Intention Inference and Decision Making with Hierarchical Gaussian Process Dynamics Models

Inferenz von Intentionen zur Entscheidungsfindung mittels Hierarchischen Gaußprozess Dynamik-Modellen
Vom Fachbereich Informatik zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation von M.Eng. Zhikun Wang aus Fujian, China
September 2013 — Darmstadt — D 17

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Informatik

IAS

Intention Inference and Decision Making with Hierarchical Gaussian Process Dynamics Models
Inferenz von Intentionen zur Entscheidungsfindung mittels Hierarchischen Gaußprozess Dynamik-Modellen

Genehmigte Dissertation von M.Eng. Zhikun Wang aus Fujian, China

1. Gutachten: Prof. Dr. Jan Peters
2. Gutachten: Prof. Dr. Bernhard Schölkopf

Tag der Einreichung: July 28, 2013
Tag der Prüfung: September 17, 2013

Darmstadt — D 17

Erklärung zur Dissertation

Hiermit versichere ich, die vorliegende Dissertation ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Tübingen, den August 4, 2013

(Zhikun Wang)

Abstract

Anticipation is crucial for fluent human-robot interaction, which allows a robot to independently coordinate its actions with human beings in joint activities. An anticipatory robot relies on a predictive model of its human partners, and selects its own action according to the model's predictions. Intention inference and decision making are key elements towards such anticipatory robots. In this thesis, we present a machine-learning approach to intention inference and decision making, based on Hierarchical Gaussian Process Dynamics Models (H-GPDMs).

We first introduce the H-GPDM, a class of generic latent-variable dynamics models. The H-GPDM represents the generative process of complex human movements that are directed by exogenous driving factors. Incorporating the exogenous variables in the dynamics model, the H-GPDM achieves improved interpretation, analysis, and prediction of human movements. While exact inference of the exogenous variables and the latent states is intractable, we introduce an approximate method using variational Bayesian inference, and demonstrate the merits of the H-GPDM in three different applications of human movement analysis. The H-GPDM lays a foundation for the following studies on intention inference and decision making.

Intention inference is an essential step towards anticipatory robots. For this purpose, we consider a special case of the H-GPDM, the Intention-Driven Dynamics Model (IDDM), which considers the human partners' intention as exogenous driving factors. The IDDM is applicable to intention inference from observed movements using Bayes' theorem, where the latent state variables are marginalized out. As most robotics applications are subject to real-time constraints, we introduce an efficient online algorithm that allows for real-time intention inference. We show that the IDDM achieved state-of-the-art performance in intention inference using two human-robot interaction scenarios, i.e., target prediction for robot table tennis and action recognition for interactive robots.

Decision making based on a time series of predictions allows a robot to be proactive in its action selection, which involves a trade-off between the accuracy and confidence of the prediction and the time for executing a selected action. To address the problem of action selection and optimal timing for initiating the movement, we formulate the anticipatory action selection using Partially Observable Markov Decision Process, where the H-GPDM is adopted to update belief state and to estimate transition model. We present two approaches to policy learning and decision making, and show their effectiveness using human-robot table tennis.

In addition, we consider decision making solely based on the preference of the human partners, where observations are not sufficient for reliable intention inference. We formulate it as a repeated game and present a learning approach to safe strategies that exploit the humans' preferences. The learned strategy enables action selection when reliable intention inference is not available due to insufficient observation, e.g., for a robot to return served balls from a human table tennis player.

In this thesis, we use human-robot table tennis as a running example, where a key bottleneck is the limited amount of time for executing a hitting movement. Movement initiation usually requires an early decision on the type of action, such as a forehand or backhand hitting movement, at least 80 ms before the opponent has hit the ball. The robot, therefore, needs to be anticipatory and proactive of the opponent's intended target. Using the proposed methods, the robot can predict the intended target of the opponent and initiate an appropriate hitting movement according to the prediction. Experimental results show that the proposed intention inference and decision making methods can substantially enhance the capability of the robot table tennis player, using both a physically realistic simulation and a real Barrett WAM robot arm with seven degrees of freedom.

Zusammenfassung

Antizipation ist wichtig für eine flüssige Mensch-Roboter Interaktion, da sie es dem Roboter ermöglicht, seine Aktionen mit dem menschlichen Partner zu koordinieren. Dazu wird ein Modell benötigt, welches das Verhalten des Menschen vorhersagt. Entsprechend dieser Vorhersagen kann der Roboter Aktionen auswählen und durchführen. Intentionsinferenz und Entscheidungsfindung sind Schlüsselelemente solcher antizipierender Roboter. In dieser Dissertation stellen wir einen Ansatz aus der Theorie des maschinellen Lernens zur Intentionsinferenz und Entscheidungsfindung vor, der auf Hierarchischen Gaußprocess Dynamik-Modellen (H-GPDMs) basiert.

Dafür stellen wir zuerst H-GPDMs, eine Klasse von *latent-variable dynamics models*, vor. Ein H-GPDM ist ein generativer Prozess, welcher zur Modellierung komplexer menschlicher Bewegungen verwendet wird, die von exogenen Faktoren beeinflusst werden. Durch die direkte Einbeziehung der exogenen Variablen in die Modellierung, liefert das H-GPDM eine verbesserte Interpretation, Analyse und Prädiktion menschlicher Bewegungen. Da eine exakte Inferenz der exogenen Variablen und unbekannten Zustände (latent states) nicht möglich ist, stellen wir eine Approximationsmethode vor, die auf *variational Bayesian inference* basiert. Wir stellen die Vorzüge des H-GPDMs in drei unterschiedlichen Anwendungen heraus. Das H-GPDM legt den Grundstein für die Studien über Intentionsinferenz und Entscheidungsfindung in dieser Arbeit.

Intentionsinferenz ist ein wichtiger Schritt für antizipatorische Roboter. Aus diesem Grund betrachten wir einen Sonderfall des H-GPDMs, nämlich *Intention-Driven Dynamics Models* (IDDMs), welche die Intention des menschlichen Partners als exogene Variable betrachten. Das IDDM erschließt die unbekannte Intention aus Beobachtungen unter Verwendung der Bayesschen Regel, wobei die unbekannten Größen ausintegriert werden. Da viele Roboteranwendungen Echtzeitanforderungen unterliegen, stellen wir einen effizienten Online-Algorithmus vor, der Intentionsinferenz in Echtzeit ermöglicht. Wir zeigen, dass die Leistungsfähigkeit des IDDM auf dem neuesten Stand der Technik in Intentionsinferenz ist. Um dies zu verifizieren betrachten wir zwei Szenarien der Mensch-Roboter Interaktion: Zielprädiktion für Robotertischtennis und Verhaltenserkennung für interaktive Roboter.

Entscheidungsprozesse basierend auf Prädiktionszeitreihe ermöglichen dem Roboter seine Aktionen proaktiv auswählen. In diesem Fall muss er zwischen Genauigkeit, Prädiktionssicherheit und Dauer der auszuführenden Aktion abwägen. Wir behandeln dieses Problem im Rahmen von *Partially Observable Markov Decision Processes*, wobei wir das H-GPDM anpassen, um die *Belief States* zu schätzen und die Transitionsfunktion zu lernen. Wir präsentieren Ansätze zum Lernen einer Policy und für die Entscheidungsfindung. Die Effektivität dieser Ansätze verifizieren wir im Kontext von Mensch-Roboter Tischtennis.

Desweiteren betrachten wir Entscheidungsfindungen, die ausschließlich auf der Präferenz des menschlichen Partners basieren, da die Beobachtungen nicht für eine zuverlässige Intentionsinferenz ausreichen. Wir formulieren dieses Problem als Spiel und stellen einen Lernalgorithmus vor, der sichere Strategien unter Ausnutzung der menschlichen Präferenz lernt. Die gelernte Strategie kann zur Auswahl von geeigneten Aktionen verwendet werden, wenn keine zuverlässige Intentionsinferenz möglich ist. Dies kann zum Beispiel im Tischtennis der Fall sein, wenn ein Roboter Bälle zu einem Menschen zurückspielen soll.

In dieser Dissertation verwenden wir das Szenario des Mensch-Roboter Tischtennisspiels als durchgehendes Beispiel. Mensch-Roboter Tischtennis is ein sehr anspruchsvolles Beispiel, da die für die Schlagbewegung erforderliche Zeit begrenzt ist. Der Roboter muss teilweise die Schlagbewegung initiieren, bevor der Gegner den Ball überhaupt gespielt hat. Aus diesem Grund muss der Roboter antizipatorisch und proaktiv die Intention des Gegners erkennen. Mit visuellem Feedback von der Bewegung des Gegners kann der Roboter den Aufprallpunkt des Balles vorhersagen und dementsprechend eine Schlagbewegung auswählen, z.B. einen Vor- oder Rückhandschlag. Unsere Experimente belegen, dass unser Ansatz zur Intentionsinferenz und Entscheidungsfindung die Leistung des Roboters signifikant verbessert, wobei wir sowohl physikalisch realistische Simulationen als auch einen realen Barrett WAM Roboterarm mit sieben Freiheitsgraden verwenden.

Acknowledgements

I would like to thank Prof. Jan Peters for his excellent supervision, inspiring discussions, patience, and encouragement during the four years. I am deeply grateful to Prof. Bernhard Schölkopf for longstanding support of my research, and for providing an ideal environment at the Max Planck institute. I also appreciate the support from Prof. Stefan Schaal.

I am grateful to my thesis referees, Prof. Jan Peters and Prof. Bernhard Schölkopf for evaluating this thesis, to Prof. Michael Goesele for heading the thesis committee, and to Prof. Karsten Weihe and Prof. Jan Bender for reading this thesis and participating in the defense.

I am wholeheartedly appreciative to my main collaborators Abdeslam Boularias, Marc Deisenroth, and Kun Zhang, for their continuous support and valuable advice throughout the years. I learned a lot from them and it was a joy to work with them. I am very thankful to Katharina Mülling for assisting me with experiments in simulation and on the real robot. My former labmates Botond Bócsi, Jens Kober, Oliver Kroemer, Timm Meyer, and Duy Nguyen-Tuong have helped me with their support on projects, research collaborations, uncountable discussions, and, most importantly, valuable feedback. In addition, I thank many AGBS and IAS members for the inspiring discussions, some of which have led to successful projects.

Finally, I am grateful to Wenting Yang, for making my life in Tübingen a colorful experience.

Table of Contents

Abbreviations

In this thesis, we use the following mathematical notation.

Notation	Description
a, A	a scalar or discrete variable
$\mathbf{a} = [a_1, \ldots, a_n]$	a vector
$\mathbf{a}^T$	the transpose of a vector
$\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$	a matrix
$\mathbf{A}^T$	the transpose of a matrix
$\mathbf{a}^{-1}$	the inverse of a matrix
$p(x)$	probability density
$\mathbb{E}[x]$	expectation of x

Throughout this thesis, we use the following abbreviations.

Abbreviation	Description
EM	Expectation Maximization
GP	Gaussian Process
GPDM	Gaussian Process Dynamics Model
GPLVM	Gaussian Process Latent Variable Model
H-GPDM	Hierarchical GPDM
HMM	Hidden Markov Model
HRI	Human-Robot Interaction
IDDM	Intention-Driven Dynamics Model
KL	Kullback-Leibler divergence
MAE	Mean Absolute Error
MDP	Markov Decision Process
RBF	Radial Basis Function
POMDP	Partially Observable Markov Decision Process
RMSE	Root-Mean-Square Error
VB	Variational Bayesian inference

The following mathematical symbols are used in many chapters.

Symbol	Description
$\mathbf{z}_t$	observation in H-GPDMs or POMDPs at time t
$\mathbf{x}_t$	state in H-GPDMs at time t
g	intention, exogenous variables in H-GPDMs
$\mathbf{s}_t$	state in MDPs or POMDPs at time t
a_t	action in MDPs or POMDPs at time t
r_t	reward in MDPs or POMDPs at time t

1 Introduction

Human beings possess the ability to independently coordinate their actions with others in joint actions. *Anticipation* plays an important role in such emergent coordination, by integrating the prediction of the others' actions in one's own action planning (Sebanz et al., 2006). Similarly, anticipation is also a crucial component in competitive interaction between human beings (Parker et al., 2010). For example in competitive sports (Aglioti et al., 2008), "elite sports performance not only involves the ability to execute complex actions . . . , but also the ability to predict and anticipate the behavior of other players".

In artificial intelligence, anticipation has been introduced as a concept of an agent making decisions based on predictions, expectations, or beliefs about the future (Pezzulo et al., 2008). Based on these findings, it is reasonable to believe that such ability of anticipation can be equally important for human-robot interaction, in both competitive and cooperative contexts, where an intelligent robot would coordinate its actions in anticipation of the human beings' potential actions (Hoffman and Breazeal, 2007). For a hypothetical example, imagine that an autonomous vehicle needs to avoid an obstacle when planning its optimal moving path. The vehicle could simply find the shortest path such that no obstacle is on this path. Now imagine that the autonomous vehicle needs to avoid an pedestrian in motion, which is crucial for driving safely in an urban environment. The path planning not only needs to detect the current location of the pedestrian and moving direction but also her potential intention underlying her movement (Bandyopadhyay et al., 2013). Here, two problems arise, which are (1) the robot has to identify the intention and behavior of the pedestrian (Ziebart et al., 2009), and (2) the path planning needs to take into account the uncertainty in the predicted intention and future location (Bandyopadhyay et al., 2013). Therefore, building an autonomous vehicle that avoids an pedestrian in motion requires two key components, namely, prediction and proactive planning based on uncertain predictions.

In general, Rosen et al. (2012) considered an anticipatory system to be

> *a system containing a predictive model of itself and/or its environment, which allows it to change state at an instant in accord with the model's predictions pertaining to a later instant,*

which relies on two key components, namely, prediction of the environment (especially human partners) and planning according to the predictions. Inspired by the fact that "anticipation is a phenomenon which is learned" for human beings (Davies and Armstrong, 1989), we focus on developing *machine learning* approaches for anticipatory robots, specifically for intention inference and decision making. We first propose machine learning methods for inferring the human partners' intention and predicting their behavior. Subsequently, we introduce approaches to proactive action selection based on the uncertain predictions.

To motivate and evaluate the proposed ideas for anticipatory systems, we use human-robot table tennis as a running example. Playing table tennis against human beings is a challenging task for robots, and, hence, has been used by many researchers as a benchmark task in robotics (Anderson, 1988; Billingsley, 1984; Fässler et al., 1990; Matsushima et al., 2005; Mülling et al., 2011). Up to now, none of the groups that have been working on robot table tennis ever reached levels of a young child, despite having robots with better perception, processing power, and accuracy than humans (Mülling et al., 2011). The human's ability to predict hitting points from opponent movements is among the reasons for this performance gap (Mülling et al., 2011). In this thesis, we illustrate the effectiveness of intention inference and decision making to enhance the capability of a robot when playing against human players.

1.1 Intention Inference

Recent advances in sensors and algorithms allow for robots with improved perception abilities. For example, robots can now recognize human poses in real time using depth cameras (Shotton et al., 2013), which can potentially enhance their ability to interact with humans. However, effective perception alone is not sufficient for Human-Robot Interaction (HRI), since the robot's reactions should depend on comprehension of the human's action. An important comprehension problem is inferring the others' *intention* (also referred to as *goal, target, desire, plan*) (Simon, 1982), which humans heavily rely on, for example, in sports, games and social activities. Human can learn and improve the ability of prediction by training. For example, skilled tennis players are usually trained to possess better anticipation than amateurs (Williams et al., 2002). This observation raises the question of whether/how a robot can also learn to infer the underlying intention from a perceived ongoing human action.

We focus on intention inference from observed human movements, based on modeling how the dynamics of a movement are governed by the intention. This idea was inspired by the hypothesis that a human movement usually follows a goal-directed policy (Baker et al., 2006; Rao et al., 2004). To this end, we study a generic machine-learning approach to inferring the exogenous driving factors, e.g., the intention, from observed human movements. The resulting intention-driven dynamics model allows to estimate the probability distribution over intentions from observations using Bayes' theorem, and to update the belief as additional observations are obtained.

1.2 Decision Making

An anticipatory robot takes into account the prediction of the humans' intention for selecting its own action. One important problem for the anticipatory action selection is to deal with the uncertainty that naturally arose in prediction. As the input, such as observed human movement, is processed in a serial manner, the prediction usually becomes more accurate and less uncertain while more input is obtained. However, waiting for a confident prediction causes delay in action selection and results in reduced time for the robot to execute its action (Wang et al., 2013). The anticipatory robot is, thus, forced to make decisions given a sequence of uncertain predictions, where a trade-off between prediction accuracy and reaction delay needs to be addressed. We formulate the decision making process using a Partially Observable Markov Decision Process (POMDP), and propose approaches to choosing the robot's optimal action and deciding the timing to initiate the action.

In addition, we consider decision making solely based on the preference of the human partners, where observations are not sufficient for reliable intention inference. Opponent modeling is a critical mechanism in the repeated interaction with human. It allows the robot to adapt its strategy in order to better respond to the presumed preferences of its opponents. Similarly, the decision making based on opponent modeling needs to take into account the uncertainty in the estimated predictive model while exploiting the opponent's weakness. To this end, we propose methods for learning strategies that balance the safety and exploitability.

1.3 Main Contributions

This thesis contributes to the development of intention inference and decision making towards anticipatory robots, by proposing both novel models and practical methods. Specifically, main contributions include (1) proposing Hierarchical Gaussian Process Dynamics Model (H-GPDM), a generic model that lays the foundation for intention inference and decision making, (2) developing a spectrum of approximate intention inference methods based on the H-GPDM, ranging from a variational Bayesian inference method to a real-time online inference method based on moment matching, (3) introducing approaches to anticipatory action selections, including policy learning and Monte-Carlo planning for decision making, and strategy learning to exploit the opponent's preference, and (4) presenting a prototype anticipatory robot table tennis player.

1.3.1 Hierarchical Gaussian Process Dynamics Model

Gaussian Process Dynamics Models (GPDM) are a flexible class of latent-variable models, playing an increasingly important role in computer vision, robotics, and signal processing. GPDMs can represent complex nonlinear human movements where the dynamics in the state space follow a Markov chain. In practice, the dynamics are often driven by other higher level exogenous factors, which, however, are not modeled by the GPDMs. For example, the dynamics of human poses are driven by exogenous factors, such as gaits, goals, or styles of the movements. Inferring these exogenous variables is beneficial in many scenarios; for instance, recognition of gait styles facilitates the prediction accuracy of subsequent poses and, hence, enhances the tracking of human movements. In addition, inferring the goal behind observed actions can potentially enhance the fluent interaction between a human and an intelligent system, e.g., in the context of cooperative and interactive game-playing or human-robot interaction.

We incorporate the exogenous variables in the dynamics model to improve the interpretation, analysis, and prediction of human movements. The resulting Hierarchical GPDMs (H-GPDM) represent the generative process of human movements that are driven by exogenous variables, which can be learned from data in both supervised and unsupervised settings. The H-GPDM lays the foundation for the following intention inference. Since exact inference is intractable in this model, we introduce a variational Bayesian approach to jointly inferring the latent states and the unknown exogenous variables. We demonstrate the merits of this model in three scenarios: (a) action recognition and pose prediction using motion capture data, (b) character recognition and recovery from handwriting trajectories, and (c) target prediction in human-robot table tennis.

1.3.2 Online Intention Inference

Intention inference is an essential step toward efficient human-robot interaction. For this purpose, we consider a special case of the H-GPDM, where the exogenous driving factor is the human partner's intention. The resulting Intention-Driven Dynamics Model (IDDM) probabilistically models the generative process of movements that are directed by the intention. The IDDM allows to infer the intention from observed movements using Bayes' theorem. The IDDM simultaneously finds a latent state representation of noisy and high-dimensional observations, and models the intention-driven dynamics in the latent states.

As most robotics applications are subject to real-time constraints, we develop an efficient online algorithm that allows for real-time intention inference. Two human-robot interaction scenarios, i.e., target prediction for robot table tennis and action recognition for interactive humanoid robots, are show that the proposed algorithms achieve the state-of-the-art performance in intention inference.

1.3.3 Anticipatory Action Selection

The anticipatory robot can predict the intention of its human partners and select its own actions according to the prediction. To address the problem of action selection and optimal timing, we formulate the anticipatory action selection as a Partially Observable Markov Decision Process, and present two approaches to policy learning and online planning. Experimental results using a simulated environment show that the proposed algorithms could substantially enhance the capability of the robot table tennis player.

Furthermore, we consider decision making according to the preference of the human partners. We formulate it as a repeated game and present a learning approach to safe strategies that exploit the humans' preferences. The learned strategy enables action selection when reliable intention inference is not available due to insufficient observation, e.g., for a robot to return served balls from a human table tennis player.

1.3.4 Anticipatory Robot Table Tennis Player

In this thesis, we use human-robot table tennis as a running example. We develop a proof-of-concept prototype system to illustrate the presented methods for anticipatory robots. Anticipation is necessary for the robot to have sufficient time to execute the hitting movement when playing against a human player. The anticipatory robot player, equipped with three pre-trained actions, i.e., default, forehand, and backhand hitting movements, can initiate the responding hitting movement before the opponent has hit the ball himself. The robot player, hence, benefits from the anticipation system with a substantially expanded hitting region that covers almost its entire accessible workspace. For a demonstration see

`http://robot-learning.de/Research/ProbabilisticMovementModeling`

1.4 Outline

The chapters in this thesis can largely be read independently but partially build upon results of the preceding chapters. Figure 1.1 illustrates the outline of this thesis and the dependencies between chapters.

Chapter 2 introduces the Hierarchical Gaussian Process Dynamics Model (H-GPDM), which is the foundation for the following intention inference and decision making methods. The H-GPDM is a generic model for exogenous driving factors in the dynamics. Particularly, we show the merit of the H-GPDM in the context of human movement analysis and prediction, for example, for pose prediction, handwriting character recognition, and goal prediction. This chapter is based on (Wang et al., submittedb).

Chapter 3 extends the results of the previous chapter and proposes real-time intention inference algorithms for human-robot interaction, based on the Intention-Driven Dynamics Model

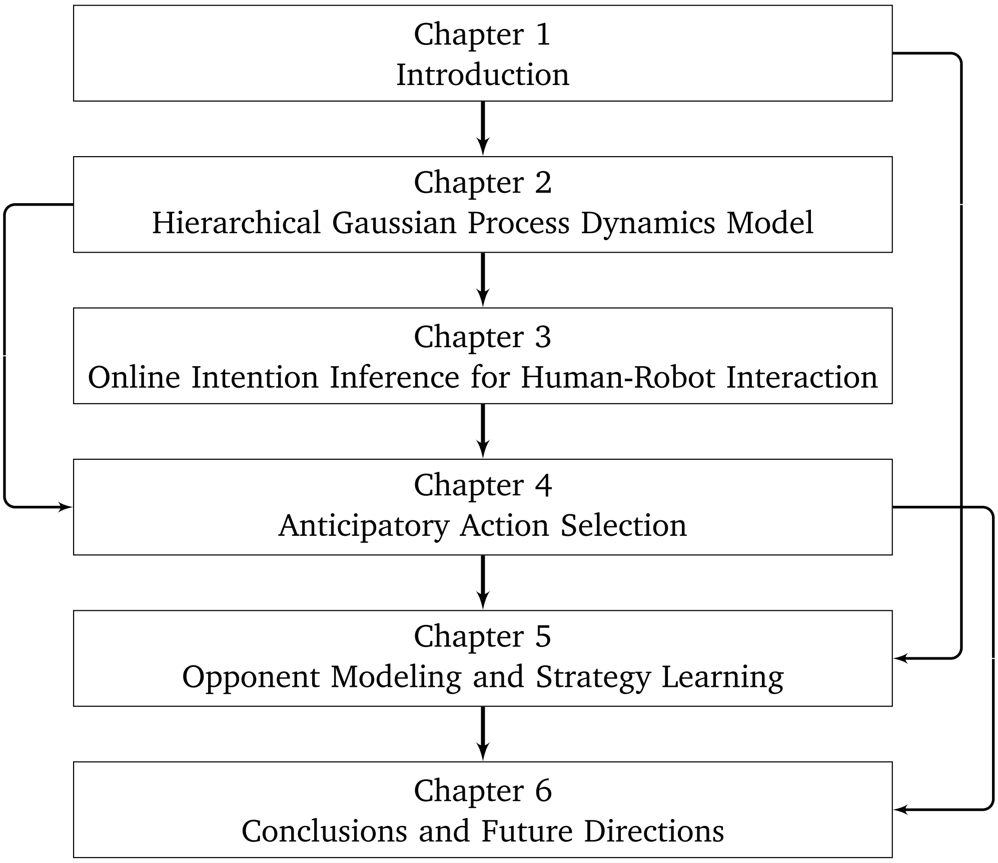

Chapter 1
Introduction
Chapter 2
Hierarchical Gaussian Process Dynamics Model
Chapter 3
Online Intention Inference for Human-Robot Interaction
Chapter 4
Anticipatory Action Selection
Chapter 5
Opponent Modeling and Strategy Learning
Chapter 6
Conclusions and Future Directions

Figure 1.1: Structure of this thesis and dependencies between chapters. Chapter 2 introduces the Hierarchical Gaussian Process Dynamics Model (H-GPDM), which is the foundation for the following intention inference methods. Chapter 3 proposes real-time intention inference algorithms for human-robot interaction. Chapter 4 discusses proactive action selection based on the uncertain predictions of the human's intention. Chapter 5 presents a strategy learning approach to exploiting the human opponent's preferences. Chapter 6 summarizes the content of this thesis and provides an outlook on future directions.

(IDDM), which is a special case of the H-GPDM. We evaluate the online inference algorithm in two scenarios, i.e., target prediction in human-robot table tennis and action recognition for building an anticipatory humanoid robot. This chapter is based on (Wang et al., 2012b, 2013).

Chapter 4 discusses the proactive action selection based on the uncertain predictions of the human partner. We formulate the proactive action selection as an optimal stopping problem using a Partially Observable Markov Decision Process (POMDP). We present two approaches to decision making, i.e., policy learning using model-free reinforcement learning, and Monte-Carlo planning based on the IDDM. This chapter is based on (Wang et al., 2011b) and (Wang et al., submitteda).

Chapter 5 presents strategy learning approach to exploiting the human opponent's preferences, which takes into account the trade-off between exploitability and uncertainty in the opponent model. This strategy is a complement to the intention inference and decision making, providing a solution when only the prior information of the human partner's preference is available, for example, for a robot to return served balls from a human table tennis player. This chapter is based on Wang et al. (2011a).

Chapter 6 summarizes the content of this thesis and provides an outlook on future directions.

2 Hierarchical Gaussian Process Dynamics Model

In this chapter, we introduce the Hierarchical Gaussian Process Dynamics Model (H-GPDM), which is the foundation for the following intention inference methods. The H-GPDM is a generic model for exogenous driving factors in the dynamics. Particularly, we show the merit of the H-GPDM in the context of human movement analysis and prediction, for example, for pose prediction, handwriting character recognition, and goal prediction. This chapter is based on (Wang et al., submittedb).

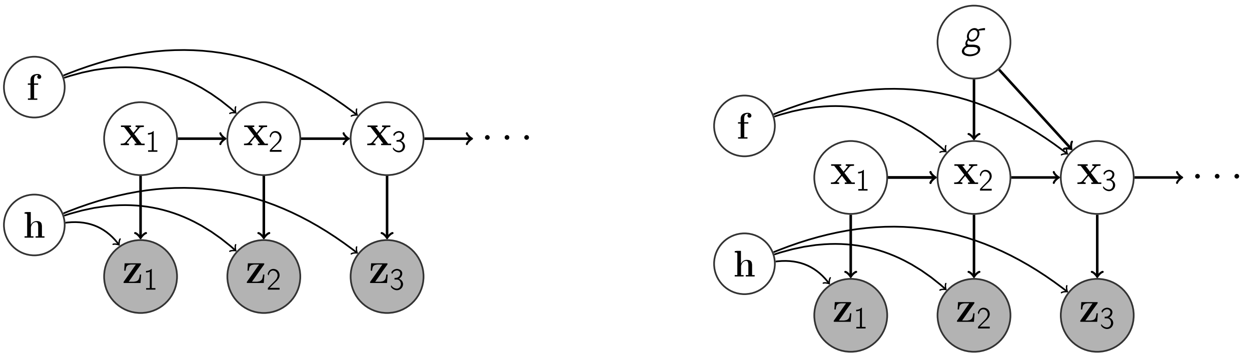
2.1 Prologue

Human movements are usually driven by factors such as goals, gait or motion styles, which we refer to as *exogenous variables*. These exogenous variables are not explicitly measurable but have considerable impact on the dynamics. Inferring such exogenous variables is highly beneficial in many scenarios. For example, recognition of action styles improves the prediction accuracy of subsequent poses (Fleet, 2011), which enhances the tracking of human movements (Urtasun et al., 2006). In human-robot interaction, anticipation of the human's goal, such as a target to reach or an object to hit, may allow the robot to act in a proactive manner (Wang et al., 2012b). Distinguishing different types of activities can also enhance the performance of multi-view tracking (Yao et al., 2011).

Inference of exogenous variables can be achieved by learning a *dynamics model* that captures how the exogenous variables affect the dynamics of human poses. While current technology allows for real-time tracking of human poses (Shotton et al., 2013), the obtained observations, e.g., a stream of skeleton joints, usually remain rather noisy or high-dimensional. State-space dynamics models, as shown in Figure 2.1a, provide a sensible approximation to the generative process of observed human movements $\mathbf{z}_t$ by modeling a measurement function $\mathbf{h}$ that generates observations from states and a transition function $\mathbf{f}$ that governs the dynamics in the state space. However, the corresponding latent state $\mathbf{x}_t$ of a human movement is not directly observed and even lacks a clear interpretation. Lawrence (2005) proposed to learn the *latent state* variables that are most likely to generate the observations with Gaussian Process Latent Variable Model (GPLVM).

