A. (2009). Activity recognition using the velocity histories of tracked keypoints. In *International Conference on Computer Vision (ICCV)*, pages 104–111.

Mitchell, I., Bayen, A. M., and Tomlin, C. J. (2001). Validating a hamilton-jacobi approximation to hybrid system reachable sets. In *Hybrid Systems: Computation and Control*, pages 418–432. Springer Verlag.

Mitchell, I. M. (2007). A toolbox of level set methods. In *UBC Department of Computer Science Technical Report TR-2007-11*.

Mitchell, I. M., Bayen, A. M., and Tomlin, C. J. (2005). A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50:947–957.

Nemec, B., Vuga, R., and Ude, A. (2011). Exploiting previous experience to constrain robot sensorimotor learning. In *IEEE/RAS International Conf. on Humanoid Robots*.

Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*.

Ng, B., Meyers, C., Boakye, K., and Nitao, J. (2010). Towards applying interactive POMDPs to real-world adversary modeling. In *Innovative Applications of Artificial Intelligence*.

Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V. V. (2007). *Algorithmic Game Theory*. Cambridge University Press, New York, NY.

Nyga, D. and Beetz, M. (2012). Everything robots always wanted to know about housework (but were afraid to ask). In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Ojeda, L. and Borenstein, J. (2007). Personal dead-reckoning system for gps-denied environments. In *IEEE Int. Workshop on Safety, Security and Rescue Robotics*.

Ong, S. C. W., Png, S. W., Hsu, D., and Lee, W. S. (2009). Pomdps for robotic tasks with mixed observability. In *Robotics: Science and Systems (RSS)*.

Papadimitriou, C. and Tsitsiklis, J. N. (1987). The complexity of markov decision processes. *Mathematics of Operations Research*, 12:441–450.

Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 30(3):286–297.

Pastor, P., Hoffmann, H., Asfour, T., and Schaal, S. (2009). learning and generalization of motor skills by learning from demonstration. In *International Conference on Robotics and Automation (ICRA)*.

Pineau, J. and Gordon, G. (2005). Pomdp planning for robust robot control. In *International Symposium on Robotics Research*.

Pineau, J., Gordon, G., and Thrun, S. (2003). Point-based value iteration: An anytime algorithm for POMDPs. In *International Joint Conference on Artificial Intelligence*.

Porta, J. M., Vlassis, N., Spaan, M. T. J., and Poupart, P. (2006). Point-based value iteration for continuous pomdps. *J. of Machine Learning Research*, 7:2329–2367.

Riley, P. and Veloso, M. (2002). Planning for distributed execution through use of probabilistic opponent models. In *International Conference on AI Planning and Scheduling (AIPS)*, pages 72–81.

Riley, P., Veloso, M., and Kaminka, G. (2002). An empirical study of coaching. In *Distributed Autonomous Robotic Systems 5*, pages 215–224.

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–35.

Röfer, T., Laue, T., Müller, J., Fabisch, A., Feldpausch, F., Gillmann, K., Graf, C., de Haas, T. J., Härtl, A., Humann, A., Honsel, D., Kastner, P., Kastner, T., Könemann, C., Markowsky, B., Riemann, O. J. L., and Wenk, F. (2011). B-human team report and code release 2011. `http://www.b-human.de/downloads/bhuman11_coderelease.pdf`.

Rosenthal, S. and Veloso, M. (2011). Modeling humans as observation providers using pomdps. In *International Symposium on Robots and Human Interactive Communications (RO-MAN)*.

Schaal, S. and Atkeson, C. G. (1998). Constructive incremental learning from only local information. *Neural Computation*, 10(8):2047–2084.

Schrempf, O., Hanebeck, U., Schmid, A., and Worn, H. (2005). A novel approach to proactive human-robot cooperation. In *IEEE International Workshop on Robot and Human Interactive Communication*.

Short, E., Hart, J., Vu, M., and Scassellati, B. (2010). No fair!! an interaction with a cheating robot. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 219–226.

Shotton, J., Fitzgibbon, A. W., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., and Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304.

Shum, H. P. H., Komura, T., and Yamazaki, S. (2008). Simulating interactions of avatars in high dimensional state space. In *Symposium on Interactive 3D graphics and games*, I3D '08, pages 131–138. ACM.

Siegel, M., Breazeal, C., and Norton, M. I. (2009). Persuasive robotics: The influence of robot gender on human behavior. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Southey, F., Bowling, M., Larson, B., Piccione, C., Burch, N., Billings, D., and Rayner, C. (2005). Bayes' bluff: Opponent modelling in poker. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

Stone, P., Kaminka, G. A., Kraus, S., and Rosenschein, J. S. (2010). Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *National Conference on Artificial Intelligence (AAAI)*.

Stone, P. and Kraus, S. (2010). To teach or not to teach? decision making under uncertainty in ad hoc teams. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Stone, P., Sutton, R. S., and Kuhlmann, G. (2005). Reinforcement learning for RoboCup-soccer keepaway. *Adaptive Behavior*, 13(3):165–188.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Sutton, R. S., Precup, D., and Singh, S. P. (1998). Intra-option learning about temporally abstract actions. In *International Conference on Machine Learning (ICML)*.

Sutton, R. S., Precup, D., and Singh, S. P. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211.

Taha, T., Miró, J. V., and Dissanayake, G. (2011). A POMDP framework for modelling human interaction with assistive robots. In *International Conference on Robotics and Automation (ICRA)*, pages 544–549.