Designing a generic parametric model for dynamics, such as autoregressive and moving average models (Veeraraghavan et al., 2005), is difficult due to the complexity of human movements, e.g., its nonlinear and stochastic nature. To address this issue, Bayesian nonparametric models, in particular Gaussian processes (GP), see (Rasmussen and Williams, 2006), have been successfully applied to modeling human movements. For example, Wang et al. (2008) proposed Gaussian Process Dynamics Models (GPDM) that use GPs for modeling the transition in the latent state space, which serves as a temporal prior leading to a hierarchical GPLVM (Lawrence and Moore, 2007). Note that the GPDM can also be seen as a nonparametric state-space model shown in Figure 2.1a, where the transition function $\mathbf{f}$ and measurement function $\mathbf{h}$ are given GP priors and marginalized out.

However, GPDMs do not incorporate exogenous driving factors that are not measurable. Such exogenous variables violate the Markov assumption in GPDMs, as the dynamics of latent states depend not only on the latent state variables but also on the exogenous variables. In this thesis, we generalize the GPDM with a new layer of exogenous variables as additional inputs to the



(a) GPDM

(b) H-GPDM

Figure 2.1: Graphical models of (a) the Gaussian process dynamics model (GPDM) and (b) the considered Hierarchical GPDM (H-GPDM). The considered model explicitly incorporates the exogenous variables as an input to the transition function. Here, we use gray nodes to represent the observed variables.

transition model, and introduce the Hierarchical GPDM (H-GPDM). The H-GPDM is shown in Figure 2.1b, in which the the transition in the latent states is driven by the exogenous factor g. Such an exogenous factor can, for example, represent the gait dynamics during walking or goal tha directs a hitting movement. The H-GPDM simultaneously finds latent states that represent the poses and describes the gait-specific or goal-directed transition in the latent state space.

Hierarchical dynamics models allow inferring the unobserved exogenous variable g from a time series of observations $\mathbf{z}_{1:T}$ by computing the posterior distribution $p(g|\mathbf{z}_{1:T})$, where the latent state variables $\mathbf{x}_{1:T}$ are marginalized out. Nevertheless, we are also interested in the inference of latent state variables, since they allow predicting future or missing observations. The use of a GP transition model renders exact inference intractable. Therefore, we consider the Variational Bayesian (VB) inference framework (Wainwright and Jordan, 2008) and propose an approximate inference method. We adopt the mean-field approximation, i.e., we approximate the posterior of latent states $\mathbf{x}_{1:T}$ by a factorized distribution. The resulting approximation yields an estimate of the marginal likelihood for the exogenous variable g for a segmented movement.

This chapter is organized as follows. We first discuss the inference of exogenous variables for human movement analysis scenarios and corresponding related work in the remainder of this section. We present the dynamics model with exogenous variables and show how the model can be learned from data in both supervised and unsupervised settings in Section 2.2. We propose a variational Bayesian method for inference in H-GPDM with exogenous variables in Section 2.3. We verify the feasibility of the proposed methods in multiple scenarios, and compare the experimental results to GPDM in Section 2.4. Finally, we conclude and discuss future directions in Section 2.5.

2.1.1 Considered Scenarios

In this chapter, we focus on developing methods for modeling and inferring of exogenous variables that are applicable to human movements. As a generic model, the H-GPDM can model the exogenous variables such as the intention, goal, or gait style of human movements, which are important for interpretation, analysis, and prediction of human movements. In this chapter, we focus on two main types of applications:

Pose Prediction. Unobserved exogenous variables increase the uncertainty on the human movements. For example, there may exist a range of underlying movement templates corresponding to different styles in a single trajectory, e.g., during walking. For improved interpretation and prediction of the human movements, it is beneficial to model the unobserved style variable as exogenous driving factors of the dynamics. Therefore, we advocate explicit modeling and inference of the exogenous variables for improved interpretation of modeled human movements. For example, it is natural to simultaneously recognize the type of gait and forecast subsequent poses when tracking a walking subject.

Goal inference. Efficient Human-Robot Interaction (HRI) and Human-Computer Interaction (HCI) depend on a comprehension of the human's action. Inferring the *goal* (also referred to as *intention, target, desire, or plan*) of the human (Simon, 1982), which can not be directly measured, can be crucial in HRI or HCI. Such goals can be seen as exogenous variables that governs the dynamics of the human movements. For example, a human movement usually follows a goal-directed policy (Baker et al., 2009; Friesen and Rao, 2011). The H-GPDM, shown in Figure 2.1b, models how the movements of the human partner is driven by the goal, and can hence infer the goal from an incomplete movement. This goal inference allows intelligent systems to interact with humans in a proactive manner.

2.1.2 Related Work

Our work relies on modeling human movements with nonlinear dynamics models, especially the Gaussian Process Dynamics Models (Wang et al., 2008). Inference in such models is a difficult problem due to the nonparametric transition and measurement model.

Nonlinear Dynamics Models

Observations of human actions frequently consist of continuous and high-dimensional features. Determining a low-dimensional latent state space is important for understanding, interpreting, and visualizing observed actions. The Gaussian process latent variable model (Lawrence, 2005) finds maximum likelihood latent variables by marginalizing the function that maps from latent to observed space. Its extension, the GPDM (Wang et al., 2008) can be used to model the dynamics of human motion while simultaneously finding low-dimensional latent states. A GPDM can model and extrapolate the appearance of a human or an object in motion, and, hence, is also helpful in tracking, for example, a small robotic blimp using two cameras (Ko and Fox, 2009). In follow-up work, the model was learned using the GPDM (Ko and Fox, 2011), such that the latent states did not need to be provided for learning.

Approximate Inference in GPDMs

Inference in GPDMs aims to update the belief about latent variables given evidence from a set of observations. In many applications, the inference method finds a Maximum a Posterior (MAP) estimate of the latent state variables $\mathbf{x}_{1:T}$ from observations (Wang et al., 2008; Urtasun et al., 2006) using gradient descent (Wang et al., 2008; Urtasun et al., 2006). The MAP estimate of latent state worked well for tracking in the absence of exogenous variables (Urtasun et al., 2006; Yao et al., 2011). However, inference of exogenous variables often requires to marginalize out the uncertainty in the latent states (Wang et al., 2012b), given by

$$p(g|\mathbf{z}_{1:T}) = \int p(g|\mathbf{x}_{1:T})p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T})d\mathbf{x}_{1:T}.$$

Unfortunately, the posterior distributions on latent state variables $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)$ are non-Gaussian for the general nonlinear GP transition model (Girard et al., 2002). As a result, exact inference is not tractable as the state variables $\mathbf{x}_{1:T}$ cannot be marginalized analytically. Besides methods based on sampling, e.g., particle filters (Ko and Fox, 2009), a straightforward approach to tackling this problem is approximating the posterior distribution by a Gaussian distribution $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g) \approx q(\mathbf{x}_{1:T})$. Such approximations have been proposed in previous work with extended Kalman filters (Ko and Fox, 2009), unscented Kalman filters (Ko and Fox, 2009), assumed density filters (Deisenroth et al., 2009), and general forward-backward smoothers in the context of GP dynamics models (Deisenroth et al., 2012).

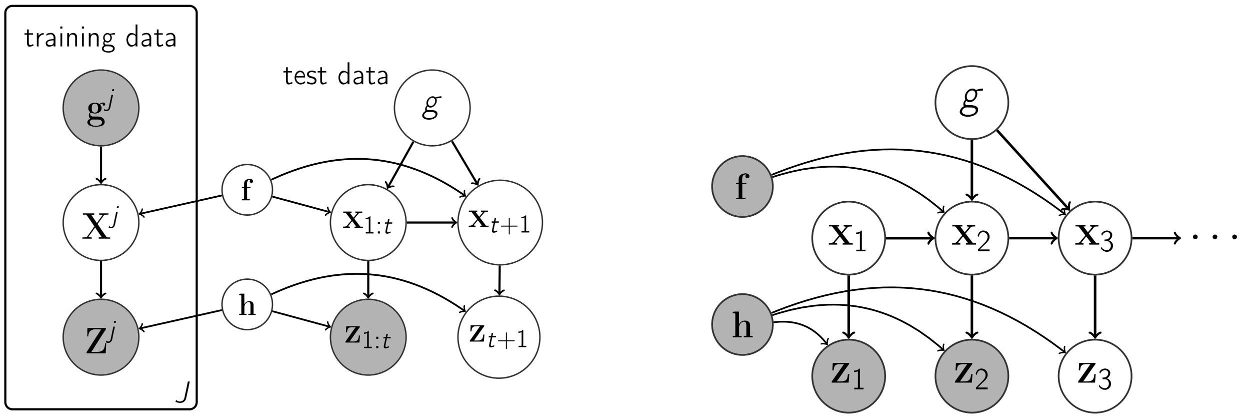
An alternative approach is to introduce a set of inducing variables (Quiñonero-Candela and Rasmussen, 2005), and adopt Fully Independent Training Conditional (FITC) approximation (Snelson and Ghahramani, 2006) to achieve Bayesian inference for the latent states (Damianou et al., 2011). Damianou et al. (2011) used an additional simplification that the dynamics only depend on time, resulting in a tree-structured graphical model. Such simplifications are not applicable in the considered scenarios, as the dynamics of human movements are often not deterministic in time.

Goal Inference in Temporally Evolving Models

Goal inference has been investigated in different settings. Most of previous work relies on probabilistic reasoning (e.g., Pentland and Liu, 1999; Liao et al., 2007; Rao et al., 2004; Baker et al., 2006; Friesen and Rao, 2011).

Goal inference with finite sets of states and actions has been studied in many settings. For example, an early approach (Pentland and Liu, 1999) used Hidden Markov Models (HMMs) to model and predict human behavior where different dynamics models were adopted to the corresponding behaviors. Probabilistic approaches to plan recognition in artificial intelligence (Liao et al., 2007) typically represent plans as policies in terms of state-action pairs. When the goal is to maximize an unknown utility function, inverse reinforcement learning (Abbeel and Ng, 2004) can be used to infer the underlying utility function from an agent's behavior, assuming that rational agents always maximize their expected utility. All these methods rely on defining a parametric model, which is often hard for human movement.

In cognitive science, Bayesian models are used for inferring goals from behavior (Rao et al., 2004), where a policy conditional on the agent's goal is learned to represent the behavior. Bayesian models can also interpret the agent's behavior and predict its behavior in a similar environment with the learned model (Baker et al., 2006). Recently, a computational framework was proposed to model gaze following (Friesen and Rao, 2011), where GPs are used to model the dynamics with actions driven by a goal. These methods assume that the states can be observed. However, the states are not observable for human movement.



(a) complete H-GPDM model

(b) fully independent test conditional

Figure 2.2: Graphical models of (a) the complete H-GPDM for inference, and (b) the approximation using the fully independent (test) conditional (FIC).

Human Motion Models

Recent techniques for monocular pose tracking often use activity-specific motion models from human motion capture data. In the past, most successful models have adopted probabilistic frameworks.

The GPDM (Wang et al., 2008) and its extensions are among the most widely used nonparametric dynamics models. One interesting extension is to impose additional structure constraints on the latent state space. For example, back-constraints ensure that similar poses are generated from close latent states (Urtasun et al., 2008).

The Switching Linear Dynamical System (Li et al., 2012) is a collection of linear dynamics models along with a discrete switching variable, so that it has the potential to model diverse styles and actions. Multi-factor GPLVMs and GPDMs (Wang et al., 2007) model individual styles factor on the measurement mapping, using side information such as gait type and subject identity. However, they do not take into account the effects of styles on the transition function.

We refer to Fleet (2011) for a comprehensive review on probabilistic motion models for tracking.

2.2 Hierarchical Gaussian Process Dynamics Model

We describe the Hierarchical Gaussian Process Dynamics Model (H-GPDM) in this section. This model is an extension of the GPDM (Wang et al., 2008), as shown in Figure 2.1a, in which the dynamics are fully determined by the latent states. In this thesis, we consider a layer of exogenous variables that drive the dynamics, as shown in Figure 2.1b. For example, the gait dynamics of walking can have substantial variations for different people or for different walking styles. These variations are associated to the exogenous variables. To model and infer these variables, we explicitly incorporate them in the H-GPDM, as shown in Figure 2.2a.

This chapter extensively uses known properties of Gaussian processes, e.g., predictive distribution and marginal likelihood. For readers who are not familiar with these properties, we refer to Rasmussen and Williams (2006) for a comprehensive introduction to GPs.

2.2.1 Model Description

The H-GPDM describes the generative process of an observed movement as follows (see Figure 2.1b for the graphical model of a single episode).

1. Sample the exogenous variable $g \sim p(g)$.
2. Start from an initial state $\mathbf{x}_1 \sim \mathcal{N}(\mathbf{m}_g, \mathbf{S}_g)$ according to the exogenous variable g.
3. For each time index $t \in \{1, \dots, T\}$:
   a) Sample an observation $\mathbf{z}_t \sim p(\mathbf{z}_t|\mathbf{x}_t)$ according to a measurement model.
   b) Transition to a new state $\mathbf{x}_{t+1} \sim p(\mathbf{x}_{t+1}|\mathbf{x}_t, g)$ following the transition model.

For notation simplicity and without loss of generality, we assume that all observed movements have the same duration T. and that the exogenous variable g is either discrete or univariate.

Measurement model. The observations of a movement are a time series $\mathbf{z}_{1:T} \triangleq [\mathbf{z}_1, \dots, \mathbf{z}_T]$, where $\mathbf{z}_t \in \mathbb{R}^{D_z}$. In the H-GPDM, an observation $\mathbf{z}_t$ is generated from the latent state $\mathbf{x}_t \in \mathbb{R}^{D_x}$, given by

$$\mathbf{z}_t = \mathbf{W}^{-1}(\mathbf{h}(\mathbf{x}_t) + \mathbf{n}_{z,t}), \quad \mathbf{n}_{z,t} \sim \mathcal{N}(\mathbf{0}, s_z^2\mathbf{I}),$$

where $\mathbf{W} = \mathrm{diag}(w_1, \dots, w_{D_z})$ scales the obtained measurements $\mathbf{z}_t$. The scaling parameters $\mathbf{W}$ allow for dealing with raw features that are measured in different units, such as positions and velocities. We place a GP prior distribution on the unknown function $\mathbf{h}$ for every dimension of the measurement $\mathbf{z}_t$, which is marginalized out during learning and inference. The GP prior $\mathcal{GP}(m_z(\cdot), k_z(\cdot,\cdot))$ is fully specified by a mean function $m_z(\cdot)$ and a covariance (kernel) function $k_z(\cdot,\cdot)$. For simplicity, we use GP prior mean functions that are zero everywhere, i.e., $m_z(\cdot) \equiv 0$. Hence, the model is fully determined by the covariance function $k_z(\cdot,\cdot)$. Without additional specific prior knowledge on the latent state space, we can only use the same covariance function for the GP prior on every dimension of the unknown measurement function $\mathbf{h}$, and use a scalar matrix $s_z^2\mathbf{I}$ for the noise covariance.

The covariance function k_z for the measurement mapping from the state space to observed space is chosen depending on the task. In this thesis, we consider two types of covariance functions. One considered covariance function is an isotropic Gaussian function

$$k_z(\mathbf{x}, \mathbf{x}'; \boldsymbol{\beta}) = \exp\left(-\frac{\beta_1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) + \beta_2\delta_{\mathbf{x},\mathbf{x}'}, \tag{1}$$

parameterized by the hyperparameters $\boldsymbol{\beta} = (\beta_1, \beta_2)$. Here δ is the Kronecker delta function. Intuitively, the latent states that generate human poses lie on a nonlinear manifold, requiring nonlinear covariance functions to model this relationship appropriately (Lawrence, 2005). Here, we do not parameterize the signal variance in the covariance function, due to the presence of the scaling parameters $\mathbf{W}$, but only parameterize the noise-signal ratio as β_2.

We also consider the linear covariance function

$$k_z(\mathbf{x}, \mathbf{x}'; \boldsymbol{\beta}) = \mathbf{x}^T\mathbf{x}' + \beta_1\delta_{\mathbf{x},\mathbf{x}'}, \tag{2}$$

when the measurements are already informative and low-dimensional, but subject to substantial noise. Some empirical comparisons of different covariance functions for the measurement mapping, e.g., the isotropic Gaussian in Eq. (1) and the linear covariance function in Eq. (2) can be found in the literature (Wang et al., 2012b).

Transition model. We consider a first-order Markov transition model (see Figure 2.2a) with a latent transition function $\mathbf{f}$, such that

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, g) + \mathbf{n}_{x,t}, \quad \mathbf{n}_{x,t} \sim \mathcal{N}(\mathbf{0}, s_x^2\mathbf{I}).$$

The state $\mathbf{x}$ at time $t+1$ depends on the latent state at time t as well as on the exogenous variable g, for example the gait style. We place a GP prior $\mathcal{GP}(0, k_x(\cdot,\cdot))$ on $\mathbf{f}$ for every dimension of the state $\mathbf{z}_{t+1}$, which has a zero mean function and a shared covariance function. The transition function $\mathbf{f}$ may also depend on external inputs, e.g., motor commands or controls. We assume that the external inputs are always observed and omit them for notational simplicity. The described methods can be straightforwardly adapted to situations with external inputs.

The underlying dynamics of human motion are usually nonlinear. For example, motions such as jumping or walking cannot be modeled well with a linear dynamic function. To account for nonlinearities, we add a Gaussian covariance function with Automatic Relevance Determination (ARD) to the linear covariance function for the dynamics, i.e.,

$$k_x([\mathbf{x}, g], [\mathbf{x}', g']; \boldsymbol{\alpha}) = \alpha_1 \exp\left(-\tfrac{\alpha_2^x}{2}\|\mathbf{x}-\mathbf{x}'\|^2 - \tfrac{\alpha_2^g}{2}\|g-g'\|^2\right) + \alpha_3^x \mathbf{x}^T\mathbf{x}' + \alpha_3^g g g' + \alpha_4 \delta_{\mathbf{x}\mathbf{x}'}\delta_{gg'},$$

for continuous g and

$$k_x([\mathbf{x}, g], [\mathbf{x}', g']; \boldsymbol{\alpha}) = \delta_{gg'}\left[\alpha_1 \exp\left(-\tfrac{\alpha_2}{2}\|\mathbf{x}-\mathbf{x}'\|^2\right) + \alpha_3 \mathbf{x}^T\mathbf{x}' + \alpha_4 \delta_{\mathbf{x}\mathbf{x}'}\right],$$

for discrete g, where $\boldsymbol{\alpha}$ is the set of all hyperparameters. For discrete g, we consider no covariance among different values of g. We assume that the gait dynamics are unrelated between distinct walking styles, as mixing different styles lead to gaits with “insufficient” energetic cost (Farley and McMahon, 1992). Nonetheless, the proposed methods can be extended straightforwardly to the cases where various gait dynamics are dependent.

2.2.2 Supervised Learning of the H-GPDM

We first consider learning the H-GPDM when the exogenous variables are provided in the training data. The training data set $\mathcal{D} = \{\mathbf{Z}, \mathbf{g}\}$ consists of J movements and corresponding labeled exogenous variables. Each movement's observations $\mathbf{Z}^j$ consist of a time series $\mathbf{Z}^j = [\mathbf{z}_1^j, \ldots, \mathbf{z}_T^j]^T$. We construct the overall observation matrix $\mathbf{Z}$ by vertically concatenating observation matrices $\mathbf{Z}^1, \ldots, \mathbf{Z}^J$, and the overall exogenous variables as a vector $\mathbf{g}$ from $g^1, \ldots, g^J$. The exogenous variables g can often be provided in the training data, e.g., by post-processing the data. The exogenous variables can also include side information (Wang et al., 2007), such as the type of gait (e.g., run, walk, jog) and the subject's identity, both of which are driving factors that contribute to the movement style. This side information is often available for motion capture data.

As shown in Figure 2.2a, we can infer the exogenous variable g given a new time series of measurements $\mathbf{z}_{1:t}$, as well as the subsequent measurement $\mathbf{z}_{t+1}$. Here, we use the superscript to index a segmented movement in the training data. Fully Bayesian inference is intractable, due to the nonlinear measurement functions $\mathbf{h}$ and transition function $\mathbf{f}$. As shown in Figure 2.2a, inference of the states on the test data $\mathbf{x}_{1:t}$ is difficult due to the marginalization of both the states on the training data $\mathbf{X}^j$ and the latent functions $\mathbf{h}$ and $\mathbf{f}$. One solution is to resort to a maximum a posterior (MAP) estimate of latent states, known as poor man's Bayesian inference (Tzikas et al., 2008). Wang et al. (2008) find the latent variables $\mathbf{x}^j$ for the training data and also the latent variables $\mathbf{x}_{1:t}$ for the test data. This method was later improved in (Ko and Fox, 2011; Deisenroth and Ohlsson, 2011; Deisenroth and Mohamed, 2012; Wang et al., 2012b, 2013), where the MAP estimation of states $\mathbf{X}^j$ was only used for the training data and the posterior distribution of states $\mathbf{x}_{1:t}$ was estimated on the test data. In this thesis, we follow this idea and find a MAP estimate of latent variables $\mathbf{X}^j$ only for the training data, as used by Wang et al. (2008).

The measurement and transition models are nonparametric and determined by the data. We find a MAP estimate of the latent states $\mathbf{X}^{\text{MAP}}$ and model parameters $\boldsymbol{\beta}^{\text{MAP}}$ and $\mathbf{W}^{\text{MAP}}$ in the training data, while the parameters $\boldsymbol{\alpha}$ are manually chosen to be $\bar{\boldsymbol{\alpha}}$ as suggested by (Wang et al., 2008). We maximize the joint posterior probability over latent states $\mathbf{X}$ and model parameters $\boldsymbol{\beta}$, and $\mathbf{W}$, given by

$$\left(\mathbf{X}^{\text{MAP}}, \boldsymbol{\beta}^{\text{MAP}}, \mathbf{W}^{\text{MAP}}\right) = \operatorname*{argmax}_{\mathbf{X},\boldsymbol{\beta},\mathbf{W}} p(\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}, \mathbf{Z}|\mathbf{g}). \tag{3}$$

This joint probability can be decomposed

$$p(\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}, \mathbf{Z}|\mathbf{g}) = p(\boldsymbol{\beta}, \mathbf{W})p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\beta}, \mathbf{W})p(\mathbf{X}|\mathbf{g}),$$

which is obtained by GP marginal likelihoods, i.e., marginalizing out the latent functions $\mathbf{f}$ and $\mathbf{h}$. We place a log-normal distribution prior on the parameters $\boldsymbol{\beta}$, and $\mathbf{W}$.

The GP marginal probability of the observations $\mathbf{Z}$ given the latent states $\mathbf{X}$ is given by a Gaussian distribution

$$p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}) = \frac{|\mathbf{W}|^M}{\sqrt{(2\pi)^{MD_z}|\mathbf{K}_z|^{D_z}}} \exp\left(-\frac{1}{2}\text{tr}\left(\mathbf{K}_z^{-1}\mathbf{Z}\mathbf{W}^2\mathbf{Z}^T\right)\right), \tag{4}$$

where M is the total number of observations $\mathbf{Z}$ and $\mathbf{K}_z$ the covariance matrix of $\mathbf{Z}$ computed by the covariance function $k_z(\cdot,\cdot)$.

Given the exogenous variables $\mathbf{g}$, the sequence of latent states $\mathbf{X}$ has a Gaussian distribution

$$p(\mathbf{X}|\mathbf{g}) = p(\mathbf{X}_1)p(\mathbf{X}_{2:T}|\mathbf{X}_{1:T-1}, \mathbf{g}) = \frac{p(\mathbf{X}_1)}{\sqrt{(2\pi)^{mD_x}|\mathbf{K}_x|^{D_x}}} \exp\left(-\frac{1}{2}\text{tr}\left(\mathbf{K}_x^{-1}\mathbf{X}_{2:T}\mathbf{X}_{2:T}^T\right)\right), \tag{5}$$

where $\mathbf{X}_{\text{indices}}$ is constructed by vertically concatenating state variables $\mathbf{x}^1_{\text{indices}}, \dots, \mathbf{x}^J_{\text{indices}}$, m is the length of $\mathbf{X}_{2:T}$, and $\mathbf{K}_x$ is the covariance matrix of $\mathbf{X}_{1:T-1}$ computed by the covariance function $k_x(\cdot,\cdot)$. We use a Gaussian prior distribution on the initial states $\mathbf{X}_1$.

Based on Eq. (4) and (5), the MAP estimate of the states and parameters is obtained by maximizing the posterior in Eq. (3), which is equivalent to minimizing the negative log-posterior

$$\begin{aligned}\mathscr{L}(\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}, \mathbf{g}) = {} & \frac{D_z}{2}\log|\mathbf{K}_z| + \frac{1}{2}\text{tr}\left(\mathbf{K}_z^{-1}\mathbf{Z}\mathbf{W}^2\mathbf{Z}^T\right) - M\log|\mathbf{W}| + \frac{D_x}{2}\log|\mathbf{K}_x| \\ & + \frac{1}{2}\text{tr}\left(\mathbf{K}_x^{-1}\mathbf{X}_{2:T}\mathbf{X}_{2:T}^T\right) + \frac{1}{2}\text{tr}\left(\mathbf{X}_1\mathbf{X}_1^T\right) + \frac{1}{2}\|\mathbf{log}\boldsymbol{\beta}\|^2 + \frac{1}{2}\|\mathbf{log}\,\text{diag}(\mathbf{W})\|^2 + \text{const}\end{aligned} \tag{6}$$

Algorithm 2.1: The unsupervised algorithm that discovers the latent driving factors $\mathbf{g}$ and learns the model parameters $\mathbf{X}^{\text{MAP}}, \boldsymbol{\beta}^{\text{MAP}}$, and $\mathbf{W}^{\text{MAP}}$. Step 1 is skipped when the exogenous variables $\mathbf{g}$ are provided. In the considered experiments in this chapter, we chose the number of iterations $I = 100$.

```
  Input  : Data: $\mathscr{D} = \{\mathbf{Z}\}$
  Input  : Number of iterations: $I$
  Output: Model parameters: $\boldsymbol{\Theta} = \{\mathbf{g}, \mathbf{X}, \boldsymbol{\beta}, \mathbf{W}\}$
1 for $i \leftarrow 1$ to $I$ do
2     for $j \leftarrow 1$ to $J$ do
3         For the $j$-th episode of movement, $g^j \leftarrow \text{argmax}_{g^j \in \{1,\dots,K\}} \mathscr{L}(\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}, \mathbf{g})$ ;
4     Minimize $\mathscr{L}(\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}, \mathbf{g})$ by optimizing $\mathbf{X}$, $\boldsymbol{\beta}$, and $\mathbf{W}$ ;
```

with respect to the states $\mathbf{X}$ and model parameters $\boldsymbol{\beta}$ and $\mathbf{W}$, using the Scaled Conjugate Gradient (SCG) method, for instance (Møller, 1993). Here, $\mathbf{log}\boldsymbol{\beta}$ denotes the componentwise log of the vector $\boldsymbol{\beta}$, and diag($\mathbf{W}$) denotes the vector that consists of the diagonal elements of $\mathbf{W}$.

The model also depends on the hyperparameter D_x, which is the dimension of the latent state space. The Bayesian GPLVM (Titsias and Lawrence, 2010) can compute an estimate of marginal likelihood of a specific dimensionality D_x, which could help select the proper dimensionality of latent states. One can also use model selection, for example based on cross-validation (Wang et al., 2012b).

2.2.3 Unsupervised Discovery of Latent Exogenous Variables

In practice, the exogenous variables may not be provided as side information in the training data. For example, defining styles and annotating them manually in human movements are often difficult due to the complexity and large variance of human motion. Distinguishing these styles can improve the interpretation of the observed movements and prediction subsequent poses for tracking. Therefore, we consider unsupervised learning of the H-GPDM by discovering latent exogenous factors to improve the interpretation of the observed movements $\mathscr{D} = \{\mathbf{Z}\}$. Here, we assume that the exogenous variable is discrete and time-invariant for each episode of movement. For example, the gait dynamics can be assumed static during a segmented walking movement for a specific person.

Based on the H-GPDM, we can discover the exogenous variables from episodic movements. We expect that the discovered driving factors $\mathbf{g}$ can best describe the generative process of the observed movements and, hence, maximize the marginal likelihood $p(\mathbf{Z}|\mathbf{g})$. As the computation of the marginal likelihood $p(\mathbf{Z}|\mathbf{g})$ is intractable, we instead maximize the joint probability $p(\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}, \mathbf{Z}|\mathbf{g})$ with respect to the exogenous variables $\mathbf{g}$, which is also the objective for learning the latent states and model parameters. To this end, we alternate between (1) optimizing the exogenous variables g^j for one movement at a time and (2) optimizing the states $\mathbf{X}$ and model parameters $\boldsymbol{\beta}$ and $\mathbf{W}$, by minimizing the negative log-posterior $\mathscr{L}(\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}, \mathbf{g})$ in Eq. (6). This learning approach, summarized in Algorithm 2.1, monotonically increases the posterior $p(\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}, \mathbf{Z}|\mathbf{g})$, and will converge to a local optimum.

To summarize, the model $\mathcal{M} = \{\mathbf{g}, \mathbf{X}^{\text{MAP}}, \boldsymbol{\beta}^{\text{MAP}}, \mathbf{W}^{\text{MAP}}, \bar{\boldsymbol{\alpha}}\}$ can be learned from the data set $\mathcal{D}$ even when the exogenous variables are not provided.

2.3 Approximate Inference for H-GPDM

We introduce the Variational Bayesian (VB) inference method for the H-GPDM in this section. The model $\mathcal{M} = \{\mathbf{g}, \mathbf{X}^{\text{MAP}}, \boldsymbol{\beta}^{\text{MAP}}, \mathbf{W}^{\text{MAP}}, \bar{\boldsymbol{\alpha}}\}$ learned from the data set $\mathcal{D}$, is omitted hereafter for notational simplicity.

As discussed in Section 2.2.2, we first compute the MAP estimate of latent states for the training data. The corresponding posterior of the measurement function $\mathbf{h}$ is subsequently determined by the obtained MAP estimate $\mathbf{X}^{\text{MAP}}$, and the predictive probability of the observations $\mathbf{z}_t$ is given by a Gaussian distribution $\mathbf{z}_t \sim \mathcal{N}(\mathbf{m}_z(\mathbf{x}_t), \sigma_z^2(\mathbf{x}_t)\mathbf{I})$. The predictive mean and variance are given by

$$\mathbf{m}_z(\mathbf{x}_t) = \mathbf{Z}\mathbf{K}_z^{-1}\mathbf{k}_z(\mathbf{x}_t),$$

and

$$\sigma_z^2(\mathbf{x}_t) = k_z(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{k}_z(\mathbf{x}_t)^T\mathbf{K}_z^{-1}\mathbf{k}_z(\mathbf{x}_t), \tag{7}$$

where $\mathbf{K}_z \triangleq k_z(\mathbf{X}^{\text{MAP}}, \mathbf{X}^{\text{MAP}})$ is the covariance matrix for the MAP estimate of training states. Here, we use the shorthand notation $\mathbf{k}_z(\mathbf{x}_t)$ to represent the cross-covariance vector between $\mathbf{h}(\mathbf{X}^{\text{MAP}})$ and $\mathbf{h}(\mathbf{x}_t)$.

Similarly, the predictive distribution of the latent state $\mathbf{x}_{t+1}$ conditioned on $\mathbf{x}_t$ and the exogenous variable g is a Gaussian distribution given by $\mathbf{x}_{t+1} \sim \mathcal{N}(\mathbf{m}_x(\mathbf{x}_t, g), \sigma_x^2(\mathbf{x}_t, g)\mathbf{I})$ with

$$\mathbf{m}_x(\mathbf{x}_t, g) = \mathbf{X}_{2:T}^{\text{MAP}}\mathbf{K}_x^{-1}\mathbf{k}_x(\mathbf{x}_t, g)$$

and

$$\sigma_x^2(\mathbf{x}_t, g) = k_x([\mathbf{x}_t, g], [\mathbf{x}_t, g]) - \mathbf{k}_x(\mathbf{x}_t, g)^T\mathbf{K}_x^{-1}\mathbf{k}_x(\mathbf{x}_t, g),$$

where $\mathbf{K}_x \triangleq k_x(\mathbf{X}_{1:T-1}^{\text{MAP}}, \mathbf{X}_{1:T-1}^{\text{MAP}})$.

The inference of exogenous variables is difficult as the joint probability $p(\mathbf{x}_{1:t}, \mathbf{z}_{1:t}|g)$ cannot be factorized in the H-GPDM (or the GPDM), as shown in Figure 2.2a. Most previous inference methods on GPDMs (Ko and Fox, 2011; Deisenroth and Ohlsson, 2011; Deisenroth and Mohamed, 2012; Wang et al., 2012b, 2013) adopt the Fully Independent Conditional (FIC) approximation (Quiñonero-Candela and Rasmussen, 2005; Snelson, 2007). Our VB method also uses the FIC approximation to factorize the joint distribution $p(\mathbf{x}_{1:t}, \mathbf{z}_{1:t}|g)$ in the H-GPDM. As shown in Figure 2.2b, the FIC approximation decomposes the test conditional given by

$$p(\mathbf{x}_{1:t}|g, \mathcal{M}) \approx p(\mathbf{x}_1|g, \mathcal{M}) \prod_{\tau=2}^{t} p(\mathbf{x}_\tau|\mathbf{x}_{\tau-1}, g, \mathcal{M}), \tag{8}$$

and

$$p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t}, \mathcal{M}) \approx \prod_{\tau=1}^{t} p(\mathbf{z}_\tau|\mathbf{x}_\tau, \mathcal{M}), \tag{9}$$

where the latent functions $\mathbf{f}$ and $\mathbf{h}$ are integrated out, i.e.,

$$p(\mathbf{x}_\tau | \mathbf{x}_{\tau-1}, g, \mathcal{M}) = \int p(\mathbf{x}_\tau | \mathbf{x}_{\tau-1}, g, \mathbf{f}) p(\mathbf{f} | \mathcal{M}) d\mathbf{f},$$

and

$$p(\mathbf{z}_\tau | \mathbf{x}_\tau, \mathcal{M}) = \int p(\mathbf{z}_\tau | \mathbf{x}_\tau, \mathbf{h}) p(\mathbf{h} | \mathcal{M}) d\mathbf{f},$$

which cannot be factorized otherwise. We omit the model $\mathcal{M}$ hereafter.

In the remainder of this subsection, we first discuss VB inference algorithm for discrete exogenous variables in Section 2.3.1 and for predicting subsequent poses in Section 2.3.2. We later generalize the algorithm for continuous exogenous variables in Section 2.3.3.

2.3.1 Inference of Discrete Exogenous Variables

We first detail the VB inference method for discrete exogenous variables g that are time-invariant. For example, the exogenous variables of episodic movements can often be assumed time invariant, or they can be considered time invariant during a short period of time. The corresponding graphical model is shown in Figure 2.2b. Given a new series of observations $\mathbf{z}_{1:t}$, inferring the exogenous variables $p(g|\mathbf{z}_{1:t}) \propto p(g)p(\mathbf{z}_{1:t}|g)$ involves finding the marginal likelihood

$$p(\mathbf{z}_{1:t} | g) = \int p(\mathbf{x}_{1:t}, \mathbf{z}_{1:t} | g) d\mathbf{x}_{1:t},$$

where the latent state $\mathbf{x}_{1:t}$ are integrated out. According to the Jensen's inequality, the log marginal likelihood $\log p(\mathbf{z}_{1:t}|g)$ is lower-bounded by the negative free energy

$$F(q, g) \triangleq \mathbb{E}_q \left[\log p(\mathbf{x}_{1:t}, \mathbf{z}_{1:t} | g)\right] + \mathcal{H}(q(\mathbf{x}_{1:t}))$$

where $q(\mathbf{x}_{1:t})$ is a probability distribution on the latent states and $\mathcal{H}(q)$ is the entropy of $q(\mathbf{x}_{1:t})$. Since $\log p(\mathbf{z}_{1:t}|g) = F(q, g) - D_{\mathrm{KL}}\left(q || p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t}, g)\right)$ holds for arbitrary probability distributions q (Bishop, 2006), the nonnegative KL divergence $D_{\mathrm{KL}}\left(q || p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t}, g)\right)$ determines the tightness of the bound. Therefore, we can approximate the log marginal likelihood by a maximized negative free energy with respect to the variational distribution q given by

$$\log p(\mathbf{z}_{1:t} | g) \approx \max_q F(q, g). \tag{10}$$

This optimization is equivalent to minimizing the KL divergence between the variational distribution $q(\mathbf{x}_{1:t})$ and the posterior distribution of latent states $p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t}, g)$, i.e., we approximate the posterior distribution $p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t}, g)$ by $q(\mathbf{x}_{1:t})$.

To obtain a tractable solution, we use the mean-field approximation (Wainwright and Jordan, 2008) and consider the factorized variational distribution

$$q(\mathbf{x}_{1:t}; \Phi) = \prod_{\tau=1}^{t} q(\mathbf{x}_\tau; \Phi), \qquad q(\mathbf{x}_\tau; \Phi) \sim \mathcal{N}(\boldsymbol{\mu}_\tau, \boldsymbol{\Sigma}_\tau), \tag{11}$$

for $\tau = 1, \ldots, t$, where the distribution q is determined by the parameters $\Phi = \{\boldsymbol{\mu}_{1:t}, \boldsymbol{\Sigma}_{1:t}\}$. With the constraint that the variational distribution q can be factorized into independent Gaussian distributions, we obtain an approximate marginal likelihood based on Eq. (10), given by

$$\log p(\mathbf{z}_{1:t}|g) \approx \max_{\Phi} F(q, g; \Phi) = \max_{\Phi} \mathbb{E}_q[\log p(\mathbf{x}_{1:t}, \mathbf{z}_{1:t}|g)] + \mathscr{H}(q), \tag{12}$$

where the entropy $\mathscr{H}(q)$ and its gradients with respect to the parameters Φ have closed-form expressions. We can decompose the negative free energy $F(q, g; \Phi)$ according to the FIC approximation in Eq. (8) and (9), given by

$$F(q,g;\Phi)=\sum_{\tau=1}^{t} \mathbb{E}_q[\log p(\mathbf{z}_\tau|\mathbf{x}_\tau)]+\sum_{\tau=1}^{t-1} \mathbb{E}_q[\log p(\mathbf{x}_{\tau+1}|\mathbf{x}_\tau, g)]-\mathscr{H}(q(\mathbf{x}_1), p(\mathbf{x}_1))+\mathscr{H}(q)\,. \tag{13}$$

We use $\mathscr{H}(q, p)$ to denote the cross-entropy of q and p. For Gaussian distributions $q(\mathbf{x}_1) = \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $p(\mathbf{x}_1) = \mathcal{N}(\mathbf{m}_1, \mathbf{S}_1)$, the cross entropy and its gradients with respect to the variational parameters have closed-form expressions. We focus on the expected log likelihoods for the transition $\mathbb{E}_q[\log p(\mathbf{x}_{\tau+1}|\mathbf{x}_\tau, g)]$ and measurement mapping $\mathbb{E}_q[\log p(\mathbf{z}_\tau|\mathbf{x}_\tau)]$ in the following.

Expected log likelihood for measurement mapping. Consider the expected log likelihood $\mathbb{E}_q[\log p(\mathbf{z}_\tau|\mathbf{x}_\tau)]$ at time index τ in Eq. (13). The probability distribution $p(\mathbf{z}_\tau|\mathbf{x}_\tau)$ is the GP prediction given by a Gaussian distribution with mean $\mathbf{m}_z(\mathbf{x}_\tau)$ and covariance $\boldsymbol{\Sigma}_z(\mathbf{x}_\tau) = \sigma_z^2(\mathbf{x}_\tau)\mathbf{I}$. The expected log likelihood is thus given by

$$\begin{aligned}\mathbb{E}_q\left[\log p(\mathbf{z}_\tau|\mathbf{x}_\tau)\right] =& -\tfrac{1}{2}\mathbb{E}_q[\log|\boldsymbol{\Sigma}_z(\mathbf{x}_\tau)|] - \tfrac{D_z}{2}\log(2\pi)\\ &-\tfrac{1}{2}\mathbb{E}_q\left[(\mathbf{z}_\tau - \mathbf{m}_z(\mathbf{x}_\tau))^T\boldsymbol{\Sigma}_z^{-1}(\mathbf{x}_\tau)(\mathbf{z}_\tau - \mathbf{m}_z(\mathbf{x}_t))\right],\end{aligned} \tag{14}$$

which does not have a closed-form expression. This intractability is due to the GP predictive covariance $\boldsymbol{\Sigma}_z(\mathbf{x}_\tau)$, given by Eq. (7), which is a function of the unknown state as it encodes the model uncertainty.

Note that in Eq. (14), the expected log likelihood is averaged over a Gaussian distribution q, which reflects the log likelihood at a local region. In this chapter, we propose to approximate the model uncertainty $\mathbf{x}_\tau$ locally ($\mathbf{x}_\tau \sim \mathcal{N}(\boldsymbol{\mu}_\tau, \boldsymbol{\Sigma}_\tau)$) by the model uncertainty at the mean $\boldsymbol{\mu}_\tau$, leading to the predictive variance

$$\sigma_z^2(\mathbf{x}_\tau) \approx \sigma_z^2(\boldsymbol{\mu}_\tau) = k_z(\boldsymbol{\mu}_\tau, \boldsymbol{\mu}_\tau) - \mathbf{k}_z(\boldsymbol{\mu}_\tau)^T\mathbf{K}^{-1}\mathbf{k}_z(\boldsymbol{\mu}_\tau), \tag{15}$$

which is independent of $\mathbf{x}_\tau$ but still takes into account the local model uncertainty of the measurement GP, evaluated at the mean $\boldsymbol{\mu}_\tau$ of the variational distribution given in Eq. (11). This approximation assumes a constant value of model uncertainty around $\boldsymbol{\mu}_\tau$. To obtain tractable solutions, local approximations of the model uncertainty are frequently used. For example, the approximation we propose is also used for approximate inference in GPs based on linearization of the posterior GP mean function (Ko and Fox, 2009).

With the approximation in Eq. (15), we obtain an expected log likelihood for the measurement mapping, given by

$$\mathbb{E}_q\left[\log p(\mathbf{z}_\tau|\mathbf{x}_\tau)\right] \approx -\tfrac{D_z}{2}\log\sigma_z^2(\boldsymbol{\mu}_\tau) - \tfrac{D_z}{2}\log(2\pi) - \tfrac{1}{2\sigma_z^2(\boldsymbol{\mu}_\tau)}\mathbb{E}_q\left[||\mathbf{z}_\tau - \mathbf{m}_z(\mathbf{x}_\tau)||^2\right]. \tag{16}$$

Algorithm 2.2: The VB inference algorithm that finds the most likely exogenous variables g according to estimated marginal likelihoods. The algorithm also provides an approximate posterior distribution of the latent states $\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, which can be used for pose prediction.

```
Input  : Observations: $\mathbf{z}_{1:t}$.
Output: Most likely exogenous variable $g^{\mathrm{ML}}$.
Output: Variational parameters $\Phi$.
Initialize the variational parameters $\Phi_0 = \{\boldsymbol{\mu}_{1:t}, \boldsymbol{\Sigma}_{1:t}\}$ ;
for each $k \in \{1, \ldots, K\}$ do
    $\Phi_k \leftarrow \Phi_0$ ;
    Maximize $F(q, g = k; \Phi_k)$ in Eq. (12) w.r.t. $\Phi_k$ ;
    Log (unnormalized) marginal likelihood $\mathscr{L}(k) \leftarrow F(q, g = k; \Phi_k)$ ;
$g^{\mathrm{ML}} \leftarrow \mathrm{argmax}_{k \in \{1,\ldots,K\}} \mathscr{L}(k)$ ;
$\Phi \leftarrow \Phi_{g^{\mathrm{ML}}}$ ;
```

The expected log likelihood and its gradient have closed-form expressions for the considered linear and Gaussian covariance functions, and, thus, can be computed efficiently. We provide the expressions in Appendix 2.A. One can also obtain an approximation to $\mathbb{E}_q\left[\log p(\mathbf{z}_\tau|\mathbf{x}_\tau)\right]$ using Taylor expansion. However, computing the second order derivatives term is too expensive in practice.

Expected log likelihood for transition. The same approximation is applied to determine the expected log likelihood term in Eq. (13), given by

$$\sigma_x^2(\mathbf{x}_\tau, g) \approx \sigma_x^2(\boldsymbol{\mu}_\tau, g) = k_x([\boldsymbol{\mu}_\tau, g], [\boldsymbol{\mu}_\tau, g]) - \mathbf{k}_x(\boldsymbol{\mu}_\tau, g)^T \mathbf{K}_{1:T-1}^{-1} \mathbf{k}_x(\boldsymbol{\mu}_\tau, g).$$

This approximation results in the expected log likelihood for the transition

$$\mathbb{E}_q\left[\log p(\mathbf{x}_{\tau+1}|\mathbf{x}_\tau, g)\right] \approx -\frac{D_x}{2}\log\sigma_x^2(\boldsymbol{\mu}_t, g) - \frac{1}{2\sigma_x^2(\boldsymbol{\mu}_\tau, g)}\mathbb{E}_q\left[\|\mathbf{x}_{\tau+1} - \mathbf{m}_x(\mathbf{x}_\tau, g)\|^2\right] - \frac{D_x}{2}\log(2\pi), \tag{17}$$

which also has a closed-form solution for the adopted covariance function.

Based on the expected log likelihood terms in Eq. (16) and (17), we can compute the value of the negative free energy $F(q, g; \Phi)$ in Eq. (13) and its gradients with respect to the parameters Φ. Therefore, we obtain an approximate marginal likelihood in Eq. (12) using gradient-based optimization, such as the scaled conjugate gradient method (Møller, 1993). This marginal likelihood allows recognizing the exogenous variable g, such as the corresponding action or style or an observed movement, as described in Algorithm 2.2.

2.3.2 Pose Prediction

Tracking of human poses often relies on a prior model that predicts a subsequent pose $\mathbf{z}_t$ given a sequence of prior observations $\mathbf{z}_{1:t-1}$. With a minor modification to Eq. (13), i.e., removing the expected log likelihood $\mathbb{E}_q[\log p(\mathbf{z}_t|\mathbf{x}_t)]$ for missing measurements $\mathbf{z}_t$, the proposed inference method can recognize the exogenous variable g, which could be the gait styles, and also obtain the posterior distribution of state $\tau \in \{1, \ldots, t\}, \mathbf{x}_\tau \sim \mathcal{N}(\boldsymbol{\mu}_\tau, \boldsymbol{\Sigma}_\tau)$.

As shown in Figure 2.2b, the approximate posterior distribution $q(\mathbf{z}_t) \approx p(\mathbf{z}_t|\mathbf{z}_{1:t-1})$ can be estimated given the obtained belief on the corresponding state $\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, given by

$$q(\mathbf{z}_t) = \int p(\mathbf{z}_t|\mathbf{x}_t)\mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)d\mathbf{x}_t.$$

While this integral is intractable as $p(\mathbf{z}_t|\mathbf{x}_t)$ is determined by the nonlinear GP measurement function $\mathbf{h}$, it can still be approximated well by a Gaussian distribution (Girard et al., 2002), where the mean and covariance of $\mathbf{z}_t$ can be computed analytically. As a result, the posterior distribution $p(\mathbf{z}_t|\mathbf{z}_{1:t-1}) \approx q(\mathbf{z}_t)$ is approximated by a Gaussian distribution using moment matching.

However, according to the FIC approximation, as shown in Figure 2.2b, we estimate the posterior distribution $q(\mathbf{z}_t)$ assuming that the corresponding state $\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ contains all the information to recover the pose $\mathbf{z}_t$. This way, the correlation between the subsequent pose $\mathbf{z}_t$ and the observed poses $\mathbf{z}_{1:t-1}$ in the test data is not accounted for. To address this issue, we estimate the joint posterior distribution of all poses $\mathbf{z}_{1:t}$ based on the obtained belief on states $\mathbf{x}_{1:t}$, given by

$$q(\mathbf{z}_{1:t}) = \int p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})\prod_{\tau=1}^{t}\mathcal{N}(\mathbf{x}_\tau|\boldsymbol{\mu}_\tau, \boldsymbol{\Sigma}_\tau)d\mathbf{x}_\tau,$$

where we compute the predictive cross-covariance between $\mathbf{z}_{\tau_1}$ and $\mathbf{z}_{\tau_2}$ for $\tau_1, \tau_2 \in \{1, \ldots, t\}$, based on the results in Appendix 2.B. The resulting joint distribution of the measurements $q(\mathbf{z}_{1:t})$ is a Gaussian distribution with a full covariance matrix that captures the correlation among the poses $\mathbf{z}_{1:t}$ in the test data. Hence, the posterior distribution of the subsequent measurement $p(\mathbf{z}_t|\mathbf{z}_{1:t-1}) \approx q(\mathbf{z}_t|\mathbf{z}_{1:t-1})$ can be obtained by Gaussian conditional distribution given the measurements $\mathbf{z}_{1:t-1}$. The same method also applies to the recovery of missing measurements $\mathbf{z}_\tau$ given $\mathbf{z}_{1:\tau-1}$ and $\mathbf{z}_{\tau+1:t}$, i.e., $p(\mathbf{z}_\tau|\mathbf{z}_{1:\tau-1}, \mathbf{z}_{\tau+1:t}) \approx q(\mathbf{z}_\tau|\mathbf{z}_{1:\tau-1}, \mathbf{z}_{\tau+1:t})$.

2.3.3 Inference of Continuous Exogenous Variables

We generalize the VB method for continuous exogenous variables g, such as the coordinate of a target to hit. Instead of finding a MAP estimate, we seek for the posterior distribution $q(g) \approx p(g|\mathbf{z}_{1:t})$ approximated by a Gaussian distribution. We consider the variational distribution

$$q(g, \mathbf{x}_{1:t}) = q(g)\prod_{\tau=1}^{t} q(\mathbf{x}_\tau), \qquad q(g) = \mathcal{N}(\mu_g, \sigma_g^2),\ q(\mathbf{x}_\tau) = \mathcal{N}(\boldsymbol{\mu}_\tau, \boldsymbol{\Sigma}_\tau),$$

which consists of a factorized distribution $q(\mathbf{x}_{1:t})$ on states and $q(g)$ for the exogenous variable g, with the parameters $\Phi = \{\boldsymbol{\mu}_{1:t}, \boldsymbol{\Sigma}_{1:t}, \mu_g, \sigma_g^2\}$. We approximate the posterior distribution $q(g, \mathbf{x}_{1:t}) \approx p(g, \mathbf{x}_{1:t}|\mathbf{z}_{1:t})$ by minimizing the KL divergence between them. This step is equivalent to finding parameters Φ that maximizes the negative free energy

$$F(q; \Phi) = \mathbb{E}_q[\log p(\mathbf{x}_{1:t}, \mathbf{z}_{1:t}, g)] + \mathcal{H}(q).$$

The obtained $q(g) = \mathcal{N}(\mu_g, \sigma_g^2)$ allows predicting the time-invariant exogenous variable g, which could be the coordinate of a target that directs the observed movement.

2.3.4 Summary of the Inference Methods

Algorithm 2.2 describes the proposed VB inference algorithm for discrete exogenous variables g. The variational distribution $q(\mathbf{x}_{1:t})$ provides a solution to forecasting subsequent poses or recovering missing measurements. In addition, the VB method approximates the posterior distribution of the latent state variables and can, therefore, also be used for inference in the GPDM.

The proposed VB inference algorithm is generalized for continuous exogenous variables. The variational distribution q provides an approximation to the joint posterior distribution $p(\mathbf{x}_{1:t}, g|\mathbf{z}_{1:t})$ of latent state and exogenous variables, including the mean μ_g and variance σ_g^2 of the exogenous variables g.

To summarize, we make use of the mean-field approximation to obtain a tractable variational free energy for the marginal likelihood of exogenous variables, which gives rise to the posterior distribution of both exogenous variables and latent states. The posterior of latent states allows for the prediction of subsequent poses and the recovery of missing observations.

2.4 Experiments

We evaluate the proposed inference method in three applications:

(1) Gait recognition and pose prediction using Motion Capture (MOCAP) data. We showed that H-GPDM can better interpret the observed movements that consist of multiple gait styles, which were considered as discrete exogenous variables. We first showed in the supervised setting that the H-GPDM led to improved prediction accuracy of the subsequent poses and, hence, can potentially enhance a pose tracking system (Urtasun et al., 2006). Then, we showed that the unsupervised algorithm can discover latent gait styles from the observed movements.

(2) Character recognition from handwriting trajectories and recovery of missing observations. In this proof-of-concept experiment, we demonstrated that H-GPDM can recognize characters from the dynamics of handwriting trajectories and, therefore, better complete the missing trajectories in comparison to GPDM.

(3) Target prediction from table tennis hitting movements. In this experiment, we considered the “target” (where a table tennis player intends to shoot the ball) as a continuous exogenous variable that drives the hitting movement. H-GPDM allowed to predict the target from the hitting movement, which is crucial for building an anticipatory robot table tennis player (Wang et al., 2013).

2.4.1 Motion Capture Data

We show that our proposed inference method can recognize the gait style in human movements and concurrently predict subsequent poses. This capability of gait recognition provides an improved prior model for pose prediction, which is important in tracking human poses (Urtasun et al., 2006) for instance. We use the CMU MOCAP database[1] and follow the preprocessing

[1] http://mocap.cs.cmu.edu/

routine used by Taylor et al. (2011). Following the signal-noise ratio suggested by Wang et al. (2008), we chose the transition model parameters $\bar{\alpha} = [0.9, 0.001, 0.1, 10^{-4}]$. We considered a three-dimensional latent state space in the following experiments.

Pose Prediction and Gait Recognition

As discussed previously, pose prediction and gait recognition are fully dependent, as the gait is an exogenous driving factor of the dynamics. Given a new movement $\mathbf{z}_{1:t}$, the VB method finds the approximate posterior distribution of the latent states $q(\mathbf{x}_{1:t})$, as well as an estimate of the marginal likelihood of such the exogenous driving factor $p(\mathbf{z}_{1:t}|g)$, specifically by solving the optimization problem in Eq. (12).

We first consider the *Walkers* data set that consists of walking MOCAP data from four subjects. These subjects would have different poses and walking gait dynamics. We collected MOCAP trials from several subjects and split the trials from each subjects into training data and test data. Specifically, we used Trials 1–3 from Subject 7, Trials 1–3 from Subject 8, Trials 15-16 from Subject 16, and Trials 1–3 from Subject 35 for training. The test data comprised Trials 6–10 from Subject 7, Trials 8–9 from Subject 8, Trial 22 from Subject 16, and Trials 4–10 from Subject 35. All trials were down-sampled by a factor of ten.

In the experiments, we progressively predicted the subsequent pose $\mathbf{z}_t$ based on the recent ν measurements $\mathbf{z}_{t-\nu:t-1}$ for every time index t. The corresponding gait g is assumed static in this sliding window of measurements. At each time index t, we first computed the marginal likelihood for every gait $p(\mathbf{z}_{t-\nu:t-1}|g)$, and chose the maximum-likelihood gait g. Subsequently, we used the approximate posterior distribution $\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$ of the corresponding gait to predict the pose $\mathbf{z}_t$. We compared the VB inference method to the MAP estimation method described as conditional GPDM by Wang et al. (2008). As the MAP estimation method cannot straightforwardly recognize the gait variable g, we considered the oracle H-GPDM model, where the ground-truth value of g is provided. We also compared to both VB and MAP inference methods in the GPDM model using the same sliding windows, where different gait dynamics are not distinguished.

In Table 2.4.1, we report the results of the VB and MAP methods in H-GPDM and GPDM, using sliding windows of size $\nu = 5$. Based on the estimated marginal likelihoods, the VB method correctly recognized the walker in 90% cases. Its capability of gait recognition allowed for choosing the appropriate dynamics model. As a result, pose prediction using H-GPDM performed better than the inference methods using GPDM, as the H-GPDM better describes the gait dynamics of every individual subject. In both the oracle H-GPDM and the GPDM, the VB inference method substantially outperformed the MAP method. The combination the VB inference and the H-GPDM model has a comparative advantage at modeling walking motion and predicting subsequent poses, leading to an improved prior model for tracking. Note that the inference with VB and H-GPDM is advantageous regardless of the size of sliding windows ν. By changing the window size to $\nu = 10$, we obtained similar results, as shown in Table 2.4.1.

We also consider the *Weird Walking* data set that consists of MOCAP data from a single subject walking in four weird styles, including (1) walking with arms out, (2) fast walking, (3) hopping on the left foot, and (4) slow walking. These walking styles led to different gait dynamics although they may have similar poses. The models were trained on Trials 1–5, 17–19, 23–25, and 45–47 from Subject 132. For testing, we used Trials 6–12, 20–22, 26–28, and 48–50 from the same subject. All the trials were down-sampled by a factor of ten.

In Table 2.4.1, we report the results of the VB and MAP methods in the H-GPDM and the GPDM. Based on the estimated marginal likelihoods, the VB method could correctly recognize the walker in 61% cases. Note that for the style *hopping on the left foot*, the training data was not sufficient for learning the complex dynamics of this walking style. In addition, the movements of these four walking styles contain similar poses. Hence, in the test Trials 132-26, 132-27, and 132-28, the gait was not correctly recognized. Nevertheless, incorrect gait recognition did not necessarily result in a lower performance in the pose prediction, as the VB algorithm chose the gait dynamics model that best fit the recent measurements. As a result, pose

	H-GPDM		oracle H-GPDM		GPDM	
test trials	VB	accuracy	VB	MAP	VB	MAP
07-06	**0.069**	100%	**0.069**	0.082	0.095	0.111
07-07	0.137	52%	**0.096**	0.097	0.111	0.113
07-08	0.142	65%	**0.098**	0.133	0.158	0.140
07-09	**0.076**	100%	**0.076**	0.094	0.124	0.136
07-10	0.157	55%	**0.108**	0.114	0.136	0.148
08-08	0.108	83%	**0.103**	0.105	0.116	0.169
08-09	0.114	63%	**0.113**	0.117	0.122	0.175
16-22	**0.047**	100%	**0.047**	0.140	0.073	0.093
35-04	**0.044**	100%	**0.044**	0.064	0.070	0.087
35-05	**0.044**	100%	**0.044**	0.062	0.068	0.139
35-06	**0.042**	100%	**0.042**	0.075	0.066	0.083
35-07	**0.039**	100%	**0.039**	0.060	0.067	0.090
35-08	**0.044**	100%	**0.044**	0.081	0.066	0.109
35-09	**0.046**	100%	**0.046**	0.071	0.067	0.123
35-10	**0.039**	100%	**0.039**	0.069	0.064	0.079
average	0.077	90%	0.067	0.091	0.093	0.120

Table 2.1: Root-mean-square (RMS) error of the subsequent pose prediction on the *Walkers* data set using sliding windows of size 5. The test trials show the subject ID and trail number in the CMU MOCAP database. In the second and third columns, we show the RMS error of pose prediction and accuracy of gait recognition using the VB inference in H-GPDM model. In the fourth and fifth columns, we show the results of the oracle H-GPDM, provided with the ground-truth values of the exogenous variables (walkers' identities). In the last two columns, we also report the results using the GPDM model. The VB inference based on H-GPDM is advantageous in most of the test trials.