Tautges, J., Zinke, A., Krüger, B., Baumann, J., Weber, A., Helten, T., Müller, M., Seidel, H.-P., and Eberhardt, B. (2011). Motion reconstruction using sparse accelerometer data. *ACM Trans. Graph.*, 30(3):18:1–18:12.

Tedrake, R. (2009). LQR-trees: Feedback motion planning on sparse randomized trees. In *Robotics: Science and Systems (RSS)*.

Tenorth, M., Nyga, D., and Beetz, M. (2010). Understanding and executing instructions for everyday manipulation tasks from the world wide web. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Thrun, S. (2000). Monte carlo POMDPs. In *Neural Information Processing Systems (NIPS)*.

Tomlin, C. J., Lygeros, J., and Sastry, S. S. (2000). A game theoretic approach to controller design for hybrid systems. In *Proceedings of the IEEE*, pages 949–970.

Tomlin, C. J., Mitchell, I., Bayen, A. M., and Oishi, M. (2003). Computational techniques for the verification of hybrid systems. In *Proceedings of the IEEE*, pages 986–1001.

Valtazanos, A. (2012a). Bayesian interaction shaping: learning to influence strategic interactions in mixed robotic domains – supporting video, `http://www.youtube.com/watch?v=5rYVhHZzHQQ`.

Valtazanos, A. (2012b). Evaluating the effects of perceptual constraints on interactive decisions in mixed robotic environments – supporting video, `http://www.youtube.com/watch?v=6xi7WPgg46A`.

Valtazanos, A., Arvind, D. K., and Ramamoorthy, S. (2010). Comparative study of segmentation of periodic motion data for mobile gait analysis. In *ACM International Conference on Wireless Health*, pages 145–154.

Valtazanos, A., Arvind, D. K., and Ramamoorthy, S. (2013a). Latent space segmentation for mobile gait analysis. *ACM Trans. on Embedded Computing Systems*, 12(4).

Valtazanos, A., Arvind, D. K., and Ramamoorthy, S. (2013b). Using wearable inertial sensors for posture and position tracking in unconstrained environments through learned translation manifolds. In *ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*.

Valtazanos, A. and Ramamoorthy, S. (2011a). Intent inference and strategic escape in multi-robot games with physical limitations and uncertainty. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 3679–3685.

Valtazanos, A. and Ramamoorthy, S. (2011b). NaOISIS: a 3-D behavioural simulator for the NAO humanoid robot. In Roefer, T., Mayer, N. M., Savage, J., and Saranli, U., editors, *RoboCup-2011: Robot Soccer World Cup XV*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin.

Valtazanos, A. and Ramamoorthy, S. (2011c). Online motion planning for multi-robot interaction using composable reachable sets. In Roefer, T., Mayer, N. M., Savage, J., and Saranli, U., editors, *RoboCup-2011: Robot Soccer World Cup XV*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin.

Valtazanos, A. and Ramamoorthy, S. (2013a). Bayesian interaction shaping: learning to influence strategic interactions in mixed robotic domains. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Valtazanos, A. and Ramamoorthy, S. (2013b). Evaluating the effects of limited perception on interactive decisions in mixed robotic environments. In *International Conference on Human-Robot Interaction (HRI)*.

Vázquez, M., May, A., Steinfeld, A., and Chen, W.-H. (2011). A deceptive robot referee in a multiplayer gaming environment. In *International Conference on Collaboration Technologies and Systems (CTS)*, pages 204–211.

Vidal, R., Shakernia, O., Kim, H., Shim, D., and Sastry, S. (2002). Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *Trans. on Robotics and Automation*, 18(5):662 – 669.

Wagner, A. R. and Arkin, R. C. (2011). Acting deceptively: Providing robots with the capacity for deception. *International Journal of Social Robotics*, 3(1):5–26.

Wampler, K., Andersen, E., Herbst, E., Lee, Y., and Popović, Z. (2010). Character animation in two-player adversarial games. *ACM Trans. Graph.*, 29(3):26:1–26:13.

Wang, Z., Deisenroth, M. P., Amor, H. B., Vogt, D., Schölkopf, B., and Peters, J. (2012). Probabilistic modeling of human movements for intention inference. In *Robotics: Science and Systems (RSS)*.

Woodward, M. P. and Wood, R. J. (2012). Learning from humans as an I-POMDP. *Computing Research Repository (CoRR)*, abs/1204.0274.

Wunder, M., Kaisers, M., Yaros, J. R., and Littman, M. (2011). Using iterated reasoning to predict opponent strategies. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Yang, A., Iyengar, S., Sastry, S., Bajcsy, R., Kuryloski, P., and Jafari, R. (2008). Distributed segmentation and classification of human actions using a wearable motion sensor network. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Young, A. D. (2010). From posture to motion: the challenge for real time wireless inertial motion capture. In *International Conference on Body Area Networks (BodyNets)*, pages 131–137.

Young, A. D., Ling, M. J., and Arvind, D. K. (2007). *Orient-2*: a realtime wireless posture tracking system using local orientation estimation. In *Workshop on Embedded Networked Sensors (EmNets)*, pages 53–57.

Yun, X., Bachmann, E., Moore, H., and Calusdian, J. (2007). Self-contained position tracking of human movement using small inertial/magnetic sensor modules. In *International Conference on Robotics and Automation (ICRA)*, pages 2526 –2533.

Zhang, H. and Parkes, D. (2008). Value-Based Policy Teaching with Active Indirect Elicitation. In *National conference on Artificial intelligence (AAAI)*.

Zinkevich, M. (2012). The lemonade game competition, `http://tech.groups.yahoo.com/group/lemonadegame/`.