	H-GPDM		oracle H-GPDM		GPDM	
test trials	VB	accuracy	VB	MAP	VB	MAP
average	0.080	91%	**0.072**	0.089	0.089	0.117

Table 2.2: RMS error of the subsequent pose prediction on the *Walkers* data set using sliding windows of size 10. We only show the averaged performance.

prediction using the H-GPDM could still achieve better performance compared to the inference methods using the GPDM, for example in the test Trial 132-11. In both the oracle H-GPDM and the GPDM, the VB inference method also substantially outperformed the MAP method. By changing the window size to $v = 10$, we obtained similar results, as shown in Table 2.4.1.

Recovery of Missing Data

The H-GPDM can recover missing measurements using the proposed VB method. Following the evaluation by Wang et al. (2008), we considered the *Four Walkers* data set. The models were trained on Trial 2 from Subject 35, Trial 4 from Subject 10, Trial 1 from Subject 12, and trial 5 from Subject 16. For testing, we used Trial 3 from Subject 35, Trials 2–3 from Subject 12, and Trial 21 from Subject 16. All trials were down-sampled by a factor of four in consistent with Wang et al. (2008). We took 50 frames from each test trial, removed 31 frames in the middle, and recovered the missing measurements using the proposed VB method. We conducted 12 experiments for each test trial with missing frames 5–35, 6–36, ..., and 16–46.

	H-GPDM		oracle H-GPDM		GPDM	
test trials	VB	accuracy	VB	MAP	VB	MAP
132-06	**0.062**	58%	0.120	0.083	0.103	0.125
132-07	**0.080**	89%	0.097	0.121	0.201	0.146
132-08	**0.056**	58%	0.060	0.070	0.086	0.106
132-09	**0.046**	51%	0.056	0.081	0.204	0.124
132-10	**0.049**	76%	**0.048**	0.070	0.094	0.097
132-11	0.049	0%	**0.040**	0.057	0.209	0.102
132-12	0.041	98%	**0.039**	0.062	0.129	0.088
132-20	**0.106**	100%	**0.106**	0.171	0.134	0.183
132-21	**0.090**	91%	0.094	0.182	0.115	0.181
132-22	0.123	85%	**0.109**	0.180	0.170	0.160
132-26	0.240	5%	0.252	0.303	**0.190**	0.238
132-27	0.326	4%	0.198	0.223	**0.178**	0.202
132-28	0.244	24%	0.223	0.238	**0.188**	0.264
132-48	**0.043**	100%	**0.043**	0.089	0.049	0.075
132-49	**0.040**	100%	**0.040**	0.121	0.053	0.106
132-50	**0.048**	100%	**0.048**	0.111	0.056	0.098
average	0.103	61%	0.098	0.135	0.135	0.143

Table 2.3: RMS error of the subsequent pose prediction on the *Weird Walking* data set using sliding windows of size 5.

	H-GPDM		oracle H-GPDM		GPDM	
test trials	VB	accuracy	VB	MAP	VB	MAP
average	**0.093**	63%	0.099	0.132	0.123	0.148

Table 2.4: RMS error of the subsequent pose prediction on the *Weird Walking* data set using sliding windows of size 10. We only show the averaged performance.

For all test trials, the walker was correctly recognized. As shown in Table 2.4.1, the VB inference using H-GPDM achieved the best accuracy of the recovered measurements.

Discovery of Latent Gaits

The proposed Algorithm 2.1 can discover the latent driving factors when the side information is absent. In the *Weird Walking* data set, four different types of gait dynamics were discovered, which exactly coincide with the four weird styles in the training data. In the *Walkers* data set, the Subjects 16 and 35 were assigned the same gait style according to the discovered exogenous variables, as they have similar gait dynamics. Hence, they share the same transition model in the learned H-GPDM, despite their different poses.

2.4.2 Handwriting Character Recognition

In the following experiment, we consider the character handwriting trajectories data set[2] used by Williams et al. (2006, 2008). The entire data set contains 2858 trajectories of 20 characters, which were originally collected for identifying motor primitives in biological movements (Williams et al., 2008).

In the following, we treat the character as an exogenous variable when learning generative models of the observed trajectories. We first show that such a model enables character recognition with high precision. Moreover, we demonstrate that the H-GPDM (Figure 2.1b) facilitates the recovery of missing observations, unlike the GPDM (Figure 2.1a) that does not distinguish the character in the transition model.

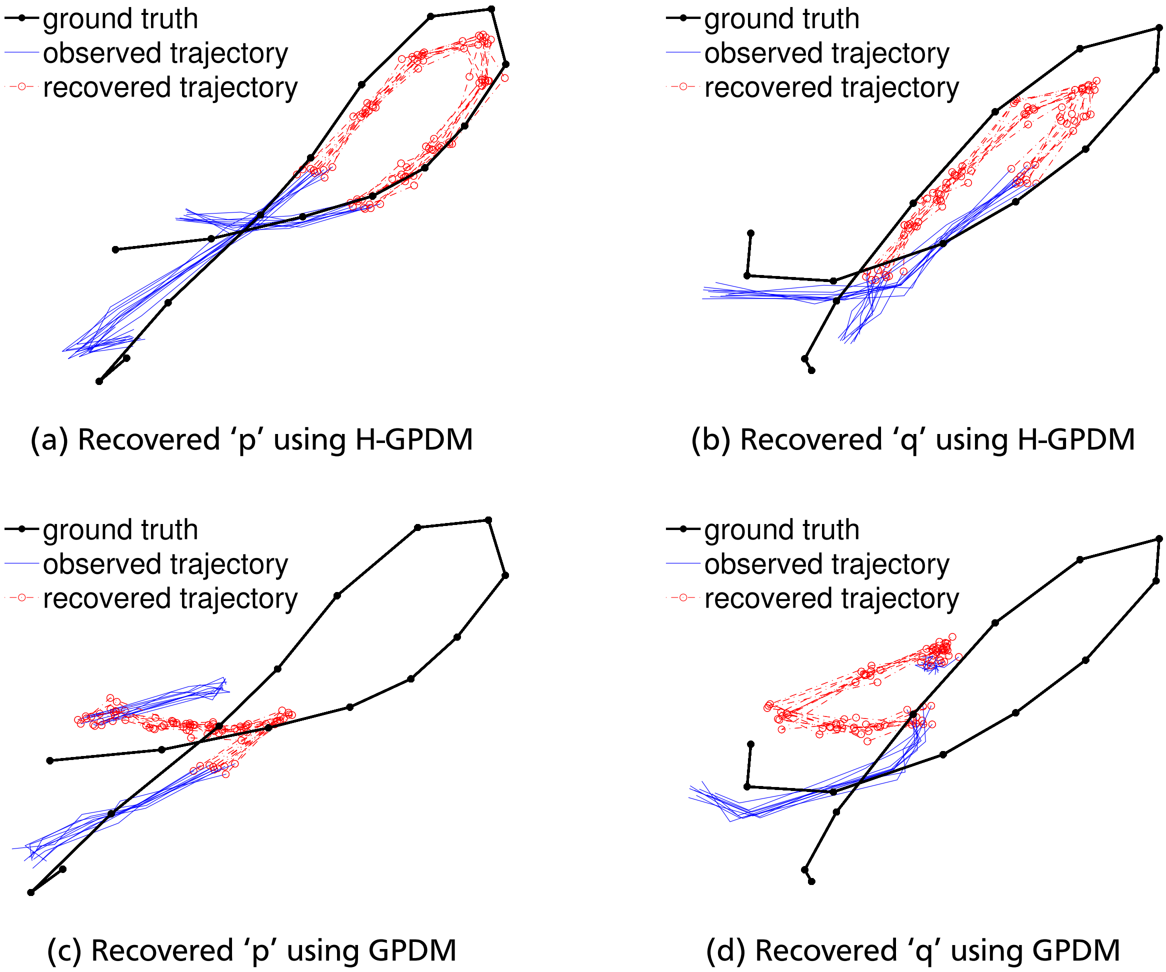
We used the following subsets of characters:

- *PQ*: We collect the character trajectories of characters ‘p’ and ‘q’. Although they have similar shapes as shown in Figure 2.3a and 2.3b, they have very different handwriting trajectories. The training data set contained 140 trajectories and the test data set contained 115 trajectories.
- *ABC*: We collect the character trajectories of characters ‘a’, ‘b’, and ‘c’. The training and test data sets contained 225 and 229 trajectories, respectively.

[2] http://archive.ics.uci.edu/ml/datasets/Character+Trajectories

	H-GPDM		GPDM	
test trials	VB	oracle MAP	VB	MAP
35-03	**0.012**	0.025	0.015	0.025
12-02	**0.015**	0.031	0.019	0.039
12-03	0.016	0.020	0.017	**0.014**
16-21	0.028	0.031	0.029	**0.020**
average	**0.018**	0.027	0.020	0.024

Table 2.5: RMS error of the missing poses recovery on the *Four Walkers* data set. The VB method based on H-GPDM achieved 100% accuracy in the recognition of walker.



(a) Recovered ‘p’ using H-GPDM

(b) Recovered ‘q’ using H-GPDM

(c) Recovered ‘p’ using GPDM

(d) Recovered ‘q’ using GPDM

Figure 2.3: Recovery of missing observations using the proposed VB method and using GPDM. We show ten trajectories sampled from the posterior distribution of latent states $p(\mathbf{x}_{1:t})$, where the red segments correspond to missing observations. The proposed VB inference in the H-GPDM simultaneously recognized the character and recovered the trajectories. The GPDM, which does not distinguish these two characters when learning the transition model, failed to recover the missing trajectories.

Each observation $\mathbf{z}$ had three features, the x and y-coordinates and the force of the pen tip. Since the observations are low-dimensional and informative for character recognition, we chose a three-dimensional latent state space and a linear covariance for the measurement model. The data was down-sampled by a factor of five.

Our proposed VB inference method had a precision of 100% on the PQ data. For comparison, the method based on motor primitives achieved a precision of 99% on the PQ data reported by Williams et al. (2008). On the ABC data, our VB inference method achieved a precision of 99.1%: The character ‘c’ was incorrectly recognized as ‘a’ twice, as to their share similar trajectories.

Our inference method can also recover missing observations. In our experiment, we used one trajectory of the character ‘p’ and one of ‘q’, and removed a substantial proportion of the handwriting trajectories. In particular, we extracted the observations with time indices 10–15

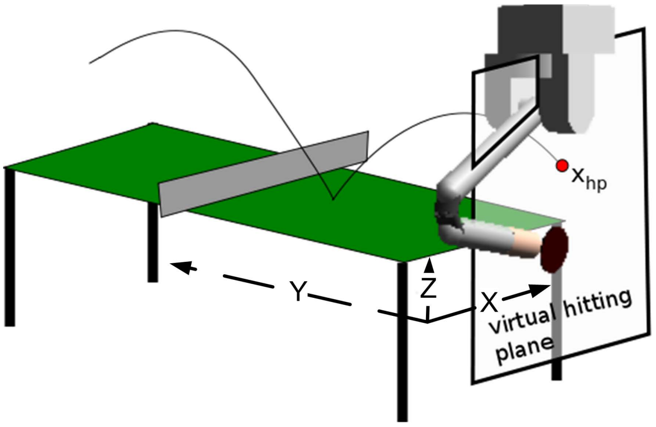


Figure 2.4: The *target*, represented by the exogenous variable g, is the intersection x_{hp} of the coming ball's trajectory and the virtual hitting plane 80 cm behind the table. Figure adapted from Mülling et al. (2011).

in the trajectories. The 'p' trajectory has an overall length of 22, and the 'q' trajectory has a length of 21. To achieve a good initialization of the variational parameters, we first conducted a forward sweep (filtering) on the belief of states $q(\mathbf{x}_\tau)$ for $\tau = 1, \ldots, t$, at each time step only optimizing $\boldsymbol{\mu}_\tau$, $\boldsymbol{\Sigma}_\tau$, and $\boldsymbol{\lambda}$ using the available observations in $\mathbf{z}_{1:\tau}$ according to Eq. (12), and keeping $\boldsymbol{\mu}_{1:\tau-1}$ and $\boldsymbol{\Sigma}_{1:\tau-1}$ fixed. When the observation $\mathbf{z}_\tau$ is missing, such an initialization exploits the learned transition model. We use the same initialization procedure for both our method and the inference with GPDM. The results are shown in Figure 2.3.

Our proposed H-GPDM inference method correctly recognized the characters from the incomplete trajectories and successfully recovered the missing parts, as shown in Figure 2.3a and 2.3b. However, GPDM, in which the transition model does not distinguish between the two characters as it does not model exogenous variables, failed to recover the missing trajectories, as shown in Figure 2.3c and 2.3d. An explanation is that the GPDM does not distinguish between the characters when learning the transition model. Hence, training data could have originated from any character, and the transition model can have multiple modes for the trajectories of 'p' and 'q'. The H-GPDM model with exogenous variables is more expressive than the GPDM, as it models the transition of each individual character. This additional flexibility is crucial for recovering trajectories from missing data.

2.4.3 Target Prediction for Table Tennis

We show that the H-GPDM can model goal-directed human movements and that the proposed VB method is applicable to goal inference. In this experiment, we consider human-robot table tennis and predict the intended *target* of the human player, i.e., where the player intends to shoot the ball, as shown in Figure 2.4, based on the observed racket's movement of the player. This prediction is important for the robot, which is mounted on the ceiling, to anticipate the human opponent's target and to prepare for hitting (Wang et al., 2013). We assume that the target is an important driving factor of the human player's racket movement (Wang et al., 2013) and, hence, treat the target as the exogenous variable in the H-GPDM of the player's racket. We also show that the unsupervised learning algorithm can discover latent styles in the hitting movements without side information, i.e., the target.

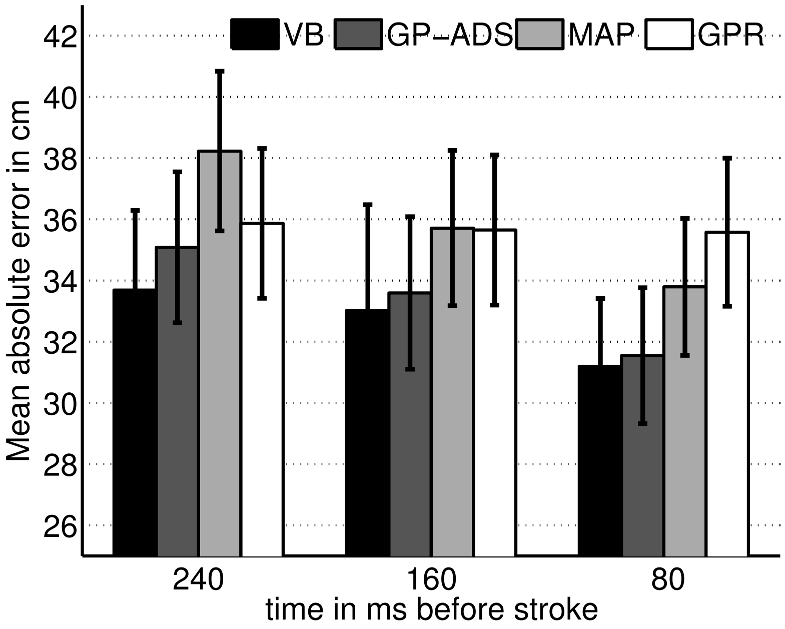
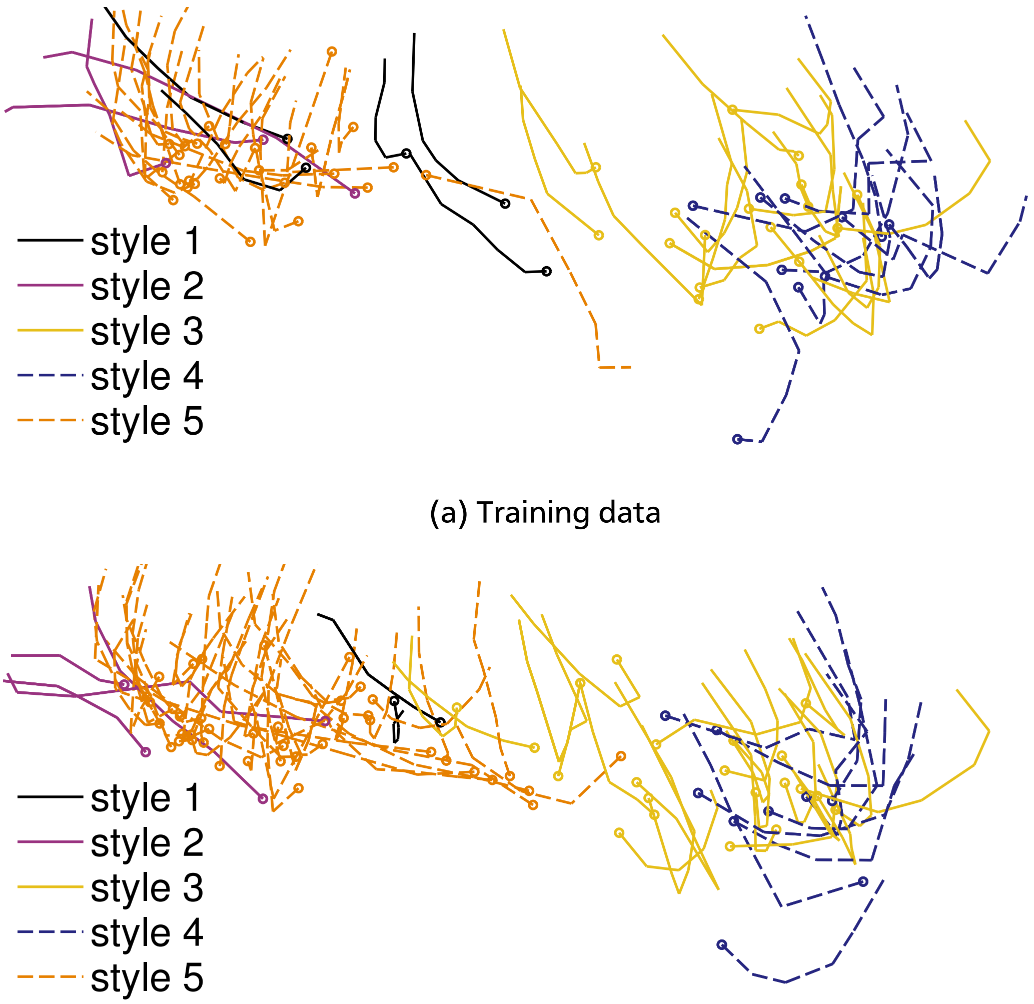


Figure 2.5: Mean absolute error of the ball's target with standard error of the mean. These errors are *before* the opponent hit the ball and only based on his movements. Target predictions are more accurate toward the end of the striking movements (from 240 ms to 80 ms before the opponent returns).

Target Prediction

The exogenous variable g considered here is the X-coordinate of the intended target, as shown in Figure 2.4, which we assume is a driving factor of the observed racket's movement $\mathbf{z}_{1:t}$. We consider the target g is time-invariant for each hitting movement. To evaluate the performance of target prediction, we use the data set with recorded striking movements from two human players (Wang et al., 2013). The true targets were obtained from a ball tracking system. The data set was divided into a training set with 100 hitting movements and a test set with 126 hitting movements, and the corresponding targets. We chose a four-dimensional latent state space, which achieved the maximum marginal likelihood estimated using the Bayesian GPLVM (Titsias and Lawrence, 2010). We compare our VB inference method to three other models: (1) *GPR*: Gaussian process regression for direct mapping from $\mathbf{z}_{t-1:t}$ to g using a sliding window of size 2, which was shown to be the optimal window size (Wang et al., 2013), (2) *GP-ADS*: goal inference based on GP Assumed Density Smoothing (Wang et al., 2012b), and (3) *MAP*: inference based on the MAP estimate of latent states, as used by Wang et al. (2008) for the inference for GPDM.

For every recorded hitting movement, we compared the proposed VB inference method with the GP-ADS method used by Wang et al. (2012b) and the GPR prediction (from $\mathbf{z}_{t-1:t}$ to $\mathbf{g}$) based on the observations up to 80 ms, 160 ms, and 240 ms *before* the human hit the ball. As demonstrated in Figure 2.5, the VB method outperformed the other methods. At 80 ms before the opponent hit the ball, the VB model resulted in a mean absolute error of 31.2 cm, which is a 12.3% improvement over the GPR with an average error 35.6 cm. The accuracy of GPR does not improve as more observations are obtained, as it learns a direct mapping from observations to the goal. In contrast, inference based on the H-GPDM becomes more reliable



(a) Training data

(b) Test data from a new opponent

Figure 2.6: Discovered latent styles in the table tennis striking movements. Figure (a) shows the striking movements in the training data. The trajectories shows the racket movement on the XY plane, starting from the position marked by a dot, moving towards the table/robot, which is above this figure. The color of each movement corresponds to its estimated style. We can observe that interpretable latent styles are obtained without annotation. For example, styles 1 and 5 correspond to backhand movements towards the left and right side, respectively; styles 3 and 4 correspond to forehand movements towards the left and right side, and movements in style 2 try to reach the ball faraway to the left. Figure (b) shows the movements from a new opponent. The color represents the *most likely* style obtained by the inference algorithm.

as more observations are obtained. The VB inference method also outperformed the other inference methods based on the H-GPDM: The inference based on GP-ADS (Wang et al., 2012b) achieved an average error of 31.5 cm. The inference based on the MAP estimate of latent states achieved an average error of 33.8 cm. We conclude that the inference based on H-GPDM should take into account the uncertainty in the latent state variables.

Discovery of Latent Styles

We also apply the unsupervised learning algorithm on the table tennis striking movements. Five different styles were discovered from the recorded racket trajectories of one particular opponent, as shown in Figure 2.6(a). The starting point of each trajectory indicates the location of the racket 400-ms before hitting the ball and the length of each trajectory is six (observations are sampled at 12.5Hz). One can see that these styles divide the trajectories into five groups reasonably; for interpretations, see the caption of Figure 2.6. Note that H-GPDM is not sensitive to the pre-defined number of styles, which is eight in this experiment. The learned styles are usually sparse when the pre-defined number of styles is sufficiently large.

The proposed inference algorithm successfully identified the styles in the test data of the strokes from a new opponent, as shown in Figure 2.6(b). We can observe that the new opponent has different preferences on the styles. These results showed that the H-GPDM has the potentially to adapt to new human subjects with difference preferences $p(g)$ over the latent styles while the dynamics remains unchanged.

2.5 Conclusions of Chapter 2

Gaussian Process Dynamics Models (GPDM) are a flexible class of latent-variable models that can represent complex nonlinear human movements, playing an increasingly important role in computer vision, robotics, and signal processing. However, GPDMs do not take into account the exogenous driving factors when the dynamics in the state space do not follow a simple Markov chain. In this chapter, we focus on the situation where the dynamics are driven by the exogenous variables, such as gait, goal, or movement styles. We incorporated the exogenous variables in the dynamics model to improve the interpretation, analysis, and prediction of human movements. The resulting Hierarchical GPDMs (H-GPDM) represent the generative process of human movements that are driven by exogenous variables.

The H-GPDM can be learned from observed movements in both supervised and unsupervised settings. We applied H-GPDM to jointly infer the exogenous variables and the missing observations. However, exact inference is analytically intractable. In this chapter, we have introduced a variational inference method for the H-GPDM, which is also applicable to the inference in GPDMs.

We have analyzed the performance of our proposed VB inference in three applications, i.e., target prediction from human-robot table tennis, character recognition and recovery from handwriting trajectories, and action recognition using motion capture data. The experimental results demonstrated the merit of both the H-GPDM and the inference algorithm.

The H-GPDM is applicable to intention inference when the human movements are driven by the underlying intention. In the next chapter, we consider a spacial case of the H-GPDM, namely Intention-Driven Dynamics Model (IDDM). As the VB inference method is time-demanding and cannot fulfill the real-time requirements in many robotic applications, we propose an online inference algorithm in the following chapter and apply it to two human-robot interaction scenarios.

2.A Expected Log Predictive Probability for GP

We consider the Gaussian process (GP) prior with the covariance function

$$k(\mathbf{x},\mathbf{x}') = k_l(\mathbf{x},\mathbf{x}') + k_g(\mathbf{x},\mathbf{x}') + \beta_4 \delta_{\mathbf{x},\mathbf{x}'}$$

which has a Gaussian function term $k_g(\mathbf{x},\mathbf{x}') = \beta_1 \exp\{-\frac{\beta_2}{2}||\mathbf{x}-\mathbf{x}'||^2\}$ and a linear term $k_l(\mathbf{x},\mathbf{x}') = \beta_3 \mathbf{x}^T\mathbf{x}'$. The GP predictive probability of a test output $\mathbf{z}$ given a test input $\mathbf{x}$ and training inputs $\mathbf{X}$ and outputs $\mathbf{Z}$ is

$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{m}_z(\mathbf{x}), \sigma_z^2(\mathbf{x})\mathbf{I}\right),$$

with

$$\begin{aligned}\mathbf{m}_z(\mathbf{x}) &= \mathbf{Z}^T\mathbf{K}_z^{-1}\mathbf{k}(\mathbf{x}),\\ \sigma_z(\mathbf{x}) &= k(\mathbf{x},\mathbf{x}) - \mathbf{k}^T(\mathbf{x})\mathbf{K}_z^{-1}\mathbf{k}(\mathbf{x}),\end{aligned}$$

where $\mathbf{K}_z \triangleq k(\mathbf{X},\mathbf{X})$ is the covariance matrix for the training data. We use the short-hand notation $\mathbf{k}(\mathbf{x}) \triangleq k(\mathbf{X},\mathbf{x})$.

We compute the expected log predictive probability $\mathbb{E}_q\left[\log p(\mathbf{z}|\mathbf{x})\right]$ with respect to $q(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$, which is essential for computing the expected log likelihoods for the transition and measurement function in Eq. (14) and (16), given by

$$\mathbb{E}_q\left[\log p(\mathbf{z}|\mathbf{x})\right] \approx -\frac{D}{2}\log\sigma_z^2(\boldsymbol{\mu}) - \frac{D}{2}\log(2\pi) - \frac{1}{2\sigma_z^2(\boldsymbol{\mu})}\mathbb{E}_q\left[||\mathbf{z}-\mathbf{m}_z(\mathbf{x})||^2\right],$$

where D is the dimension of the output $\mathbf{z}$. Here, we assume a constant value of model uncertainty around $\boldsymbol{\mu}$.

To compute $\mathbb{E}_q\left[\log p(\mathbf{z}|\mathbf{x})\right]$, one needs to compute

$$\begin{aligned}\mathbb{E}_q\left[||\mathbf{z}-\mathbf{m}_z(\mathbf{x})||^2\right] &= \mathbf{z}^T\mathbf{z} - 2\mathbf{z}^T\mathbf{V}_z\mathbb{E}_q\left[\mathbf{k}(\mathbf{x})\right] + \mathrm{tr}\left(\mathbf{V}_z^T\mathbf{V}_z\mathbb{E}_q\left[\mathbf{k}(\mathbf{x})\mathbf{k}(\mathbf{x})^T\right]\right)\\ &= \mathbf{z}^T\mathbf{z} - 2\mathbf{z}^T\mathbf{V}_z\underbrace{\mathbb{E}_q\left[\mathbf{k}(\mathbf{x})\right]}_{\text{Term (a)}} + \mathrm{tr}(\mathbf{V}_z^T\mathbf{V}_z\underbrace{\mathbb{E}_q\left[\mathbf{k}_g(\mathbf{x})\mathbf{k}_g(\mathbf{x})^T\right]}_{\text{Term (b)}})\\ &\quad + \underbrace{\mathbb{E}_q\left[\mathbf{k}_l(\mathbf{x})^T\mathbf{V}_z^T\mathbf{V}_z\mathbf{k}_l(\mathbf{x})\right]}_{\text{Term (c)}} + 2\underbrace{\mathrm{tr}\left(\mathbf{V}_z^T\mathbf{V}_z\mathbb{E}_q\left[\mathbf{k}_l(\mathbf{x})\mathbf{k}_g(\mathbf{x})^T\right]\right)}_{\text{Term (d)}},\end{aligned} \tag{18}$$

where we use $\mathbf{V}_z \triangleq \mathbf{Z}^T\mathbf{K}_z^{-1}$, $\mathbf{k}_l(\mathbf{x}) \triangleq k_l(\mathbf{X},\mathbf{x})$, and $\mathbf{k}_g(\mathbf{x}) \triangleq k_g(\mathbf{X},\mathbf{x})$ for notational simplicity.

For Term (a) in Eq. (18), we compute the vector $\mathbf{l} \triangleq \mathbb{E}_q\left[\mathbf{k}(\mathbf{x})\right]$. Its i-th element l_i is given by

$$\begin{aligned}l_i &= \int q(\mathbf{x})k(\mathbf{x},\mathbf{x}_i)d\mathbf{x} = \int q(\mathbf{x})k_l(\mathbf{x},\mathbf{x}_i)d\mathbf{x} + \int q(\mathbf{x})k_g(\mathbf{x},\mathbf{x}_i)d\mathbf{x}\\ &= \beta_3\boldsymbol{\mu}^T\mathbf{x}_i + \beta_1|\boldsymbol{\Sigma}\beta_2 + \mathbf{I}|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu})^T(\boldsymbol{\Sigma}+\beta_2^{-1}\mathbf{I})^{-1}(\mathbf{x}_i-\boldsymbol{\mu})\right),\end{aligned}$$

where $\mathbf{x}_i$ is the i-th training input.

For Term (b) in Eq. (18), we compute the matrix $\mathbf{Q} \triangleq \mathbb{E}_q\left[\mathbf{k}_g(\mathbf{x})\mathbf{k}_g(\mathbf{x})^T\right]$, where its elements are given by

$$Q_{ij} = \frac{k_g(\mathbf{x}_i, \boldsymbol{\mu})}{|2\beta_2\boldsymbol{\Sigma} + \mathbf{I}|^{\frac{1}{2}}} \exp\left(\beta_2(\tilde{\mathbf{x}}_{\mathbf{ij}} - \boldsymbol{\mu})^T(\boldsymbol{\Sigma} + \frac{1}{2}\beta_2^{-1}\mathbf{I})^{-1}\boldsymbol{\Sigma}(\tilde{\mathbf{x}}_{\mathbf{ij}} - \boldsymbol{\mu})\right)$$

with the short-hand notation $\tilde{\mathbf{x}}_{\mathbf{ij}} \triangleq \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$.

Term (c) in Eq. (18) is given by

$$\mathbb{E}_q\left[\mathbf{k}_l(\mathbf{x})^T\mathbf{V}_z^T\mathbf{V}_z\mathbf{k}_l(\mathbf{x})\right] = \beta_3^2\mathbb{E}_q\left[\mathbf{x}^T\mathbf{X}^T\mathbf{V}_z^T\mathbf{V}_z\mathbf{X}\mathbf{x}\right] = \beta_3^2\boldsymbol{\mu}^T\mathbf{S}\boldsymbol{\mu} + \beta_3^2\mathrm{tr}(\mathbf{S}\boldsymbol{\Sigma}),$$

with the short-hand notation $\mathbf{S} \triangleq \mathbf{X}^T\mathbf{V}_z^T\mathbf{V}_z\mathbf{X}$.

Term (d) in Eq. (18) is given by

$$\mathrm{tr}\left(\mathbf{V}_z^T\mathbf{V}_z\mathbb{E}_q\left[\mathbf{k}_l(\mathbf{x})\mathbf{k}_g(\mathbf{x})^T\right]\right) = \beta_3\mathrm{tr}\left(\mathbf{V}_z^T\mathbf{V}_z\mathbf{X}\mathbf{R}\right),$$

where we define the matrix $\mathbf{R} \triangleq \mathbb{E}_q\left[\mathbf{x}\mathbf{k}_g(\mathbf{x})^T\right]$. Its i-th column is given by

$$R_i = \beta_1\mathbb{E}_q\left[\mathbf{x}\exp(-\frac{\beta_2}{2}\|\mathbf{x} - \mathbf{x}_i\|^2)\right] = \beta_1 c_1 c_2^{-1}\boldsymbol{\psi}_i,$$

with

$$\begin{aligned}
c_1 &\triangleq \beta_1(2\pi)^{\frac{D}{2}}\beta_2^{-\frac{D}{2}},\\
c_2^{-1} &\triangleq (2\pi)^{-\frac{D}{2}}|\beta_2^{-1}\mathbf{I} + \boldsymbol{\Sigma}|^{-\frac{1}{2}}\exp\big(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu})^T(\beta_2^{-1}\mathbf{I} + \boldsymbol{\Sigma})^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\big),\\
\boldsymbol{\psi}_i &\triangleq (\beta_2\mathbf{I} + \boldsymbol{\Sigma}^{-1})^{-1}(\beta_2\mathbf{x}_i + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}).
\end{aligned}$$

For more details of the derivation, we refer the readers to Deisenroth (2010, Section 2.3). Combining Terms (a)–(d), we obtain the closed-form expression for the expected log predictive probability $\mathbb{E}_q[\log p(\mathbf{z}|\mathbf{x})]$. Therefore, we can analytically compute the expected log likelihoods for the measurement mapping $\mathbb{E}_q[\log p(\mathbf{z}_\tau|\mathbf{x}_\tau)]$ and for the transition $\mathbb{E}_q[\log p(\mathbf{x}_{\tau+1}|\mathbf{x}_\tau, g)]$ in the negative free energy $F(q, g; \Phi)$ in Eq. (13).

2.B GP Predictive Cross-covariance with Uncertain Inputs

Given two input variables $\mathbf{x}_1$ and $\mathbf{x}_2$,

$$\mathbf{x}_1 \sim \mathcal{N}(\boldsymbol{\mu}_{x_1}, \boldsymbol{\Sigma}_{x_1}), \mathbf{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}_{x_2}, \boldsymbol{\Sigma}_{x_2}),$$

each of which follows a Gaussian distribution.

For a GP with a zero mean function and covariance function $k(\cdot, \cdot)$, the predictive distribution is given by

$$p(y_1|x_1) \sim \mathcal{N}(m(\mathbf{x}_1), \sigma^2(\mathbf{x}_1)),$$

where the predictive mean and variance are

$$m(\mathbf{x}_1) = \mathbf{k}^T(\mathbf{X}, \mathbf{x}_1)\boldsymbol{\beta},$$

and

$$\sigma^2(\mathbf{x}_1) = k(\mathbf{x}_1, \mathbf{x}_1) - \mathbf{k}^T(\mathbf{X}, \mathbf{x}_1)\boldsymbol{\Sigma}_z^i \mathbf{k}(\mathbf{X}, \mathbf{x}_1).$$

Similarly, we can also obtain the cross-covariance,

$$\sigma(\mathbf{x}_1, \mathbf{x}_2) = k(\mathbf{x}_1, \mathbf{x}_2) - \mathbf{k}^T(\mathbf{X}, \mathbf{x}_1)\boldsymbol{\Sigma}_z^i \mathbf{k}(\mathbf{X}, \mathbf{x}_2).$$

The cross-covariance of y_1 and y_2 is given by

$$\text{cov}_{x_1,x_2}(y_1, y_2) = \mathbb{E}_{x_1,x_2}\left[\sigma(\mathbf{x}_1, \mathbf{x}_2)\right] + \text{cov}_{x_1,x_2}\left(m(\mathbf{x}_1), m(\mathbf{x}_2)\right),$$

where

$$\mathbb{E}_{x_1,x_2}\left[\sigma(\mathbf{x}_1, \mathbf{x}_2)\right] = \underbrace{\mathbb{E}_{x_1,x_2}\left[k(\mathbf{x}_1, \mathbf{x}_2)\right]}_{\triangleq c_{12}} - \underbrace{\mathbb{E}_{x_1}[\mathbf{k}^T(\mathbf{X}, \mathbf{x}_1)]}_{\triangleq \mathbf{l}_1^T} \boldsymbol{\Sigma}_z^i \underbrace{\mathbb{E}_{x_2}[\mathbf{k}(\mathbf{X}, \mathbf{x}_2)]}_{\triangleq \mathbf{l}_2},$$

and

$$\text{cov}_{x_1,x_2}\left(m(\mathbf{x}_1), m(\mathbf{x}_2)\right) = \mathbb{E}_{x_1,x_2}[m(\mathbf{x}_1)m(\mathbf{x}_2)] - \mathbb{E}_{x_1}[m(\mathbf{x}_1)]\mathbb{E}_{x_2}[m(\mathbf{x}_2)] = 0.$$

We obtain

$$\text{cov}(y_1, y_2) = c_{12} - \mathbf{l}_1^T \boldsymbol{\Sigma}_z^i \mathbf{l}_2,$$

where

$$\begin{aligned} c_{12} &= \iint \mathcal{N}(\mathbf{x}_1|\boldsymbol{\mu}_{x_1}, \boldsymbol{\Sigma}_{x_1})\mathcal{N}(\mathbf{x}_2|\boldsymbol{\mu}_{x_2}, \boldsymbol{\Sigma}_{x_2})k(\mathbf{x}_1, \mathbf{x}_2)d\mathbf{x}_1 d\mathbf{x}_2 \\ &= \beta_1 \left|\mathbf{I} + \beta_2(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T(\beta_2^{-1}\mathbf{I} + \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right), \end{aligned}$$

and

$$\begin{aligned} l_1(i) &= \int q(\mathbf{x}_1)k_g(\mathbf{x}_1, \mathbf{x}_i)d\mathbf{x}_1 \\ &= \beta_1|\boldsymbol{\Sigma}_1\beta_2 + \mathbf{I}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_1)^T(\boldsymbol{\Sigma}_1 + \beta_2^{-1}\mathbf{I})^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_1)\right). \end{aligned}$$

For more details of the derivation, we refer the reader to Girard (2004, Appendix C)

3 Online Intention Inference for Human-Robot Interaction

In this chapter, we apply the H-GPDM models for human-robot interaction. As the VB inference method introduced in the previous chapter is rather time-consuming, we propose an online intention inference algorithm that can fulfill the real-time requirements. We evaluate the online inference algorithm in two scenarios, i.e., target prediction in human-robot table tennis and action recognition for building an anticipatory humanoid robot. This chapter is based on the results presented by Wang et al. (2012b, 2013).

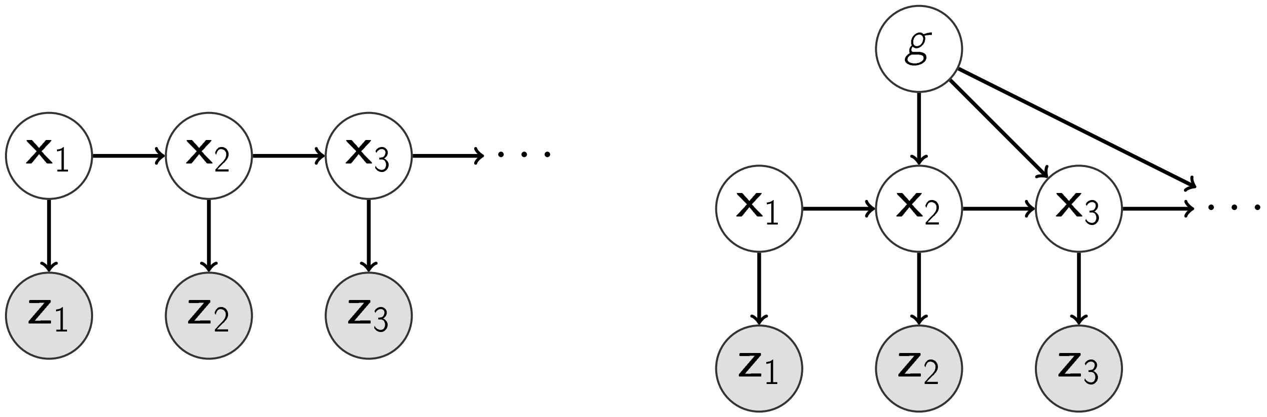
3.1 Prologue

Recent advances in sensors and algorithms allow for robots with improved perception abilities. For example, robots can now recognize human poses in real time using depth cameras (Shotton et al., 2013), which can potentially enhance the robot's ability to interact with humans. However, effective perception alone may not be sufficient for Human-Robot Interaction (HRI), since the robot's reactions ideally depend on the underlying *intention* of the human's action, including the others' goal, target, desire, and plan (Simon, 1982). Human beings rely heavily on the skill of intention inference (for example, in sports, games, and social interaction) and can improve the ability of intent prediction by training. For example, skilled tennis players are usually trained to possess substantially better anticipation than amateurs (Williams et al., 2002). This observation raises the question of how robots can learn to infer the human's underlying intention from movements.

In this chapter, we focus on intention inference from a movement based on modeling how the dynamics of a movement are governed by the intention. This idea is inspired by the hypothesis that a human movement usually follows a goal-directed policy (Baker et al., 2009; Friesen and Rao, 2011). The resulting dynamics model allows to estimate the probability distribution over intentions from observations using Bayes' theorem and to update the belief as additional observation is obtained. The human movement considered here is represented by a time series of observations, which makes discrete-time dynamics models a straightforward choice for movement modeling and intention inference. In a robotics scenario, we often rely on noisy and high-dimensional sensor data. However, the intrinsic states are typically not observable, and may have lower dimensions. Therefore, we seek a latent state representation of the relevant information in the data, and then model how the intention governs the dynamics in this latent state space, as shown in Figure 3.1b. The resulting model jointly learns both the latent state representation and the dynamics in the state space.

Designing a parametric dynamics model is difficult due to the complexity of human movement, e.g., its unknown nonlinear and stochastic nature. To address this issue, Gaussian processes (GPs), see (Rasmussen and Williams, 2006), have been successfully applied to modeling human dynamics. For example, the Gaussian Process Dynamics Model (GPDM) proposed by Wang et al. (2008) uses GPs for modeling the generative process of human motion with a nonlinear dynamical system, as shown in Figure 3.1a. Since the GP is a probabilistic nonparametric model, the unknown structure of the human moment can be inferred from data, while maintaining posterior uncertainty about the learned model itself.

As an extension to the GPDM, we propose the Intention-Driven Dynamics Model (IDDM), which models the generative process of *intention-driven movements*. The dynamics in the latent



(a) GPDM

(b) IDDM

Figure 3.1: Graphical models of the Gaussian process dynamics model (GPDM) and the proposed intention-driven dynamics model (IDDM), where we denote the intention by g, state by $\mathbf{x}_t$, and observation by $\mathbf{z}_t$. The proposed model explicitly incorporates the intention as an input to the transition function. Note that we omit the latent functions $\mathbf{f}$ and $\mathbf{h}$ in the graph for simplicity, as we adopt the Fully Independent Conditional (FIC) approximation, see Section 2.3. The full graphical model was shown in Figure 2.1b.

states are driven by the intention of the human action/behavior, as shown in Figure 3.1b. The IDDM can simultaneously find a good latent state representation of noisy and high-dimensional observations and describe the dynamics in the latent state space. The dynamics in latent state and the mapping from latent state to observations are described by GP models. Using the learned generative model, the human intention can be inferred from an ongoing movement using Bayesian inference. However, exact intention inference is not tractable due to the nonlinear and nonparametric GP transition model. Therefore, we propose an efficient approximate inference algorithm to infer the intention of a human partner.

The remainder of the chapter is organized as follows. First, in the remainder of this section, we illustrate the considered scenarios (Section 3.1.1) and discuss the related work (Section 3.1.2). Subsequently, we present the Intention- Driven Dynamics Model (IDDM) in Section 3.2. In Section 3.3, we study approximate algorithms for intention inference and extend them to online inference in Section 3.4. We evaluate the performance of the proposed methods in the two scenarios, i.e., target prediction in robot table tennis and action recognition, in Section 3.5 and 3.6. Finally, we summarize our contributions and discuss properties of the IDDM in Section 3.7.

3.1.1 Considered Scenarios

To verify the feasibility of the proposed methods, we discuss two representative scenarios where intention inference plays an important role in human-robot interactions:

(1) *Target inference in robot table tennis.* We consider human-robot table tennis games (Mülling et al., 2011), where the robot plays against a human opponent as shown in Figure 3.2. The robot's hardware constraints often impose strong limitations on its flexibility in such a high-speed scenario; for example, the Barrett WAM robot arm often cannot reach in-

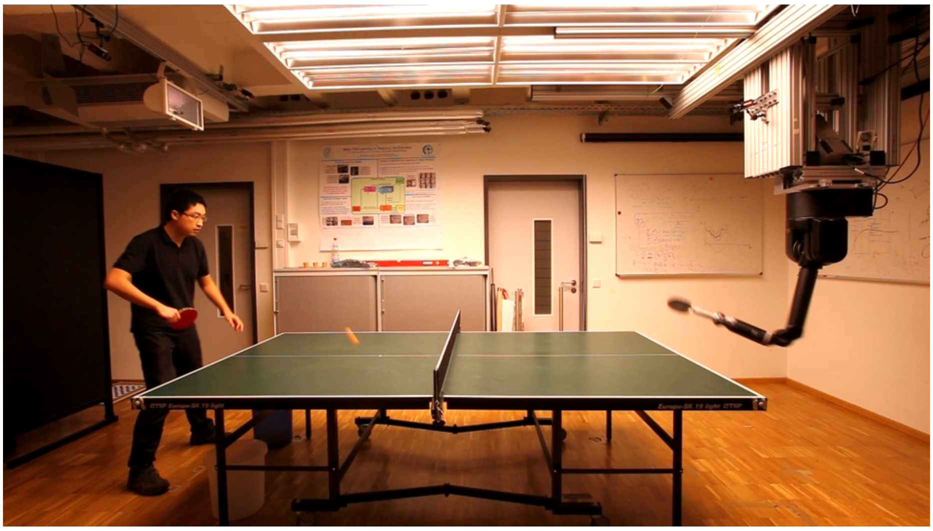


Figure 3.2: Target prediction in robot table tennis games: one example of HRI scenarios where intention inference plays an important role.

coming balls due to a lack of time caused by acceleration and torque limits for the biomimetic robot table tennis player presented by Mülling et al. (2011). The robot is kinematically capable of reaching a large hitting plane with pre-defined hitting movements such as forehand, middle, and backhand stroke movements that are capable in returning the ball shot into their corresponding hitting regions. However, movement initiation requires an early decision on the type of movement. In practice, it appears that to achieve the required velocity for returning the ball, this decision needs to be taken at least 80 ms *before* the opponent returns the ball (Wang et al., 2011b). Hence, it is necessary to choose the hitting movement before the opponent's racket has even touched the ball. This choice can be made based on inference of the target location where the opponent intends to return the ball from his incomplete stroke movement. We show that the IDDM can improve the prediction of the human player's intended target over a baseline method based on Gaussian process regression, and can thus expand the robot's hitting region substantially by using multiple hitting movements.

(2) *Action recognition for interactive humanoid robots.* In this setting, we use our IDDM to recognize the actions of the human, as shown in Figure 3.3, which can improve the interaction capabilities of a robot (Jenkins et al., 2007). In order to realize natural and compelling interactions, the robot needs to correctly recognize the actions of its human partner. In turn, this ability allows the robot to react in a proactive manner. We show that the IDDM has the potential to identify the action from movements in a simplified scenario.

In most robotics applications, including the scenarios discussed above, the decision making systems are subject to real-time constraints and need to deal with a stream of data. Moreover, the human's intention may vary over time. To address these issues, we propose an algorithm for online intention inference. The online algorithm can process the stream data and fulfill the real-time requirements. In the experiments, the proposed online intention inference algorithm achieved over four times acceleration over the previous method (Wang et al., 2012b).

Figure 3.3: Target prediction in robot table tennis games: a second example of HRI scenarios where intention inference is crucial.

3.1.2 Related Work

We review methods for intention inference and for modeling human movements that are related to the proposed IDDM and inference methods.

Intention Inference

Inference of intentions has been investigated in different settings. Most of previous work relies on probabilistic reasoning.

Intention inference with discrete states and actions has been extensively studied, using Hidden Markov Models (HMMs) to model and predict human behavior where different dynamics models were adopted to the corresponding behaviors (Pentland and Liu, 1999). Online learning of intentional motion patterns and prediction of intentions based on HMMs was proposed in (Vasquez et al., 2008), which allows efficient inference in real time. The HMM can be learned incrementally to cope with new motion patterns in parallel with prediction (Vasquez et al., 2009).

Probabilistic approaches to plan recognition in artificial intelligence (Liao et al., 2007) typically represent plans as policies in terms of state-action pairs. When the intention is to maximize an unknown utility function, inverse reinforcement learning (IRL) infers the underlying utility function from an agent's behavior (Abbeel and Ng, 2004). IRL has also been applied to model intention-driven behavior. For instance, maximum entropy IRL (Ziebart et al., 2008) has been used to model goal-directed trajectories of pedestrians (Ziebart et al., 2009) and target-driven pointing trajectories (Ziebart et al., 2012).

In cognitive science, Bayesian models were used for inferring goals from behavior in (Rao et al., 2004), where a policy conditional on the agent's goal is learned to represent the behavior. Bayesian models can be used to interpret the agent's behavior and predict its behavior in a similar environment with the learned model (Baker et al., 2006). In a recent work (Friesen and

Rao, 2011), a computational framework was proposed to model gaze following, where GPs are used to model the dynamics with actions driven by a goal. These methods assume that the states can be observed. However, in practice the states are often not well-defined or not observable for complex human movement.

One can also consider the intention inference jointly with decision making, such as autonomous driving (Bandyopadhyay et al., 2013), control (Hauser, 2012), or navigation in human crowds (Kuderer et al., 2012). For example, when the state space is finite, the problem can be formulated as a Partially Observable Markov Decision Process (Kurniawati et al., 2011) and solved efficiently (Wang et al., 2012a). In contrast, our method assumes that the robot's decision does not influence the intention of the human and considers intention inference and decision making separately, which allows us to efficiently deal with high-dimensional data stream and fulfill the real-time constraints.

Gaussian Process Dynamics Model and Extensions

Observations of human movements often consist of high-dimensional features. Determining a low-dimensional latent state space is an important issue for understanding observed actions. The Gaussian Process Latent Variable Model (GPLVM) (Lawrence, 2004) finds the most likely latent variables while marginalizing out the function mapping from latent to observed space. The resulting latent variable representation allows to model the dynamics in a low-dimensional space. For example, the Gaussian Process Dynamics Model (Wang et al., 2008) uses an additional GP transition model for the dynamics of human motion on the latent state space.

In robotics applications, the GPLVM can also be used for learning dynamical system motor primitives (Ijspeert et al., 2002) in a low-dimensional latent space, to achieve robust dynamics and fast learning (Bitzer and Vijayakumar, 2009). Nonparametric dynamics models are also applied for tracking a small robotic blimp with two cameras (Ko and Fox, 2009), where GP-Bayes filters were proposed for efficient filtering. In a follow-up work (Ko and Fox, 2011), the model is learned based on the GPLVM, so that the latent states need not be provided for learning.

The use of a GP transition model renders exact inference in the GPDM and, hence, in the IDDM, analytically intractable. Nevertheless, approximate inference methods have been successfully applied based on filtering and smoothing in nonlinear dynamical systems. For the GPDM and its extensions, approximate inference can be achieved using Particle Filters (GP-PF), Extended Kalman Filters (GP-EKF), and Unscented Kalman Filters (GP-UKF) as proposed by (Ko and Fox, 2009). GP Assumed Density Filters (GP-ADF) for efficient GP filtering, and general smoothing in GPDMs were proposed in (Deisenroth et al., 2009) and (Deisenroth et al., 2012), respectively. These filtering and smoothing techniques allow the use of Expectation-Maximization (EM) framework for approximate inference (Ghahramani and Roweis, 1999; Turner et al., 2010; Wang et al., 2012b). Int the previous chapter, we presented a Variational Bayes (VB) inference method. However, this VB method is rather time-consuming. To fulfill the real-time requirements in human-robot interaction, we present an online inference method in this chapter.

3.2 Intention-Driven Dynamics Model

We present the Intention-Driven Dynamics Model (IDDM), which is a special case of the proposed H-GPDM, and an extension to the GPDM (Wang et al., 2008). We briefly review the model description here for the self-containment of this chapter. We refer to Section 2.2.2 for the learning of the model.

3.2.1 Measurement and Transition Models

In the proposed IDDM, one set of GPs models the transition function in the latent space conditioned on the intention g. A second set of GPs models the measurement mapping from the latent states $\mathbf{x}$ and the observations $\mathbf{z}$. For notational simplicity, we assume the intention variable g is discrete or a scalar. The model and method can easily generalize to multi-variate intention variables. We detail both the measurement and transition models in the following.

Measurement model

The observations of a movement are a time series $\mathbf{z}_{1:T} \triangleq [\mathbf{z}_1, \ldots, \mathbf{z}_T]$, where $\mathbf{z}_t \in \mathbb{R}^{D_z}$. In the proposed generative model, we assume that an observation $\mathbf{z}_t \in \mathbb{R}^{D_z}$ is generated by a latent state variable $\mathbf{x}_t \in \mathbb{R}^{D_x}$ according to

$$\mathbf{z}_t = \mathbf{W}^{-1}\mathbf{h}(\mathbf{x}_t) + \mathbf{W}^{-1}\mathbf{n}_{z,t}\,, \quad \mathbf{n}_{z,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_z)\,,$$

where the diagonal matrix $\mathbf{W} = \mathrm{diag}(w_1, \ldots, w_{D_z})$ scales the outputs of $\mathbf{h}(\mathbf{x}_t)$. The scaling parameters $\mathbf{W}$ allow for dealing with raw features that are measured in different units, such as positions and velocities. We place a GP prior distribution on each dimension of the unknown function $\mathbf{h}$, which is marginalized out during learning and inference. The GP prior $\mathcal{GP}(m_z(\cdot), k_z(\cdot,\cdot))$ is fully specified by a mean function $m_z(\cdot)$ and a positive semidefinite covariance (kernel) function $k_z(\cdot,\cdot)$. Without specific prior knowledge on the latent state space, we use the same mean and covariance function for the GP prior on every dimension of the unknown measurement function $\mathbf{h}$, and use the noise (co)variance $\mathbf{S}_z = s_z^2\mathbf{I}$. The predictive probability of the observations $\mathbf{z}_t$ is given by a Gaussian distribution $\mathbf{z}_t \sim \mathcal{N}(\mathbf{m}_z(\mathbf{x}_t), \mathbf{\Sigma}_z(\mathbf{x}_t))$, where the predictive mean and covariance are computed based on training inputs $\mathbf{X}_z$ and outputs $\mathbf{Y}_z$, given by

$$\begin{aligned}
\mathbf{m}_z(\mathbf{x}_t) &= \mathbf{Y}_z\mathbf{K}_z^{-1}\mathbf{k}_z(\mathbf{x}_t),\\
\mathbf{\Sigma}_z(\mathbf{x}_t) &= \sigma_z^2(\mathbf{x}_t)\mathbf{I},\\
\sigma_z^2(\mathbf{x}_t) &= k_z(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{k}_z(\mathbf{x}_t)^T\mathbf{K}_z^{-1}\mathbf{k}_z(\mathbf{x}_t)\,,
\end{aligned}$$

where, we use the shorthand notation $\mathbf{k}_z(\mathbf{x}_t)$ to represent the cross-covariance vector between $\mathbf{h}(\mathbf{X}_z)$ and $\mathbf{h}(\mathbf{x}_t)$, and use $\mathbf{K}_z$ to represent the kernel matrix of $\mathbf{X}_z$.

Transition model

We consider first-order Markov transition model, see Figure 3.1b, with a latent transition function $\mathbf{f}$, such that

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, g) + \mathbf{n}_{x,t}, \quad \mathbf{n}_{x,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_x)\,.$$

The state $\mathbf{x}_{t+1}$ at time $t+1$ depends on the latent state $\mathbf{x}_t$ at time t as well as on the intention g. We place a GP prior $\mathcal{GP}(m_x(\cdot), k_x(\cdot,\cdot))$ on every dimension of $\mathbf{f}$ with shared mean and covariance functions. Subsequently, the predictive distribution of the latent state $\mathbf{x}_{t+1}$ conditioned on the current state $\mathbf{x}_t$ and intention g is a Gaussian distribution given by $\mathbf{x}_{t+1} \sim \mathcal{N}(\mathbf{m}_x([\mathbf{x}_t, g]), \mathbf{\Sigma}_x([\mathbf{x}_t, g]))$ based on training inputs $\mathbf{X}_x$ and outputs $\mathbf{Y}_x$, with

$$\begin{aligned}
\mathbf{m}_x([\mathbf{x}_t, g]) =& \mathbf{Y}_x \mathbf{K}_x^{-1} \mathbf{k}_x([\mathbf{x}_t, g]),\\
\mathbf{\Sigma}_x([\mathbf{x}_t, g]) =& \sigma_x^2([\mathbf{x}_t, g])\mathbf{I},\\
\sigma_x^2([\mathbf{x}_t, g]) =& k_x([\mathbf{x}_t, g], [\mathbf{x}_t, g]) - \mathbf{k}_x([\mathbf{x}_t, g])^T \mathbf{K}_x^{-1} \mathbf{k}_x([\mathbf{x}_t, g]),
\end{aligned}$$

where $\mathbf{K}_x$ is the kernel matrix of training data $\mathbf{X}_x = \big[[\mathbf{x}_1, g_1], \ldots, [\mathbf{x}_n, g_n]\big]$. The transition function $\mathbf{f}$ may also depend on environment inputs $\mathbf{u}$, e.g., controls or motor commands. We assume that environment inputs are observable and omit them in the description of model for notational simplicity.

3.2.2 Covariance Functions

By convention, we use GP prior mean functions that are zero everywhere for notational simplicity, i.e., $m_z(\cdot) \equiv 0$ and $m_x(\cdot) \equiv 0$. Hence, the model is determined by the covariance functions $k_z(\cdot,\cdot)$ and $k_x(\cdot,\cdot)$, which will be motivated in the following.

The underlying dynamics of human motion are usually nonlinear. To account for nonlinearities, we use a flexible Gaussian tensor-product covariance function for the dynamics, i.e.,

$$\begin{aligned}
k_x([\mathbf{x}_i, g_i], [\mathbf{x}_j, g_j]; \boldsymbol{\alpha}) &= k_x(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\alpha}) k_x(g_i, g_j; \boldsymbol{\alpha}) + k_{\text{noise}}\\
&= \alpha_1 \exp\left(-\tfrac{\alpha_2}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2 - \tfrac{\alpha_3}{2}(g_i - g_j)^2\right) + \alpha_4 \delta_{ij},
\end{aligned}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]$ is the set of all hyperparameters, and δ is the Kronecker delta function. When the intention g is a discrete variable, we set the hyperparameter $\boldsymbol{\alpha}_3 = \infty$ such that $k_x(g_i, g_j; \boldsymbol{\alpha}) \equiv \delta_{ij}$.

The covariance function for the measurement mapping from the state space to observation space is chosen depending on the task. For example, the GPDM in (Wang et al., 2008) uses an isotropic Gaussian covariance function

$$k_z(\mathbf{x}, \mathbf{x}'; \boldsymbol{\beta}) = \exp\left(-\tfrac{\beta_1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) + \beta_2 \delta_{\mathbf{x},\mathbf{x}'}, \tag{19}$$

parameterized by the hyperparameters $\boldsymbol{\beta}$, as, intuitively, the latent states that generate human poses lie on a nonlinear manifold. Note that the hyperparameters $\boldsymbol{\beta}$ do not contain the signal variance, which is parameterized by the scaling factors $\mathbf{W}$ in Eq. (3.2.1). In the context of target prediction in table tennis games, we use the linear kernel

$$k_z(\mathbf{x}, \mathbf{x}'; \boldsymbol{\beta}) = \mathbf{x}^T \mathbf{x}' + \beta_1 \delta_{\mathbf{x},\mathbf{x}'}, \tag{20}$$

as the observations are already low-dimensional, but subject to substantial noise.

3.3 Approximate Intention Inference

After learning the model $\mathscr{M}$ from the training data set $\mathscr{D}$, the intention g can be inferred from a sequence of new observations $\mathbf{z}_{1:T}$. For notational simplicity, we do not explicitly condition on the model $\mathscr{M}$ and the data set $\mathscr{D}$. The measurement model defined in Eq. (3.2.1) scales the observations by a diagonal matrix $\mathbf{W}$. Therefore, we pre-process every received observation with the scaling matrix $\mathbf{W}$ and omit $\mathbf{W}$ hereafter as well.

The IDDM models the generative process of movements, represented by observations $\mathbf{z}_{1:T}$, given an intention g. Using Bayes' rule, we estimate the posterior probability (belief) on an intention g from observations $\mathbf{z}_{1:T}$. The posterior is given by

$$p(g|\mathbf{z}_{1:T}) = \frac{p(\mathbf{z}_{1:T}|g)p(g)}{p(\mathbf{z}_{1:T})} \propto p(g) \int p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}|g) d\mathbf{x}_{1:T}, \tag{21}$$

where computing the marginal likelihood $p(\mathbf{z}_{1:T}|g)$ requires to integrate out the latent states $\mathbf{x}_{1:T}$. Exactly computing the posterior in Eq. (21) is not tractable due to the use of nonlinear GP transition model. Hence, we resort to approximate inference. In previous work (Wang et al., 2012b), we introduced an EM algorithm for finding the maximum likelihood estimate of the intention. However, this point estimate may not suffice for the reactive policies of the robot that also take into account the uncertainty in the intention inference (Wang et al., 2011a,b; Bandyopadhyay et al., 2013). For example, in the table tennis task, the robot may need to choose the optimal time to initiate its hitting movement, and such a choice is ideally made based on how certain the prediction of target is (Wang et al., 2011b). In this chapter, we extend our previous inference method (Wang et al., 2012b), such that the uncertainty about the intention is explicitly modeled and taken into account when making decisions.

The key challenge in estimating the belief in Eq. (21) is integrating out the latent states $\mathbf{x}_{1:T}$. A common approximation to the log marginal posterior is to compute a lower bound $\mathscr{B}(g) \leq \log p(g|\mathbf{z}_{1:T})$. The bound is given by

$$\begin{aligned} \mathscr{B}(g) &\triangleq \mathbb{E}_q\left[\log p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}, g)\right] + \mathscr{H}(q) \\ &= \log p(g|\mathbf{z}_{1:T}) - \mathrm{KL}\left(q||p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)\right) \\ &\leq \log p(g|\mathbf{z}_{1:T}), \end{aligned} \tag{22}$$

which holds for any distribution $q(\mathbf{x}_{1:T})$ on the latent states. Here, the Kullback-Leibler (KL) divergence $\mathrm{KL}\left(q||p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)\right)$ determines how well $\mathscr{B}(g)$ can approximate the belief. Based on this approximation, the inference problem consists of two steps, namely, (a) finding an approximation $q(\mathbf{x}_{1:T}) \approx p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)$, and (b) computing the approximate belief $\mathscr{B}(g)$. When using the EM algorithm for the maximum likelihood estimate of the intention g (Wang et al., 2012b), the E-step and M-step correspond to these two steps, respectively.

For step (a), we approximate the posterior of latent states $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)$ by a Gaussian distribution $q(\mathbf{x}_{1:T})$. For this purpose, we use the forward-backward smoothing method proposed in (Deisenroth et al., 2009, 2012), which is based on moment matching. Typically, Gaussian moment matching provides credible error bars, i.e., it is robust to incoherent estimates. The resulting approximate distribution q that we use in the lower bound $\mathscr{B}$ in Eq. (22) is given by

$$q(\mathbf{x}_{1:T}) = \mathscr{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) \approx p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g), \tag{23}$$

with the mean and block-tri-diagonal covariance matrix

$$\boldsymbol{\mu}_q = \begin{bmatrix} \boldsymbol{\mu}^x_{1|T} \\ \vdots \\ \boldsymbol{\mu}^x_{T|T} \end{bmatrix}, \quad \boldsymbol{\Sigma}_q = \begin{bmatrix} \boldsymbol{\Sigma}^x_{1|T} & \boldsymbol{\Sigma}^x_{1,2|T} & & \\ \boldsymbol{\Sigma}^x_{2,1|T} & \ddots & \ddots & \\ & \ddots & \ddots & \boldsymbol{\Sigma}^x_{T-1,T|T} \\ & & \boldsymbol{\Sigma}^x_{T,T-1|T} & \boldsymbol{\Sigma}^x_{T|T} \end{bmatrix}, \tag{24}$$

where we only need to consider the cross-covariance between consecutive states[3].

For step (b), based on the approximation $q(\mathbf{x}_{1:T})$, the posterior belief $p(g|\mathbf{z}_{1:T})$ can then be approximated by the lower bound $\mathscr{B}(g)$ in Eq. (22).

In the following, we first detail step (a), i.e., the computation of q for our IDDM, in Section 3.3.1. Subsequently, we discuss step (b), i.e., efficient belief estimation, in Section 3.3.2.

3.3.1 Filtering and Smoothing in the IDDM

To obtain the posterior distribution $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)$, approximate filtering and smoothing with GPs are crucial in our proposed IDDM. We place a Gaussian prior on the initial state $\mathbf{x}_1$. Subsequently, Gaussian approximations $q(\mathbf{x}_{t-1}, \mathbf{x}_t)$ of $p(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:T}, g)$ for $t = 2, \dots, T$ are computed. We explicitly determine the marginals $p(\mathbf{x}_t|\mathbf{z}_{1:T}, g)$ for $t = 1, \dots, T$, and the cross-covariance terms $\operatorname{cov}[\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:T}, g]$, $t = 2, \dots, T$. These steps yield a Gaussian approximation with a block-tri-diagonal covariance matrix, see Eq. (24). These computations are based on forward-backward smoothing (GP-RTSS) as proposed by Deisenroth et al. (2012).

As a first step, we compute the posterior distributions $p(\mathbf{x}_t|\mathbf{z}_{1:T}, g)$ with $t = 1, \dots, T$. To compute these posteriors using Bayesian forward-backward smoothing in the IDDM, it suffices to compute both joint distributions $p(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:t-1}, g)$ and $p(\mathbf{x}_t, \mathbf{z}_t|\mathbf{z}_{1:t-1}, g)$. The Gaussian filtering and smoothing updates can be expressed solely in terms of means and (cross-)covariances of these joint distributions, see (Deisenroth and Ohlsson, 2011; Deisenroth et al., 2012). Hence, we have

$$\begin{aligned} \boldsymbol{\mu}^x_{t|t} &= \boldsymbol{\mu}^x_{t|t-1} + \boldsymbol{\Sigma}^{xz}_{t|t-1}(\boldsymbol{\Sigma}^z_{t|t-1})^{-1}(\mathbf{z}_t - \boldsymbol{\mu}^z_{t|t-1}), \qquad (25)\\ \boldsymbol{\Sigma}^x_{t|t} &= \boldsymbol{\Sigma}^x_{t|t-1} - \boldsymbol{\Sigma}^{xz}_{t|t-1}(\boldsymbol{\Sigma}^z_{t|t-1})^{-1}\boldsymbol{\Sigma}^{zx}_{t|t-1}, \\ \boldsymbol{\mu}^x_{t-1|T} &= \boldsymbol{\mu}^x_{t-1|t-1} + \mathbf{J}_{t-1}(\boldsymbol{\mu}^x_{t|T} - \boldsymbol{\mu}^x_{t|t-1}), \\ \boldsymbol{\Sigma}^x_{t|T} &= \boldsymbol{\Sigma}^x_{t-1|t-1} + \mathbf{J}_{t-1}(\boldsymbol{\Sigma}^x_{t|T} - \boldsymbol{\Sigma}^x_{t|t-1})\mathbf{J}^T_{t-1}, \qquad (26) \end{aligned}$$

where we define

$$\mathbf{J}_{t-1} = \boldsymbol{\Sigma}^x_{t-1,t|t-1}(\boldsymbol{\Sigma}^x_{t|t-1})^{-1}. \tag{27}$$

In the following, we first detail the computations required for a Gaussian approximation of the joint distribution $p(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:t-1}, g)$ using moment matching. Here, we approximate the joint distribution $p(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:t-1}, g)$ by the Gaussian

$$\mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}^x_{t-1|t-1} \\ \boldsymbol{\mu}^x_{t|t-1} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}^x_{t-1|t-1} & \boldsymbol{\Sigma}^x_{t-1,t|t-1} \\ \boldsymbol{\Sigma}^x_{t,t-1|t-1} & \boldsymbol{\Sigma}^x_{t|t-1} \end{bmatrix}\right). \tag{28}$$

[3] We use the short-hand notation $\mathbf{a}^d_{b|c}$ where $\mathbf{a} = \boldsymbol{\mu}$ denotes the mean $\boldsymbol{\mu}$ and $\mathbf{a} = \boldsymbol{\Sigma}$ denotes the covariance, b denotes the time step of interest, c denotes the time step up to which we consider measurements, and $d \in \{x, z\}$ denotes either the latent space (x) or the observed space (z).

Without loss of generality, the marginal distribution $\mathcal{N}(\mathbf{x}_{t-1}\,|\,\boldsymbol{\mu}^x_{t-1|t-1}, \boldsymbol{\Sigma}^x_{t-1|t-1})$, which corresponds to the filter distribution at time step $t-1$, is assumed known. We compute the remaining elements of the mean and covariance in Eq. (28) in the following paragraphs. We will derive our results for the more general case where we have a joint Gaussian distribution $p(\mathbf{x}_{t-1}, \mathbf{x}_t, g|\mathbf{z}_{1:t-1})$. The known mean and covariance of distribution $p(\mathbf{x}_{t-1}, g|\mathbf{z}_{1:t-1})$ are given by $\tilde{\boldsymbol{\mu}}_{t-1|t-1} = [(\boldsymbol{\mu}^x_{t-1|t-1})^T, \boldsymbol{\mu}_g^T]^T$ and $\tilde{\boldsymbol{\Sigma}}_{t-1|t-1}$, respectively, where the covariance matrix $\tilde{\boldsymbol{\Sigma}}_{t-1|t-1}$ is block-diagonal with blocks $\boldsymbol{\Sigma}^x_{t-1|t-1}$ and $\boldsymbol{\Sigma}_g$. By setting the mean $\boldsymbol{\mu}_g = g$ and $\boldsymbol{\Sigma}_g = \mathbf{0}$, we obtain the results from (Wang et al., 2012b). For convenience, we define $\tilde{\mathbf{x}} = [\mathbf{x}^T, g]^T$.

Using the law of iterated expectations, the a-th dimension of the predictive *mean of the marginal* $p(\mathbf{x}_t|\mathbf{z}_{1:t-1})$ is given as

$$(\mu^x_{t|t-1})_a = \mathbb{E}_{\mathbf{x}_{t-1}}\left[\mathbb{E}_{f_a}[f_a(\tilde{\mathbf{x}}_{t-1})|\tilde{\mathbf{x}}_{t-1}]|\mathbf{z}_{1:t-1}\right]$$
$$= \int m_x^a(\tilde{\mathbf{x}}_{t-1})p(\tilde{\mathbf{x}}_{t-1}|\mathbf{z}_{1:t-1})d\tilde{\mathbf{x}}_{t-1}\,,$$

where we substituted the posterior GP mean function for the inner expectation. Note that if g is given then $\boldsymbol{\Sigma}_g = \mathbf{0}$. Writing out the posterior mean function and defining $\boldsymbol{\gamma}_a := \mathbf{K}_x^{-1}\mathbf{y}_a$, with $y_{a_i}, i = 1, \ldots, M$, being the training targets of the GP with target dimension a, we obtain

$$(\mu^x_{t|t-1})_a = \mathbf{q}^\top \boldsymbol{\gamma}_a\,,$$

where we define

$$\mathbf{q}^T = \int k_x([\mathbf{x}_{t-1}, g], \mathbf{X}_x)p(\tilde{\mathbf{x}}_{t-1}|\mathbf{z}_{1:t-1})d\tilde{\mathbf{x}}_{t-1}\,. \tag{29}$$

Here, $\mathbf{X}_x$ denotes the set of the M GP training inputs $\tilde{\mathbf{x}}_i = [\mathbf{x}_i^T, g_i]^T$ of the transition GP. Since k_x is a Gaussian kernel, we can solve the integral in Eq. (29) analytically and obtain the vector $\mathbf{q}$ with entries q_i with $i = 1, \ldots, M$ as

$$q_i = \alpha_1|\boldsymbol{\Omega}|^{-\frac{1}{2}}\exp\left(-\tfrac{1}{2}\zeta_i^T(\boldsymbol{\Lambda\Omega})^{-1}\zeta_i\right), \tag{30}$$

$$\zeta_i = \tilde{\mathbf{x}}_i - \tilde{\boldsymbol{\mu}}_{t-1|t-1}\,, \quad \boldsymbol{\Omega} = \boldsymbol{\Sigma}_{t-1|t-1}\boldsymbol{\Lambda}^{-1} + \mathbf{I}\,, \tag{31}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix of concatenated length-scales $\alpha_2\mathbf{I}$ and $\alpha_3\mathbf{I}$. By applying the law of total variances, the entries σ^x_{ab} of the *marginal predictive covariance matrix* $\boldsymbol{\Sigma}^x_{t|t-1}$ in Eq. (28) are given by

$$\sigma^x_{ab} = \begin{cases} \boldsymbol{\gamma}_a^T(\mathbf{Q}^x - \mathbf{q}\mathbf{q}^T)\boldsymbol{\gamma}_b & \text{if } a \neq b\,,\\ \boldsymbol{\gamma}_a^T(\mathbf{Q}^x - \mathbf{q}\mathbf{q}^T)\boldsymbol{\gamma}_b + \alpha_1 - \operatorname{tr}\left((\mathbf{K}_x + \alpha_4\mathbf{I})^{-1}\mathbf{Q}^x\right) + \alpha_4 & \text{if } a = b\,. \end{cases}$$

We define the entries of $\mathbf{Q}^x \in \mathbb{R}^{M\times M}$ as

$$Q^x_{ij} = \frac{k_x^a([\mathbf{x}_i, g_i], [\tilde{\boldsymbol{\mu}}_{t-1|t-1}])k_x^b([\mathbf{x}_j, g_j], [\tilde{\boldsymbol{\mu}}_{t-1|t-1}])}{\sqrt{|\mathbf{R}|}}\exp\left(\tfrac{1}{2}\mathbf{z}_{ij}^T\mathbf{T}^{-1}\mathbf{z}_{ij}\right)$$

with

$$\mathbf{R} := \tilde{\boldsymbol{\Sigma}}_{t-1|t-1}(\boldsymbol{\Lambda}_a^{-1} + \boldsymbol{\Lambda}_b^{-1}) + \mathbf{I}\,, \qquad \mathbf{T} = \left(\boldsymbol{\Lambda}_a^{-1} + \boldsymbol{\Lambda}_b^{-1} + \tilde{\boldsymbol{\Sigma}}_{t-1|t-1}^{-1}\right),$$
$$\mathbf{z}_{ij} := \boldsymbol{\Lambda}_a^{-1}(\tilde{\mathbf{x}}_i - \tilde{\boldsymbol{\mu}}_{t-1|t-1}) + \boldsymbol{\Lambda}_b^{-1}(\tilde{\mathbf{x}}_j - \tilde{\boldsymbol{\mu}}_{t-1|t-1})\,.$$

For a detailed derivation, we refer to (Deisenroth, 2010; Deisenroth et al., 2012).

To fully determine the joint Gaussian distribution in Eq. (28), the cross-covariance $\boldsymbol{\Sigma}^x_{t-1,t|t-1} = \mathrm{cov}[\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:t-1}, g]$ is given as the upper part of the cross-covariance

$$\mathrm{cov}[\mathbf{x}_{t-1}, \mathbf{x}_t, g|\mathbf{z}_{1:t-1}] = \sum_{i=1}^{M} \gamma_{a_i} q_{a_i} \tilde{\boldsymbol{\Sigma}}_{t-1|t-1} \boldsymbol{\Omega}^{-1}(\tilde{\mathbf{x}}_i - \tilde{\boldsymbol{\mu}}_{t-1|t-1})\,,$$

when we set $\boldsymbol{\mu}_g = g$ and $\boldsymbol{\Sigma}_g = \mathbf{0}$. Note that $\mathbf{q}$ and $\boldsymbol{\Omega}$ are defined in Eq. (30) and (31), respectively.

Up to now, we have computed a Gaussian approximation to the joint probability distribution $p(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:t-1}, g)$. Let us now have closer look at the second joint distribution $p(\mathbf{x}_t, \mathbf{z}_t|\mathbf{z}_{1:t-1}, g)$, which is the missing contribution for Gaussian smoothing (Deisenroth and Ohlsson, 2011), see Eq. (25)–(26). To determine a Gaussian approximation

$$\mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}^x_{t|t-1} \\ \boldsymbol{\mu}^z_{t|t-1} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}^x_{t|t-1} & \boldsymbol{\Sigma}^{xz}_{t|t-1} \\ \boldsymbol{\Sigma}^{zx}_{t|t-1} & \boldsymbol{\Sigma}^z_{t|t-1} \end{bmatrix}\right) \tag{32}$$

to $p(\mathbf{x}_t, \mathbf{z}_t|\mathbf{z}_{1:t-1}, g)$ it remains to compute the mean and the covariance of the marginal distribution $p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, g)$ and the cross-covariance terms $\mathrm{cov}[\mathbf{x}_t, \mathbf{z}_t|\mathbf{z}_{1:t-1}, g]$. We omit these computations for the nonlinear Gaussian kernel as they are very similar to the computations to determine the joint distribution $p(\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:t-1}, g)$.

For the *linear measurement kernel* in Eq. (20), we compute the *marginal mean* $\boldsymbol{\mu}^z_{t|t-1}$ in Eq. (32) for observation dimension $a = 1, \ldots, D_z$ according to

$$\begin{aligned} \mathbb{E}_{h,\mathbf{x}_{t-1}}[h_a(\mathbf{x}_t)|\mathbf{z}_{1:t-1}, g] &= \int m(\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, g)\,\mathrm{d}\mathbf{x}_t \\ &= \int \mathbf{x}_t^T \mathbf{X}_z^T p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, g)\,\mathrm{d}\mathbf{x}_t \boldsymbol{\xi}_a = \mathbf{q}^T \boldsymbol{\xi}_a\,, \\ \mathbf{q} &= \mathbf{X}_z \boldsymbol{\mu}^x_{t|t-1}\,. \end{aligned} \tag{33}$$

Here, $\mathbf{X}_z$ comprises the training inputs for the measurement model and $\boldsymbol{\xi}_a = \mathbf{K}_z^{-1}\mathbf{Y}_{z_a}$, where $\mathbf{Y}_{z_a}$ are the training targets of the ath dimension, $a = 1, \ldots, D_z$. The elements σ^z_{ab} of the *marginal covariance matrix* $\boldsymbol{\Sigma}^z_{t|t-1}$ in Eq. (32) are given as

$$\sigma^z_{ab} = \begin{cases} \boldsymbol{\xi}_a^T(\mathbf{Q}^z - \mathbf{q}\mathbf{q}^T)\boldsymbol{\xi}_a & \text{if } a \neq b\,, \\ \boldsymbol{\Sigma}^x_{t|t-1} + \boldsymbol{\mu}^x_{t|t-1}(\boldsymbol{\mu}^x_{t|t-1})^T - \mathrm{tr}\left(\mathbf{K}_z^{-1}\mathbf{Q}^z\right) + \boldsymbol{\xi}_a^T(\mathbf{Q}^z - \mathbf{q}\mathbf{q}^T)\boldsymbol{\xi}_a & \text{if } a = b\,, \end{cases} \tag{34}$$

$a, b = 1, \dots, D_z$, where we define

$$\mathbf{Q}^z = \int \mathbf{X}_z \mathbf{x}_t \mathbf{x}_t^T {\mathbf{X}_z}^T p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, g)\, \mathrm{d}\mathbf{x}_t = \mathbf{X}_z(\boldsymbol{\Sigma}^x_{t|t-1} + \boldsymbol{\mu}^x_{t|t-1}(\boldsymbol{\mu}^x_{t|t-1})^T){\mathbf{X}_z}^T \,.$$

The *cross-covariance* $\boldsymbol{\Sigma}^{xz}_{t|t-1} = \mathrm{cov}[\mathbf{x}_t, \mathbf{z}_t|\mathbf{z}_{1:t-1}, g]$ in Eq. (32) is given as

$$\boldsymbol{\Sigma}^{xz}_{t|t-1} = \boldsymbol{\Sigma}^x_{t|t-1}\mathbf{X}_z^T \boldsymbol{\xi}_a \tag{35}$$

for all observed dimensions $a = 1, \dots, D_z$. The mean $\boldsymbol{\mu}^z_{t|t-1}$ in Eq. (33), the covariance matrix $\boldsymbol{\Sigma}^z_{t|t-1}$ in Eq. (34), and the cross-covariance in Eq. (35) fully determine the Gaussian distribution in Eq. (32). Hence, following (Deisenroth and Ohlsson, 2011), we can now compute the latent state posteriors (filter and smoothing distributions) according to Eq. (25)–(26).

These smoothing updates in Eq. (25)–(26) yield the marginals of our Gaussian approximation to $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)$, see Eq. (24). The missing cross-covariances $\boldsymbol{\Sigma}^x_{t-1,t|T}$ of $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)$ that finally fully determine the block-tri-diagonal covariance matrix in Eq. (24) are given by

$$\boldsymbol{\Sigma}^x_{t-1,t|T} = \mathbf{J}_{t-1}\boldsymbol{\Sigma}^x_{t|T} \,,$$

where $\mathbf{J}_{t-1}$ is given in Eq. (27). For detailed derivations, we refer to (Deisenroth, 2010).

These computations conclude step (1) on lower-bounding the posterior distribution on the intention, see Eq. (22), i.e., the computation of the approximate distribution q in Eq. (23). It remains to compute the bound $\mathscr{B}$ itself, which is described in the following.

3.3.2 Estimating the Belief on Intention

For a given intention g, we compute a Gaussian approximation $q(\mathbf{x}_{1:T})$ to the posterior $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)$, given by

$$q(\mathbf{x}_t, \mathbf{x}_{t+1}) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}^x_{t|T} \\ \boldsymbol{\mu}^x_{t+1|T}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}^x_{t|T} & \boldsymbol{\Sigma}^x_{t,t+1|T} \\ \boldsymbol{\Sigma}^x_{t+1,t|T} & \boldsymbol{\Sigma}^x_{t+1|T}\end{bmatrix}\right)$$

for $t = 1, \dots, T-1$. The belief $p(g|\mathbf{z}_{1:T}) \approx \exp(\mathscr{B}(g))$ is estimated using Eq. (22), where the computation can be decomposed according to

$$\mathscr{B}(g) = \sum_{t=1}^{T-1} \underbrace{\mathbb{E}_q\left[\log p(\mathbf{x}_{t+1}|\mathbf{x}_t, g)\right]}_{\mathscr{Q}_t(g)} + p(g) + \mathscr{H}(q) + \mathrm{const.}$$

Here the smoothing distribution $q(\mathbf{x}_{1:T}|g) \approx p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)$ is computed given the intention g. As we only need to estimate the unnormalized belief, the constant term needs not to be computed. The entropy $\mathscr{H}(q)$ of the Gaussian distribution q can be computed analytically, and is given by

$$\mathscr{H}(q) = \frac{1}{2}\left(TD_x + TD_x\log(2\pi) + \log|\boldsymbol{\Sigma}_q|\right).$$

We define

$$\begin{aligned}\mathcal{Q}_t(g) &\triangleq \mathbb{E}_q\left[\log p(\mathbf{x}_{t+1}|\mathbf{x}_t, g)\right]\\ &= \iint q(\mathbf{x}_t, \mathbf{x}_{t+1}) \log p(\mathbf{x}_{t+1}|\mathbf{x}_t, g) d\mathbf{x}_{t+1} d\mathbf{x}_t\\ &= \iint q(\mathbf{x}_t, \mathbf{x}_{t+1}) \log \underbrace{\left(p(\mathbf{x}_{t+1}|\mathbf{x}_t, g) q(\mathbf{x}_t)\right)}_{\approx \tilde{q}(\mathbf{x}_t, \mathbf{x}_{t+1})} d\mathbf{x}_{t+1} d\mathbf{x}_t - \int q(\mathbf{x}_t) \log q(\mathbf{x}_t) d\mathbf{x}_t,\end{aligned}$$

where $p(\mathbf{x}_{t+1}|\mathbf{x}_t, g)q(\mathbf{x}_t)$ can be approximated by a Gaussian distribution $\tilde{q}(\mathbf{x}_t, \mathbf{x}_{t+1}) = \mathcal{N}(\boldsymbol{\mu}_{\tilde{\mathbf{q}}}, \boldsymbol{\Sigma}_{\tilde{\mathbf{q}}})$ based on moment matching (Quiñonero-Candela et al., 2003). Here, we only compute the diagonal elements in the covariance matrix of $\boldsymbol{\Sigma}_{\tilde{\mathbf{q}}}$. As a result, Eq. (3.3.2) is approximated as

$$\mathcal{Q}_t(g) \approx \mathrm{KL}\big(q(\mathbf{x}_t, \mathbf{x}_{t+1}) || \tilde{q}(\mathbf{x}_t, \mathbf{x}_{t+1})\big) + \mathscr{H}\big(q(\mathbf{x}_t, \mathbf{x}_{t+1})\big) + \mathscr{H}\big(q(\mathbf{x}_t)\big),$$

where $\mathscr{H}(q)$ is the entropy of the distribution q and $\mathrm{KL}(q||\tilde{q})$ is the Kullback-Leibler (KL) divergence between q and $\tilde{q}$, both of which are Gaussians. The KL divergence also has a closed-form expression, given by

$$\mathrm{KL}(q||\tilde{q}) = \frac{1}{2}\left(\mathrm{tr}(\boldsymbol{\Sigma}_{\tilde{q}}^{-1}\boldsymbol{\Sigma}_q) + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_{\tilde{q}})^T \boldsymbol{\Sigma}_{\tilde{q}}^{-1}(\boldsymbol{\mu}_q - \boldsymbol{\mu}_{\tilde{q}}) - \log\frac{|\boldsymbol{\Sigma}_q|}{|\boldsymbol{\Sigma}_{\tilde{q}}|}\right) + \mathrm{const}.$$

As a result, we can compute the unnormalized belief $\mathscr{B}(g)$ for a given intention g approximately according to Eq. (3.3.2).

We aim to determine the *posterior distribution* $p(g|\mathbf{z}_{1:T})$ of the intention g. Using the posterior distribution instead of point estimates allows us to express uncertainty about the inferred intention g. Computing Gaussian approximations of the posterior distributions can be done using the unscented transformation (Deisenroth et al., 2012), for instance. However, when the posterior is not unimodal, a Gaussian approximation may lose important information. Particle filtering can preserve all the modes (Ko and Fox, 2009), but will not be sufficiently efficient due to the real-time constraints. As we focus on one-dimensional intentions in this chapter, we advocate the discretization of intention. For example, in the table tennis task, the intention (opponent's target position) is a bounded scalar variable $g \in [g_{\min}, g_{\max}]$, where the bounds are given by physical constraints such as the table width and the length of robot arm. We uniformly choose $\{v_1, \dots, v_K\}$ from $[g_{\min}, g_{\max}]$ and represent intention by the index, i.e., $g \in \{1, \dots, K\}$.

3.3.3 Discussion of the Approximate Inference Method

To summarize, the algorithm for computing the posterior distribution over discrete or discretized intentions g is given in Algorithm 3.1. The smoothing distribution q defined in Eq. (3.3.2) depends on the current estimate of intention g.

However, it is often time-demanding to enumerate the intention g and compute the smoothing distribution q for each g individually. The computational complexity of the smoothing step in Algorithm 3.1 is $\mathcal{O}(TK(D_z^3 + D_x D_z^2 + N^2 D_x^3))$ when using the linear kernel function for the measurement mapping, and $\mathcal{O}(TK(D_z^3 + N^2 D_x(D_x^2 + D_z^2)))$ when using the Gaussian kernel function,

Algorithm 3.1: Inference of the discretized intentions by computing the posterior probabilities for every value of the intention.

```
Input  : Observations $\mathbf{x}_{1:T}$
Output: Posterior probabilities for every intention value $g \in \{1, \ldots, K\}$
foreach $g \in \{1, \ldots, K\}$ do
    Compute smoothing distribution $q(\mathbf{x}_{1:T}) \approx p(\mathbf{x}_{1:T} | \mathbf{z}_{1:T}, g)$ ;
    Compute the value of $\mathscr{B}(g) = \mathbb{E}_q \left[ \log p(\mathbf{x}_{1:T} | g) \right] + \log p(g)$ using the approximation in
    Eq. (3.3.2) ;
Estimate the posterior $p(g | \mathbf{x}_{1:T}) \approx \exp \mathscr{B}(g) / (\sum_{g'=1}^{K} \exp \mathscr{B}(g'))$.
```

where T is the number of observations obtained, K the number of (discretized) intentions, N the number of training data, and D_x and D_z the dimensionality of state and observation. The complexity of computing the belief is $\mathcal{O}(TKN^2 D_x^2)$. The computational efficiency can be improved to meet the tight time constraints in robotic applications by introducing further approximations, such as adopting GP pseudo inputs to reduce the size of training data N (Snelson and Ghahramani, 2006; Quiñonero-Candela and Rasmussen, 2005), using dimensionality reduction or feature selection techniques to obtain a small number of features D_z (van der Maaten et al., 2009; Ding and Peng, 2005), and reducing the sample size K of intention g. However, the dependence of complexity on the number of observations T still prevents the algorithm from being applied to online scenarios. For these, T keeps growing as new observations are obtained, whereas observations obtained a long time ago do not provide as much information as recent ones. To address this issue, we will introduce an approximation in the online inference method in Section 3.4.

3.4 Online Intention Inference

The introduced inference algorithm can be seen as a batch algorithm that relies on the segmentation of human movements. However, in online human-robot interaction, the intention inference algorithm faces new challenges to deal with the stream of observations. The complexity of Algorithm 3.1 grows with the number of existing observations, which does not fulfill the real-time requirements of an online method. In addition, the intention can vary over time in an online inference scenario. For example, the intended targets in table tennis games vary between strokes. Hence, the online method should model and track the change of intention.

To address these issues, we generalize the inference method to an online scenario. That is, the observations are obtained constantly, and the belief on the intention is re-estimated after receiving a new observation. A computational bottleneck in the batch method is that the smoothing distribution q is computed for every value of intention. For efficient inference, we compute a marginal smoothing distribution q according to current belief on intention $p(g)$, i.e., we integrate out the intention,

$$q(\mathbf{x}_{1:t}) \triangleq \sum_{g} p(g) q_g(\mathbf{x}_{1:t}).$$

The online inference algorithm then estimates the belief $\mathscr{B}_t(g)$ on the intention based on the marginal smoothing distribution q after receiving an observation, which can be sufficiently efficient for real-time intention inference with a small sacrifice in accuracy.

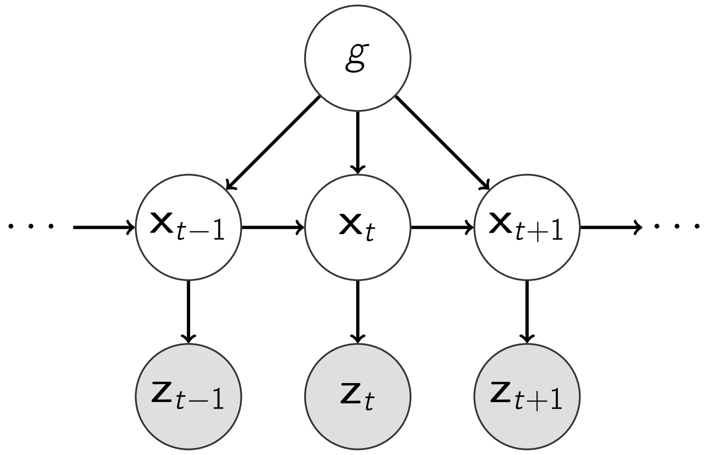


Figure 3.4: The graphical model of the IDDM in an online manner, which can handle a stream of observations.

Based on the marginal smoothing distribution, we update the belief on intention using dynamic programming. which will be discussed as follows.

3.4.1 Online Inference using Dynamic Programming

Assuming the marginal smoothing distribution q is given, we develop an online inference method using dynamic programming (see Figure 3.4). The method maintains the belief (i.e., log of the unnormalized posterior) of the intention g based on the obtained observations $\mathbf{z}_{1:t-1}$ according to Eq. (3.3.2), given by

$$\mathscr{B}_{t-1}(g) \approx \mathbb{E}_q \left[\log p(g, \mathbf{x}_{1:t-1})\right] + \text{const}.$$

Here, we consider discretized intentions $g \in \{1, \dots, K\}$, and write the belief $\mathscr{B}_{t-1}$ as a vector of length K. For a new observation $\mathbf{z}_t$, we decompose $p(g, \mathbf{x}_{1:t})$ according to

$$p(g, \mathbf{x}_{1:t}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}, g) p(g, \mathbf{x}_{1:t-1}).$$

As a result, the belief $\mathscr{B}_t$ becomes

$$\begin{aligned}\mathscr{B}_t(g) &= \mathbb{E}_q \left[\log p(g, \mathbf{x}_{1:t})\right] + \text{const} \\ &= \mathbb{E}_q \left[\log p(\mathbf{x}_t | \mathbf{x}_{t-1}, g)\right] + \mathbb{E}_q \left[\log p(g, \mathbf{x}_{1:t-1})\right] + \text{const} \\ &= \mathbb{E}_q \left[\log p(\mathbf{x}_t | \mathbf{x}_{t-1}, g)\right] + \mathscr{B}_{t-1}(g) + \text{const},\end{aligned}$$

which is in a recursive form and can be computed efficiently using dynamic programming. Given a new observation $\mathbf{z}_t$, the belief is updated based on $\mathbb{E}_q \left[\log p(\mathbf{x}_t | \mathbf{x}_{t-1}, g)\right]$, which is computed according to Eqs. (3.3.2)-(3.3.2). The belief $\mathscr{B}_t$ is then normalized, i.e., $\sum_g \exp(\mathscr{B}_t(g)) = 1$.

In addition, the intention can vary over time in an online inference scenario. As the new observation $\mathbf{z}_t$ can be more informative than the previous observations $\mathbf{z}_{1:t-1}$, we introduce a forgetting factor ϵ to shrink the belief $\mathscr{B}_{t-1}$. The recursive formula of the belief is subsequently given by

$$\mathscr{B}_t(g) = \mathbb{E}_q \left[\log p(\mathbf{x}_t | \mathbf{x}_{t-1}, g)\right] + (1 - \epsilon) \mathscr{B}_{t-1}(g),$$

where the shrinking factor ϵ determines how fast the algorithm forgets the previous observations.

Algorithm 3.2: The online algorithm for the inference of discrete intention $g \in \{1, \ldots, K\}$.

```
Obtain the initial observation $\mathbf{z}_1$ ;
Initialize the approximate distribution $q(\mathbf{x}_1)$ ;
Initialize $\mathscr{B}_1(g) = \log p(g)$ according to the prior ;
for $t = 2, 3, \ldots$ do
    Obtain the observation $\mathbf{z}_t$ ;
    Compute marginal filtering distribution $q(\mathbf{x}_t)$ according to current belief $\mathscr{B}_{t-1}$ ;
    Update marginal smoothing distribution $q(\mathbf{x}_{t-1})$ according to current belief $\mathscr{B}_{t-1}$ ;
    foreach $g_t = \{1, \ldots, K\}$ do
        Compute $\mathscr{B}^0(g) = \mathscr{Q}_{t-1}(g)$ using the approximation in Eq. (3.3.2) ;
    Update the belief $\mathscr{B}_t = \mathscr{B}^0 + (1 - \epsilon)\mathscr{B}_{t-1}$ ;
    Normalize the belief $\mathscr{B}_t \leftarrow \mathscr{B}_t - \log\left(\sum_g \exp(\mathscr{B}_t(g))\right)$ ;
```

3.4.2 Marginal Smoothing Distribution

The inference method relies on the smoothing distribution q at time t, which in turn depends on the intention belief $\mathscr{B}_{t-1}$. In analogy to the EM algorithm, we iteratively update the belief on intention $\mathscr{B}$ and the smoothing distribution q. However, full forward-backward smoothing on $\mathbf{x}_{1:t}$ is impractical as the computational complexity grows when we obtain more observations. Full smoothing is also unnecessary since we do not update the previous belief $\mathscr{B}_{1:t-1}$ on the intention. Hence, given a new observation $\mathbf{z}_t$, we only need to compute $q(\mathbf{x}_{t-1:t})$, which requires a single-step forward filtering and a single-step backward smoothing, based on the current belief $\mathscr{B}_{t-1}$.

The filtering and smoothing need to integrate out the uncertainty in the intention. For discrete intentions, we can simply compute the smoothing distributions q_g for every value of intention g_{t-1}, and average over them

$$q(\mathbf{x}_{t-1:t}) \propto \sum_g q_g(\mathbf{x}_{t-1:t}) p_{t-1}(g),$$

where the belief $p_{t-1}(g) \propto \exp(\mathscr{B}_{t-1}(g))$. The resulting distribution q will still be a Gaussian distribution.

For continuous intentions, enumerating the discretized intention may be inefficient. To address this problem, we use the moment matching to approximate the distribution on intention by a Gaussian distribution, which is also adopted in the filtering and smoothing method. Specifically, we compute the mean μ_g and variance σ_g^2 according to the belief $\mathscr{B}_{t-1}$. As a result, the marginal smoothing distribution is given by

$$q(\mathbf{x}_{t-1:t}) \approx \int q_g(\mathbf{x}_{t-1:t}) \mathcal{N}(g | \mu_g, \sigma_g^2) dg,$$

which is computed using moment matching.

3.4.3 Discussion of the Online Inference Method

The online inference algorithm described in Algorithm 3.2 iteratively updates the belief of intention and latent states.

The computational complexity of the smoothing step in Algorithm 3.2 is $\mathcal{O}(D_z^3+D_xD_z^2+N^2D_x^3)$ when using the linear kernel function for the measurement mapping, and $\mathcal{O}(D_z^3+N^2(D_xD_x^2+D_x^3))$ when using the Gaussian kernel function, which no longer depends on the number of observations T and the number of intentions K. The complexity of computing the belief is $\mathcal{O}(KN^2D_x^2)$. Compared to the batch algorithm, the efficiency is improved by a factor of T.

To summarize, we proposed an efficient online method for intention inference from a new movement. The online method updates the belief of the intention by taking into account both the current belief and the new evidence (i.e., new observation). We list the employed approximations in both the batch and online inference methods in Table 3.1.

3.5 Target Prediction for Robot Table Tennis

Playing table tennis is a challenging task for robots, and, hence, has been used by many researchers as a benchmark task in robotics (Anderson, 1988; Billingsley, 1984; Fässler et al., 1990; Matsushima et al., 2005; Mülling et al., 2011). Up to now, none of the groups that have been working on robot table tennis ever reached the level of a young child, despite having robots with better perception, processing power, and accuracy than humans (Mülling et al., 2011). Likely explanations for this performance gap are (i) the human ability to predict hitting points from opponent movements and (ii) the robustness of human hitting movements (Mülling et al., 2011). Here, we focus on the first issue: anticipation of the hitting region from opponent movements.

Using the proposed method, we can predict the where the ball is likely to be shot *before* the opponent hits the ball, which gives the robot a head start of more than 200 ms additional time to initiate its movement[4]. This additional time can be crucial due to robot's hardware constraints, for example, acceleration and torque limits in the considered setting (Mülling et al., 2011).

Note that the predicted intention is only used to choose a hitting type, e.g., forehand, middle, or backhand. Fine-tuning of the robot's movement can be done when the robot is adjusted to the forehand/middle/backhand preparation pose and once the returned ball can be reliably predicted from the ball's trajectory alone. Hence, a certain amount of intention prediction error

[4] Our methods allows the robot to initiate its movement at least 80 ms before the opponent hits the ball. As the ball can usually be reliably predicted more than 120 ms after the opponent returns, the robot could gain more than 200 ms additional execution time by using our prediction method.

Table 3.1: Important approximations employed in the batch and online inference.

	batch	online
belief $p(g\|\mathbf{z}_{1:T})$	Jensen's lower bound $\mathscr{B}(g)$; cf. Eq. (22)	
approx. belief $\mathscr{B}(g)$	moment matching; cf. Eq. (3.3.2)	
distr. $p(\mathbf{x}_{1:T}\|\mathbf{z}_{1:T}, g)$	$q(\mathbf{x}_{1:T}\|g)$ for each g	$q(\mathbf{x}_{1:T})$ for all g; cf. Eq. (3.4.2)
stream of observations	sliding window	recursive update; cf. Eq. (3.4.1)

is tolerable since the robot can apply small changes to its basic hitting plan based on the ball's trajectory. However, the robot cannot return the ball outside the corresponding hitting region once it is adjusted to a preparation pose, see the video[5]. Therefore, prediction accuracy directly influences the performance of the robot player (Wang et al., 2011b).

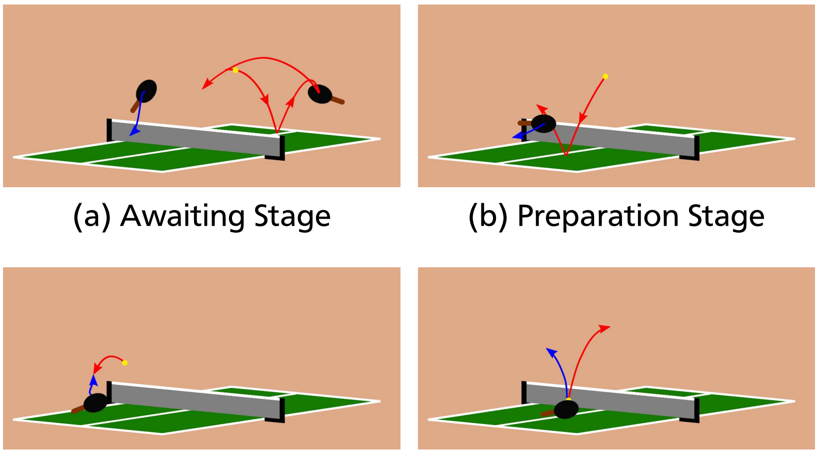
3.5.1 Experimental Setting

Our anticipation system has been evaluated in conjunction with the biomimetic robot table tennis player (Mülling et al., 2011), as this setup allowed exhibiting how much of an advantage such a system may offer. We expect that the system will help similarly or more when deployed within our skill learning framework (Mülling et al., 2013) as well as many of the recent table tennis learning systems (Huang et al., 2013; Yang et al., 2010; Matsushima et al., 2005).

We used a Barrett WAM robot arm to play table tennis against human players. The robot's hardware constraints impose strong limitations on its acceleration, which severely restricts its movement abilities. This limitation can best be illustrated using typical table tennis stroke movements as shown in Figure 3.5, see (Ramanantsoa and Durey, 1994; Mülling et al., 2011), which consist of four stages, namely *awaiting stage*, *preparation stage*, *hitting stage*, and *finishing stage*. In the awaiting stage, the ball moves toward the opponent and is returned by the opponent. The robot player moves to the awaiting pose and stays there during this stage. The preparation stage starts when the hitting movement is chosen according to the predicted opponent's target. The arm swings backward to a preparation pose. The robot requires sufficient time to execute a ball-hitting plan in the hitting stage. To achieve the required velocity for returning the ball in the hitting stage, movement initiation to an appropriate preparation pose in the preparation stage is needed, which is often *before* the opponent hits the ball. The robot player uses different preparation poses for different hitting plans. Hence, it is necessary to choose among them based on modeling the opponent's preference (Wang et al., 2011a) and inference of the opponent's target location for the ball (Wang et al., 2011b).

The robot perceives the ball and the opponent's racket in real-time, using seven Prosilica GE640C cameras. These cameras were synchronized and calibrated to the coordinate system of the robot. The ball tracking system uses four cameras to capture the ball on both courts of the table (Lampert and Peters, 2012). The racket tracking system provides the information of the opponent's racket, i.e., the position and orientation (Wang et al., 2011b). See Appendix 3.A for more details of the vision system. As a result, the observation $\mathbf{z}_t$ includes the ball's position and velocity as well as the opponent's racket position, velocity, and orientation before the human plays the ball. For the anticipation system described here, we process the observations every 80 ms. Here, the position and velocity of the ball were processed online with an extended Kalman filter, based on a known physical model (Mülling et al., 2011). However, the same smoothing method cannot be applied to the racket's trajectory, as its dynamics are directed by the unknown intended target. Therefore, the obtained states of the racket were subject to substantial noise and the model has to be robust to this noise. The proposed inference method can jointly smooth on the racket's trajectory, given by the smoothing distribution q, and infer the intended target, given by the belief $\mathscr{B}$.

[5] http://robot-learning.de/Research/ProbabilisticMovementModeling



(a) Awaiting Stage

(b) Preparation Stage

(c) Hitting Stage

(d) Finishing Stage

Figure 3.5: The four stages of a typical table tennis ball rally are shown with the red curve representing the ball trajectories. Blue trajectories depict the typical racket movements of players. The racket of human player is to the left of the table in the pictures. Figures are adapted from Mülling et al. (2011).

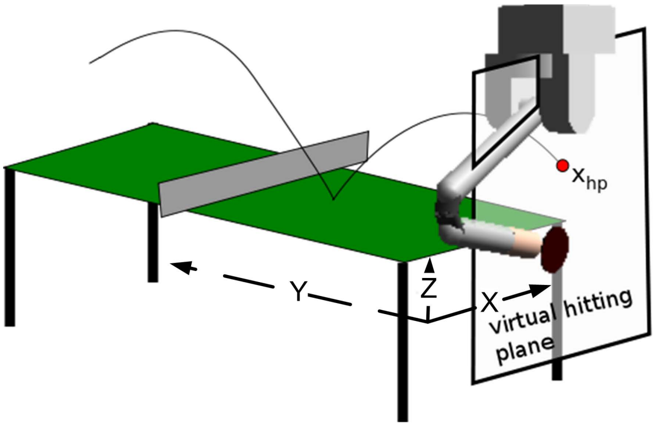


Figure 3.6: The robot's hitting point is the intersection of the coming ball's trajectory and the virtual hitting plane 80 cm behind the table. Figure is adapted from Mülling et al. (2011).

In our setting, the robot always chooses its hitting point on a virtual hitting plane, which is 80 cm behind the table, as shown in Figure 3.6. We define the human's intended target g as the intersection of the returned ball's trajectory with the robot's virtual hitting plane. As the x-coordinate (see Figure 3.6) is most important for choosing among forehand/middle/backhand hitting plans (Wang et al., 2011b), the intention g considered here is the x-coordinate of the hitting point. Physical limitations of the robot restrict the x-coordinate to the range of ± 1.2 m from the robot's base (table is 1.52 m wide).

To evaluate the performance of the target prediction, we collected a data set with recorded stroke movements from different human players. The true targets were obtained from the ball tracking system. The data set was divided into a training set of 100 plays and a test set of 126 plays. The standard deviation of the target coordinate in the test set is 102.2 cm. A

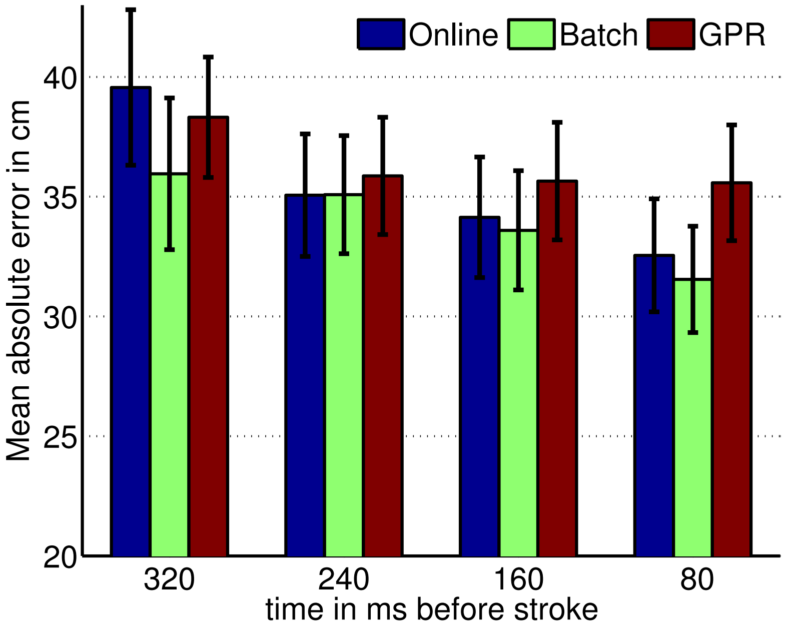


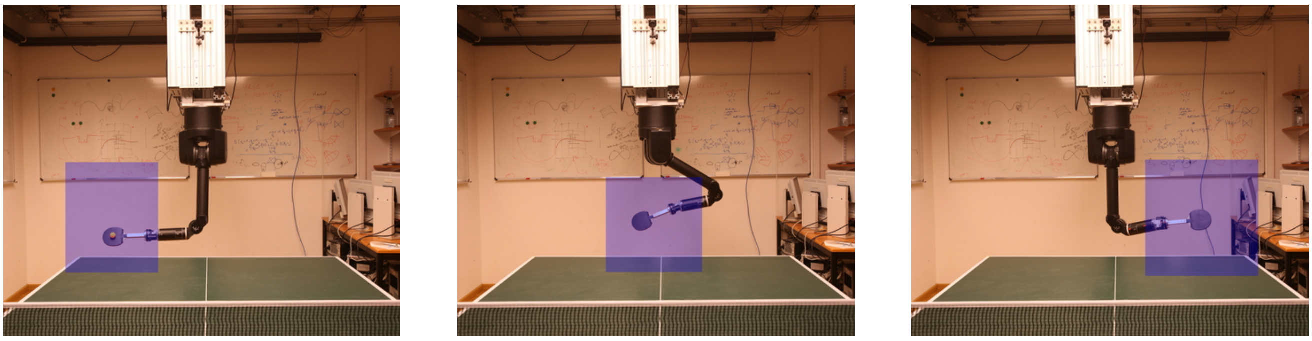
Figure 3.7: Mean absolute error of the ball's target with standard error of the mean. The algorithms use the observations obtained *before* the opponent has hit the ball.

straightforward approach to prediction is to learn a mapping from the features $\mathbf{z}_t$ (including the position, orientation, and velocity of the racket and the position and velocity of the ball) to the target g. We compared our method to this baseline Gaussian Process Regression (GPR) using a Gaussian kernel with automatic relevance determination (Rasmussen and Williams, 2006). We considered using a sliding window on the sequence of observations, and conducted model selection to choose the optimal window size. The best accuracy of GPR was achieved when using a sliding window of size two, i.e., the input features consist of $\mathbf{z}_{t-1}$ and $\mathbf{z}_t$. The hyperparameters were learned by maximizing the marginal likelihood of training data, following the standard routine (Rasmussen and Williams, 2006).

For every recorded play, we compared the performance of the proposed IDDM intention inference and the GPR prediction at 80 ms, 160 ms, 240 ms, and 320 ms *before* the opponent hits the ball. Note that this time step was only used such that the algorithms could be compared, and that the algorithms were not aware of the hitting time of the opponent in advance. We evaluated both the batch algorithm and online algorithm.

3.5.2 Results

As demonstrated in Figure 3.7, the proposed IDDM model outperformed the GPR baseline. At 80 ms before the opponent hit the ball, the batch algorithm resulted in a mean absolute error of 31.5 cm, which achieved a 11.3% improvement over the GPR, whose average error was 35.6 cm. The online algorithm had a mean absolute error of 32.5 cm, which also outperformed GPR by an 8.5% improvement in the accuracy. One model-free naive intention prediction is to always predict the median of the intentions in the training set. This naive prediction model caused an error of 78.8 cm. Hence, both the GPR and IDDM substantially outperformed naive goal prediction.



(a) Forehand pose.

(b) Middle pose.

(c) Backhand pose.

Figure 3.8: Preparation poses of the three pre-defined hitting movements in the prototype system, i.e., (a) forehand, (b) middle, and (c) backhand. The shadowed areas represent the corresponding hitting regions.

The online algorithm, with a shrinking factor $\epsilon = 0.2$ given in Eq. (3.4.1), took on average 70 ms to process every observation, which can potentially fulfill the real-time requirements of 80 ms. The batch algorithm used a sliding window of size 4, and took on average 300 ms to process every observation. The online algorithm was greatly faster than the batch algorithm, with a small loss in accuracy[6]. Nevertheless, a certain amount of error is tolerable since the robot can apply small changes to its basic hitting plan based on the ball's trajectory. Therefore, we advocate the use of the online algorithm for applications with tight real-time constraints.

We performed *model selection* to determine the covariance function k_z, which can be either an isotropic Gaussian kernel, see Eq. (19), or a linear kernel, see Eq. (20). Furthermore, we performed model selection to find the dimension D_x of the latent states. In the experiments, the model was selected by cross-validation on the training set. The best model under consideration was with a linear kernel and a four dimensional latent state space. Experiments on the test set verified the model selection result, as shown in Table 3.2.

Our results demonstrated that the IDDM can improve the target prediction in robot table tennis and choose the correct hitting plan. We have developed a proof-of-concept prototype system in which the robot is equipped with three pre-defined hitting movements, i.e., forehand, middle, and backhand movements, with their hitting regions shown in Figure 3.8. As exhibited in Figure 3.9, our method allows the robot to choose the responding hitting movement before

[6] The reason is that the online algorithm only updates the smoothing distribution $q(\mathbf{x}_{t-1:t})$ instead of the entire smoothing distribution $q(\mathbf{x}_{1:T})$, and, hence, reduces the time complexity by a factor of T, see Section 3.3.3 and 3.4.3

Table 3.2: The mean absolute errors (in cm) with standard error of the mean of the goal inference made 80 ms *before* the opponent hits the ball, where D_x denotes the dimensionality of the state space.

kernel	$D_x = 3$	$D_x = 4$	$D_x = 5$	$D_x = 6$
linear	41.5 ± 3.0	**31.5 ± 2.2**	35.4 ± 2.4	37.0 ± 2.6
Gaussian	38.5 ± 2.7	34.2 ± 2.5	34.4 ± 2.7	37.3 ± 2.7

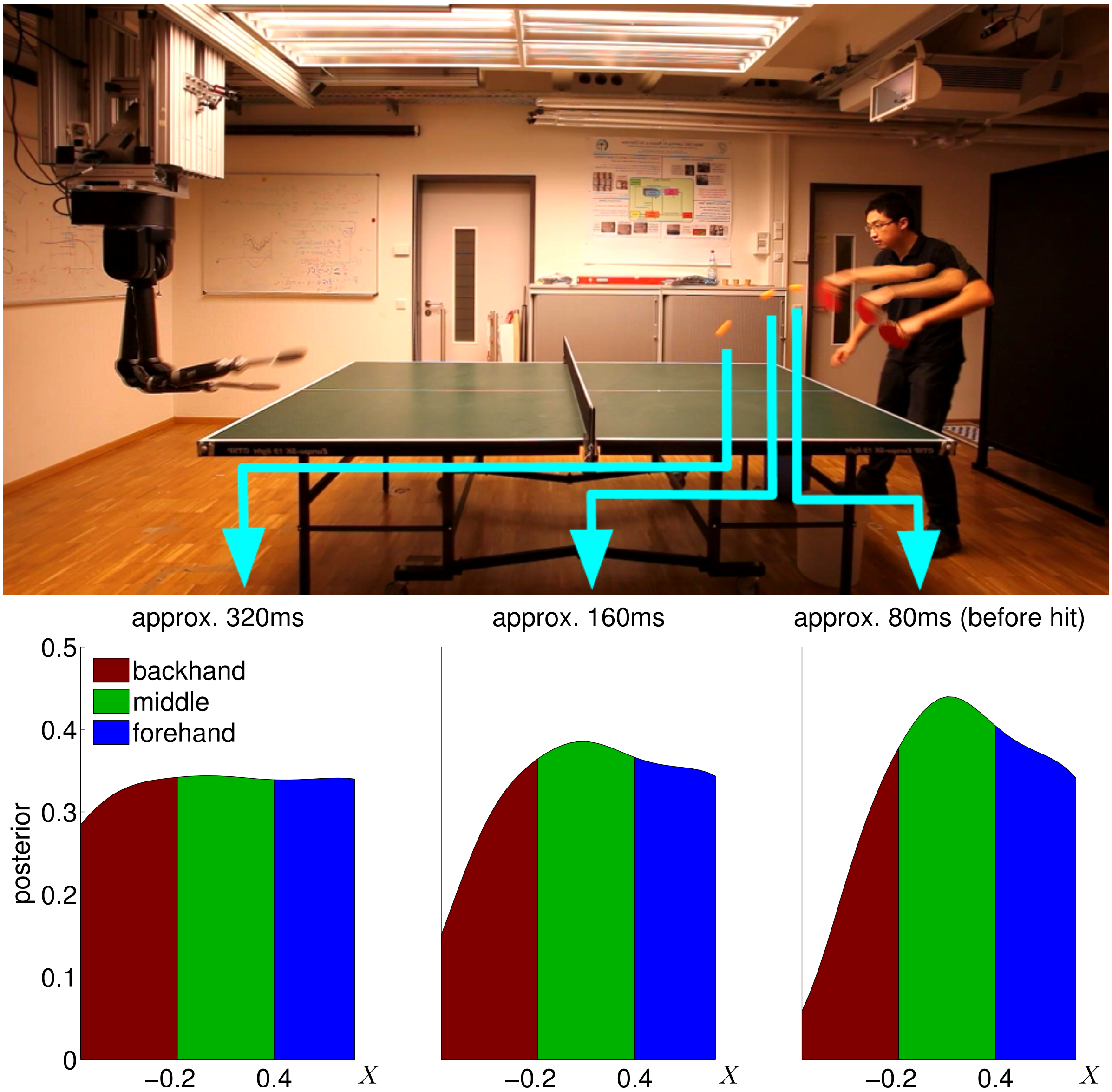


Figure 3.9: Bar plots show the distribution of the target (X coordinate) at approximately 320ms, 160ms, and 80ms before the player hits the ball. The prediction became increasingly confident as the player finishes the hitting movement, and the robot later chose the middle hitting movement accordingly.

the opponent has hit the ball himself, which is often necessary for the robot to have sufficient time to execute the hitting movement, and substantially expands the robot's overall hitting region to cover almost the entire accessible workspace, see the video. Furthermore, we expect that the method can further enhance the robot's capability when equipped with more and self-improving hitting primitives (Mülling et al., 2013).

3.6 Action Recognition in Human-Robot Interaction

To realize safe and meaningful human-robot interaction, it is important that robots can recognize the human's action. The advent of robust, marker-less motion capture techniques (Shotton et al., 2013) has provided us with the technology to record the full skeletal configuration of the

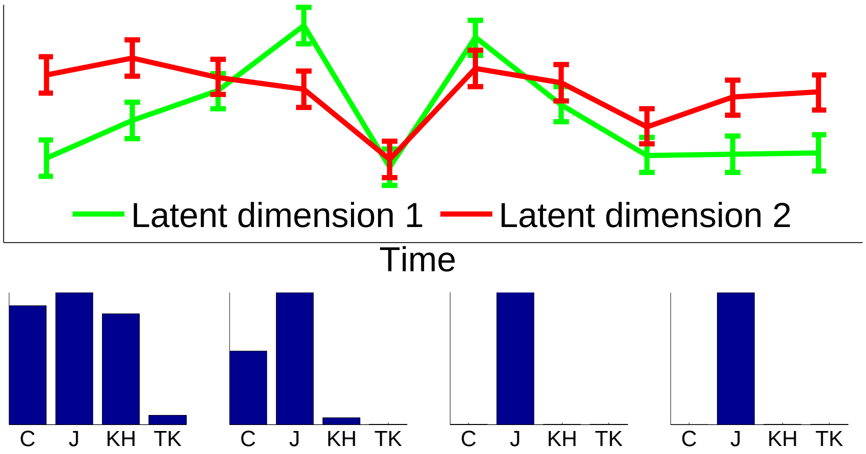


Figure 3.10: The trajectories of the 2D latent states for two consecutive Jumping actions obtained by the inference algorithm. The error bars represent the corresponding standard deviations. The bar charts correspond to the likelihood of Crouching, Jumping, Kick-High and Turn-Kick at different stages of an action.

human during HRI. Nevertheless, recognition of the human's action from this high-dimensional data set poses serious challenges.

In this section, we show that the IDDM has the potential to infer the intention of actions from movements in a simplified scenario. Using a Kinect camera, we recorded the 32-dimensional skeletal configuration of a human during the execution of a set of actions namely: crouching (C), jumping (J), kick-high (KH), kick-low (KL), defense (D), punch-high (PH), punch-low (PL), and turn-kick (TK). For each type of action we collected a training set consisting of ten repetitions and a test set of three repetitions. The system down-sampled the output of Kinect and processed three skeletal configurations per second.

In this task, the intention g is a discrete variable and corresponds to the type of action. Action recognition can be regarded as a classification problem. We compared our proposed algorithms to Support Vector Machines (SVMs), see (Schölkopf and Smola, 2001), and multi-class Gaussian Process Classification (GPC), see (Khan et al., 2012). We used off-the-shelf toolboxes, i.e., LIBSVM (Chang and Lin, 2011) and catLGM[7], and followed their standard routines for prediction.

The algorithms made a prediction after observing a new skeletal configuration. The batch algorithm used a sliding window of length $n = 5$, i.e., it recognized actions based on the recent n observations. We chose the IDDM with a linear covariance function for the covariance function k_z of the measurement GP and a two-dimensional latent state space. The batch algorithm achieved the precision of 83.8%, which outperformed SVM (77.5%) and GPC (79.4%) using the same sliding windows. The online algorithm achieved a precision of 83.0% with significantly reduced computation time. We observed that both the SVM and GPC confused crouching with jumping, as they were similar in the early and late stages. In contrast, the IDDM could distinguish between crouching (C) and jumping (J) from their different dynamics, as shown in Figure 3.10. which became clearly separable while the human performed the actions.

[7] `http://www.cs.ubc.ca/~emtiyaz/software/catLGM.html`

The batch algorithm needs to choose the size of sliding windows, which influences both the accuracy and efficiency. As shown in Table 3.3, the batch algorithm could yield real-time action recognition at a rate of 3 Hz with a sliding window of size 5. The online algorithm, as shown in Table 3.3, achieved a speedup of over four times compared to the batch algorithm with a sliding window. The online algorithm relies on the shrinking factor ϵ in Eq. (3.4.1), which describes how likely the type of actions is expected to change. We also found that the performance of the online algorithm is not sensitive to this parameter.

3.7 Conclusions of Chapter 3

In this chapter, we have discussed the intention-driven dynamics model (IDDM), a latent-variable model for inferring intentions from observed human movements. We have introduced efficient approximate inference algorithms that allow for real-time inference. We verified the proposed model in two human-robot interaction scenarios, namely, target inference in robot table tennis and action recognition for interactive robots. In these two scenarios, we showed that modeling the intention-driven dynamics can achieve better predictions than algorithms without modeling the dynamics.

The proposed method outperformed the GPR in the robot table tennis scenario and SVM and GPC in the action recognition scenario. Nevertheless, we would not draw the overstated conclusion that IDDM is a better model than SVM or GP based on these empirical results, as this discussion would be a comparison of generative and discriminative models. The performance of IDDM and SVM/GP should be studied on a case-by-case basis. However, two important properties of these approaches should be noticed: (1) computational efficiency and (2) robustness to measurement noise. and (3) curse of dimensionality. Firstly, the IDDM is often more computationally demanding than GP and SVM. Nevertheless, the proposed online inference method, and described possible approximations, make the IDDM applicable to real-time scenarios. As demonstrated in the prototype robot table tennis system, the IDDM was successfully used in a real system with tight time constraints. Secondly, the IDDM is generally less prone to measurement noise than SVM/GP, as it models the noise in the generative process of observations. Finally, the IDDM suffers less from the curse of dimensionality, and can thus better capture the model using a relatively small sample: The IDDM learns a measurement model ($\mathbb{R}^{D_x} \to \mathbb{R}^{D_z}$)

Table 3.3: Comparison of the accuracy and efficiency using different algorithms for the action recognition task. Here, n denotes the size of sliding windows and ϵ is the shrinking factor of the online method.

algorithm	accuracy	time(s)
SVM(n=5)	77.5%	<0.01
GPC(n=5)	79.4%	>1
batch(n=4)	79.0%	0.27
batch(n=5)	**83.8%**	**0.32**
batch(n=6)	83.0%	0.39
online(ϵ=0.3)	83.0%	0.07
online(ϵ=0.2)	**83.0%**	**0.07**
online(ϵ=0.1)	82.6%	0.07

and a transition model ($\mathbb{R}^{D_x} \rightarrow \mathbb{R}^{D_x}$). In comparison, the SVM/GP learns a mapping with a sliding window of size M to the intention ($\mathbb{R}^{TD_z} \rightarrow \mathbb{R}$), where the input space has a considerably higher dimensionality.

In conclusion, the IDDM takes into account the generative process of movements in which the intention is the driving factor. Hence, we advocate the use of IDDM when the movement is indeed driven by the intention (or target to predict), as the IDDM captures the causal relationship of the intention and the observed movements.

The IDDM offers the promise of anticipatory robots that incorporate anticipatory action selection in their action planning. In the next chapter, we focus on the decision making of anticipatory action selection, based on the outputs of the IDDM (the belief distribution of the intention).

3.A Vision System for the Robot Table Tennis

To track the opponent's racket, the vision system employs three Prosilica GE640C cameras mounted above the robot. Their position and direction are chosen so that the opponent can always be seen from every camera and the racket surface is fully visible from at least two cameras. These cameras are synchronized and calibrated to the coordinate system of the robot. Each camera outputs a stream of frames with frequency of 60Hz, ensuring the possibility of real-time racket tracking. We divide the tracking problem into localizing the racket in each camera and reconstructing its 3D configuration from camera pairs. These problems are both solved in parallel on a multi-core computer.

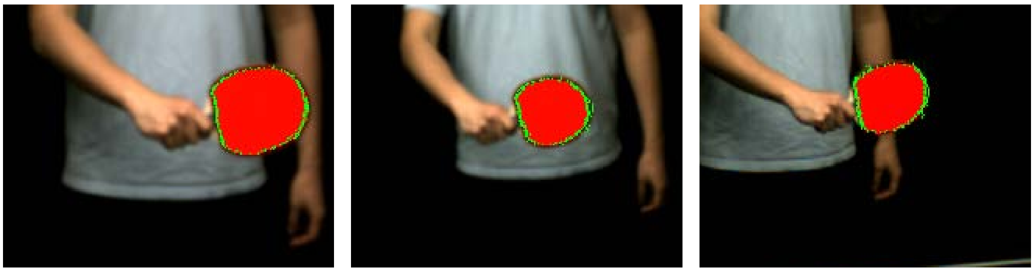
For each camera, we use linear-chain Condition Random Fields (CRF) presented by Sutton and McCallum (2007) for localizing the racket in each frame. For a frame $\mathbf{I}_t$ indexed by time step t, we compute the most likely racket configuration $\boldsymbol{\theta}_t$ represented by a 2D sliding window. We also include the shift of configurations in consecutive frames in the model as the speed of the racket is constrained by the physical motor limits of a human. The joint conditional probability of configurations given N frames is given by

$$P(\boldsymbol{\theta}_{1\ldots N}|\mathbf{I}_{1\ldots N}) = \frac{1}{Z(\mathbf{I}_{1\ldots N})} \exp\left\{\sum_{t=1}^{N} \boldsymbol{\alpha}^T \mathbf{f}(\boldsymbol{\theta}_t, \mathbf{I}_t) + \sum_{t=2}^{N} \boldsymbol{\beta}^T \mathbf{g}(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_t)\right\},$$

where $Z(\mathbf{I}_{1\ldots N})$ is the partition function, vector $\mathbf{g}(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_t)$ measures the differences of the image coordinates between two consecutive configurations, vector $\mathbf{f}(\boldsymbol{\theta}_t, \mathbf{I}_t)$ represents the features extracted from the sliding window, and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are corresponding parameters in the CRF model. The projection of the racket surface on a 2D image roughly resembles a solid ellipse without any texture inside. Many popular features for tracking, for example HOG (Dalal and Triggs, 2005) and SIFT (Lowe, 1999), cannot be employed in this case due to the lack of texture. By contrast, color-based features are strong indicators of the presence of the racket, and are additionally very efficient to compute. Hence, we use the local color histogram in the sliding window as the features $\mathbf{f}$, with the HSV space quantized into one hundred bins[8].

The parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ were learned by maximizing the likelihood of the labeled configurations, from the training data that consists of manually labeled sliding windows in one hundred

[8] We group all pixels in 1000 captured images into 100 clusters by k-means, and quantize the HSV space by nearest neighbor mapping.



(a) Camera 1 (b) Camera 2 (c) Camera 3

Figure 3.11: The reconstruction of racket surface. Pixels with a score higher than the threshold are highlighted by red color. Green dots are matched points from a pair of cameras.

frames. Subsequently, the tracking at time t is to find the configuration with the maximal marginal probability $P(\boldsymbol{\theta}_t|\mathbf{I}_{1\ldots t})$, which can be decomposed as

$$P(\boldsymbol{\theta}_t|\mathbf{I}_{1\ldots t}) \propto \exp\left(\boldsymbol{\alpha}^T \mathbf{f}(\boldsymbol{\theta}_t, \mathbf{I}_t)\right) \left\{ \sum_{\boldsymbol{\theta}_{t-1}} P(\boldsymbol{\theta}_{t-1}|\mathbf{I}_{1\ldots t-1}) \exp\left(\boldsymbol{\beta}^T \mathbf{g}(\boldsymbol{\theta}_{t-1}, \boldsymbol{\theta}_t)\right) \right\}.$$

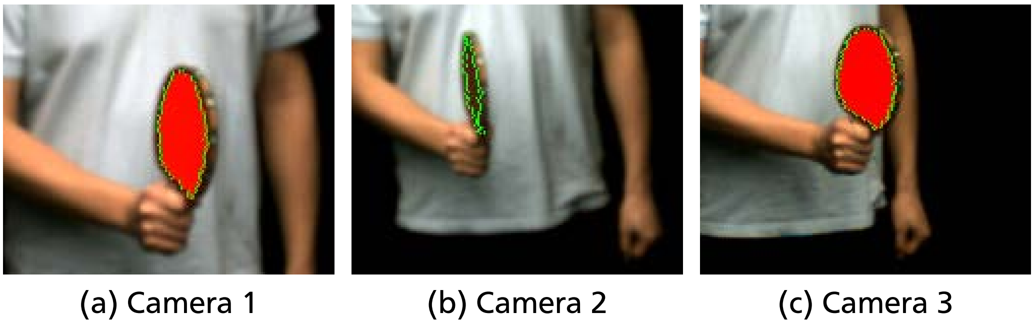
As the features are a histogram, we can obtain $\boldsymbol{\alpha}^T\mathbf{f}(\boldsymbol{\theta}_t, \mathbf{I}_t)$ efficiently for all $\boldsymbol{\theta}_t$ using 2D convolution. The second factor can be approximately estimated by only considering $\boldsymbol{\theta}_{t-1}$ whose distance from $\boldsymbol{\theta}_t$ is bounded by a constant, where 2D convolution is also applicable. As a result, we can efficiently compute the marginal probability $P(\boldsymbol{\theta}_t|\mathbf{I}_{1\ldots t})$.

We can assign a score to every color according to the parameters $\boldsymbol{\alpha}$. We highlight the shape of the racket as shown in Figure 3.11. The racket may not always be detected correctly, especially when its surface's normal vector is perpendicular to the camera direction as shown in Figure 3.11b. Nonetheless, the racket is still correctly localized in the other two cameras, and, hence, can be recovered in the 3D reconstruction stage, as shown in Figure 3.12.

We reconstruct the racket's 3D configuration from matched points for every camera pair where the racket is visible. The racket surface has no texture, rendering the matching of the key-points difficult. As the projection of the racket on an epipolar line (Hartley and Zisserman, 2003) forms a line segment, we match the left most points with a score higher than the threshold for pair of epipolar lines,, and the right most points as well. Those pairs of matched points are converted into a set of 3D points.

Although noise is inevitable, we expect that the majority of the matched points are on the surface plane of the racket. We apply RANSAC (Fischler and Bolles, 1981) to robustly estimate the normal vector of the racket surface, and concurrently eliminate outliers. The procedure can be performed in parallel. The green dots in Figure 3.12 correspond to all matched points, from which we can recover the surface of the racket. The surface is subsequently projected back into the 2D images. As shown in Figure 3.12b, we can detect and recover the incorrect racket configuration.

In summary, we developed a robust racket tracking system that detects the racket's position and orientation in real time. Together with a real-time ball tracking system (Lampert and Peters, 2012), the robot can perceive the state of the ball and the opponent's racket, the information needed for prediction and decision making. Note that the filtering method, such as Kalman



(a) Camera 1

(b) Camera 2

(c) Camera 3

Figure 3.12: Images from the cameras together with the reconstructed racket surface. We highlight the pixels on the racket surface whose scores are higher than the threshold by the red color. Green dots are matched points from a pair of cameras. Although the racket is not detected in the image of Camera 2 due to the perpendicular viewing angle, it can be recovered from the other cameras.

filtering, cannot be applied to the racket's trajectory, as its dynamics are directed by the unknown intended target. Therefore, the obtained states of the racket are still subject to substantial noise. The proposed inference method in this chapter can jointly smooth on the racket's trajectory and infer the target where the player intends to shoot.

4 Anticipatory Action Selection

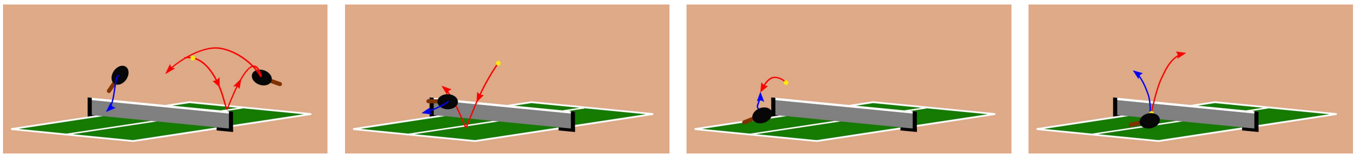
In this chapter, we address the problem of action selection and optimal timing for initiating the movement, provided a time series of predictions. We formulate the anticipatory action selection as a Partially Observable Markov Decision Process, where both transition model and observation model are modeled by the IDDM. We present two approaches to policy learning and decision making. Experimental results using a simulated environment showed that the proposed algorithms could substantially enhance the capability of the robot table tennis player. This chapter is based on (Wang et al., 2011b) and (Wang et al., submitteda).

4.1 Introduction

Humans possess the ability to coordinate their actions with others in joint activities, where *anticipation* plays an important role to integrate the prediction of the others' actions in one's own action planning (Sebanz et al., 2006). Such ability of anticipation is also crucial in Human-Robot Interaction (HRI), where an anticipatory robot needs to coordinate for its actions while interacting with humans (Hoffman and Breazeal, 2007), in both competitive and cooperative contexts. An anticipatory robot usually relies on a predictive model of the environment, especially its human partners, which "allows it to change state at an instant in accord with the model's predictions pertaining to a later instant" according to Rosen et al. (2012). Predicting the underlying intention and the future actions of the human partners has been extensively studied in robotics (e.g. Kurniawati et al., 2011; Hauser, 2012; Kuderer et al., 2012; Wang et al., 2013; Bandyopadhyay et al., 2013). This prediction capability offers the promise of anticipatory robots that incorporate *anticipatory action selection* in their planning.

In this chapter, we focus on the anticipatory action selection. Specifically, the robot chooses an action from a repertoire of motor skills based on the prediction of the human partner's intention. One important problem for the anticipatory action selection is prediction uncertainties that naturally arose due to the complexity and stochasticity of human behavior. The prediction usually becomes more accurate and less uncertain as more input, e.g., observed human movement, is obtained. However, waiting for a confident prediction causes delayed action selection and reduces the available time for the robot to execute its action (Wang et al., 2013). The anticipatory robot is, thus, forced to make decisions given a sequence of uncertain predictions, where a trade-off between prediction accuracy and reaction delay needs to be addressed. To address this issue, we formulate the decision making process as a Partially Observable Markov Decision Process (POMDP), and propose two different approaches to efficiently choose the robot's optimal action and decide the timing to initiate action.

The remainder of this chapter is organized as follows. We first illustrate the necessity of anticipation in robot table tennis in Section 4.1.1, and discuss the related work in Section 4.1.2. We formulate the anticipatory action selection using POMDPs, and present approaches to policy learning and decision making in Section 4.2. In Section 4.3, we first introduce the setup of the considered robot table tennis player. Subsequently, we evaluate the effectiveness of the proposed approaches, and show that they substantially enhance the capability of the considered robot player. Finally, we conclude and summarize the contribution in Section 4.4.



(a) awaiting stage (b) preparation stage (c) hitting stage (d) finishing stage

Figure 4.1: The four stages of a typical table tennis ball rally, where the trajectories in red represent the ball trajectories and the trajectories in blue represent the racket's movements of the robot player. Figure adapted from Mülling et al. (2011).

4.1.1 Anticipation in Human-Robot Table Tennis

We conduct a case study on human-robot table tennis, where anticipation is crucial for giving the robot sufficient time to execute its hitting movements (Wang et al., 2011b).

Playing table tennis is a challenging task for robots due to reasons ranging from the robot's deficiencies in perceiving the environment to the hardware limitations that restrict the actions. Hence, robot table tennis has been used as a benchmark task for high-speed vision (Acosta et al., 2003; Fässler et al., 1990), fast movement generation (Ángel et al., 2005; Mülling et al., 2011), learning (Matsushima et al., 2005; Miyazaki et al., 2005; Mülling et al., 2013), and many other research problems in robotics. Mülling et al. (2013) showed that a single type of hitting movement (action), e.g., forehand hitting movement, can be learned and constantly improved while interacting with human players. In practice, the robot needs a repertoire of actions, such as forehand and backhand hitting movements, to cover the potential hitting points in its entire accessible workspace. Despite that the robot is faster and more precise than a human being, the robot player still suffers from certain hardware limitations, such as torque and acceleration limits, which severely restrict its movement abilities in this high-speed scenario. Even a beginner human player would have the upper hand simply by choosing regions that the robot cannot reach in time if the robot's action is only based on the extrapolated trajectory of the incoming ball.

The robot's hitting movement imitates typical human table tennis strokes, as shown in Figure 4.1, which consist of four stages (Ramanantsoa and Durey, 1994). In the awaiting stage, the ball moves towards the human opponent and is returned back. The robot stays at the *awaiting pose* during this stage. The preparation stage starts when the coming ball passes over the net, and the robot moves to a *preparation pose*. The *hitting stage* begins once a hitting state is decided. The racket moves towards this hitting state and hits the ball at the end of the hitting stage. It follows through in the *finishing stage* and recovers to the awaiting pose. The duration of the hitting stage is constant for expert players and lasts approximately 80ms. Even against a slower opponent, which allows less than 300ms for the robot's hitting movement, this short time often does not suffice for the racket to reach the hitting state from the preparation position. Therefore, many desired hitting movements are not feasible due to time limits; the robot needs to initiates its hitting movement during the awaiting stage, that is, before the human opponent returns the ball.

To gain sufficient time for executing a hitting movement, the robot needs to be anticipatory about its human opponent's intention, which human players rely heavily on (Alexander and

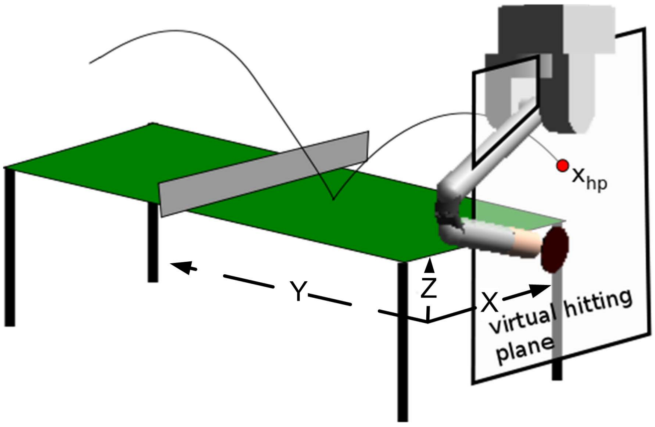


Figure 4.2: Demonstration of the robot's hitting plane and the human opponent's target. The hitting point is the intersubsection x_{hp} of the coming ball's trajectory and the virtual hitting plane 80 cm behind the table, which is considered the *target* where the human opponent intents to shoot the ball. Our goal is to select an action according to the prediction of the target x_{hp}. Figure adapted from Mülling et al. (2011).

Honish, 2009). The prediction of the human partner's intention can be realized by modeling how the intention directs the dynamics of hitting movement (Wang et al., 2013). In the table tennis scenario, this Intention-Driven Dynamics Model (IDDM) leads to an online algorithm that, given a time series of observations, continually predicts the human player's intended *target*, i.e., where the human intents to shoot the ball (Wang et al., 2013), as shown in Figure 4.2.

Anticipatory action selection needs to take into account the prediction uncertainty. The robot is likely to fail to return the ball if it has initiated a forehand ("right" in Figure 4.2) hitting movement and the ball is shot to its backhand ("left") region, or vice versa. The prediction of the intended target tends to become increasingly accurate and confident as the human opponent finishes his movement (Wang et al., 2013). On the other hand, the robot requires a certain minimum time to execute its hitting movement. Therefore, the essence of the anticipatory action selection is deciding when and how to initiate the hitting movement based on the increasingly confident predictions.

4.1.2 Related Work

Intention inference has been investigated in different settings, for example, using Hidden Markov Models (HMMs) to model and predict human behavior where different dynamics models were adopted to the corresponding behaviors (Pentland and Liu, 1999). Online learning of intentional motion patterns and prediction of intentions based on HMMs was proposed by Vasquez et al. (2008), which allows efficient inference in real time. The HMM can be learned incrementally to cope with new motion patterns in parallel with prediction (Vasquez et al., 2009).

Anticipation is important in many human-robot interaction scenarios (e.g., Ziebart et al., 2009; Dragan and Srinivasa, 2012; Wang et al., 2012b). Decision making can also be achieved jointly with intention inference, in scenarios such as autonomous driving (Bandyopadhyay et al., 2013), control (Hauser, 2012), or navigation in human crowds (Kuderer et al., 2012). For example, when the state space is finite, the problem can be formulated as a partially observable Markov decision process (Kurniawati et al., 2011) and solved efficiently (Wang et al., 2012a).

The anticipatory action selection decides when it is time to initiate the hitting movement. This decision is based on a trade-off between prediction accuracy and reaction delay, which is a generalization of optimal stopping problems. The optimal stopping problems have been extensively investigated in sequential analysis using Markov decision process (MDP), where the focus is usually the existence of optimal stopping rules given the transition model or obtaining closed-form solutions in specific problems (e.g., Shiryaev, 2007). We do not have closed-form solutions for the decision making problem in our application due to the complexity of the human dynamics. The optimal stopping problem can be applied in the context of classification and feature selection (Póczos et al., 2009; Gaudel and Sebag, 2010; Dulac-Arnold et al., 2012), using reinforcement learning to obtain an optimal stopping policy. The optimal stopping problem has also been studied under partial observation (Zhou, 2013), and in many applications, such as quality control (Jensen and Hsu, 1993) and finance (Décamps et al., 2005; Rishel and Helmes, 2006).

4.2 Anticipatory Action Selection

The essence of the anticipatory action selection is decision making given a time series of predictions, where two fundamental issues have to be addressed. First, the uncertainties in the prediction and the outcome need to be considered. For example in the robot table tennis setup, the uncertainty in the prediction is mainly due to the fact that the opponent may still change the target before the racket hits the ball. The prediction of the opponent's intended target, based on the observed partial movement of his stroke movement, tends to become increasingly accurate as the opponent finishes the hitting movement. Furthermore, the outcome of executing a selected action, e.g., the robot's success of returning the ball to the opponent's court with the chosen hitting movement, is not deterministic as the underlying dynamics of the robot arm are often too complicated to be modeled precisely at high speed. The decision making algorithm should be able to deal with the associated uncertainties.

The second fundamental issue is the timing for the robot to initiate the selected action, as the robot often requires sufficient time to execute an action. In the table tennis setup, while the predictions tend to become increasingly accurate, the robot needs sufficient time to move the arm from the awaiting pose to the desired preparation pose. The anticipatory action selection needs to trade off between prediction accuracy and delay in action selection, as both influence the success probability of the selected action. Hence, it is essential to choose the optimal action at the right time.

Modeling the problem of anticipatory action selection as a Partially Observable Markov Decision Process (POMDP), we present two different approaches to decision making. In the first approach, we transform the POMDP into an equivalent fully observable Markov Decision Process (MDP). The states in the equivalent MDP are the belief states of the POMDP, which are the posterior distribution of the unobserved state given the observation history. We adopt the Intention-Driven Dynamics Model (IDDM) for belief updates and the Least-Square Policy Iteration (LSPI) for policy learning. Howver, the model-free LSPI algorithm does not make use of the estimated transition model in the IDDM framework, and can be sample-insufficient in the considered application. Consequently, we present a more sample-efficient approach using the Monte-Carlo Planning (MCP), where actions are chosen according to the value function estimated using the Monte-Carlo method (Thrun, 2000).

4.2.1 Optimal Stopping under Partial Observability

The anticipatory action selection is a generalizaion of optimal stopping under partial observation (Mazziotto, 1986), and, similarly, can be formulated as a POMDP. A discrete-time POMDP is defined as a tuple $(\mathscr{S}, \mathscr{A}, \mathscr{Z}, \mathscr{P}, \Omega, \mathscr{R})$, where $\mathscr{S}$ is a state space, $\mathscr{A} = \{0, \ldots, n\}$ is a set of actions, and $\mathscr{Z}$ is a observation space. The states evolve following a Markov transition model, governed by $\mathscr{P}$, where $\mathscr{P}(\mathbf{s}'|\mathbf{s}, a)$ represents the probability of going to state $\mathbf{s}'$ from state $\mathbf{s}$ when taking action a. The observations are generated from the states following the observation model Ω, where $\Omega(\mathbf{z}|\mathbf{s})$ is the probability of observing $\mathbf{z}$ in state $\mathbf{s}$. The reward function $\mathscr{R}(\mathbf{s}, a)$ represents the expected immediate reward obtained for taking action a in state $\mathbf{s}$.

The set of actions $\mathscr{A}$ consists of a waiting action $a = 0$ and a set of stopping actions $a \in \mathscr{A} \setminus \{0\}$. In the table tennis scenario, taking the waiting action $a = 0$ means to postpone the selection of a hitting action and to wait for the subsequent observation to be available; and each stopping action $a \neq 0$ leads to selecting and initiating a particular type of hitting movement, and, hence, to the termination of an episode. The immediate reward is only nonzero when a stopping action is taken, and corresponds to the outcome of the selected type of hitting movement. We consider a continuous state representation $\mathbf{s} = [\mathbf{x}, g]$ that consists of the state $\mathbf{x}$ of the environment and the intention g of the human. Here, the intention g is assumed to be invariant during an episode; for example, the intended target does not change during the human player's hitting movement. An observation $\mathbf{z} \in \mathscr{Z}$ includes perceived features of the environment, such as the position and velocity of the ball and the configuration of the opponent's racket.

Finding the optimal timing to initiate the appropriate action is a POMDP problem. The decision at every time step is made by maximizing the expected future reward that it leads to, e.g., the chance of successfully returning the ball in the table tennis setup. Specifically, we want to maximize the aggregated expected reward during an episode, given by

$$J = \mathbb{E}\left[r(\mathbf{s}_T, a_T)\right],$$

where the variable T denotes the stopping time, and a reward is obtained only by taking a stopping action a_T.

4.2.2 Belief Update with Intention-Driven Dynamics Model

A key step towards solving the optimal stopping problem is updating the belief on state $\mathbf{s}_t = [\mathbf{x}_t, g]$ according to the history $\mathbf{z}_{1:t}$, given by $p(\mathbf{x}_t, g|\mathbf{z}_{1:t})$.

We apply the online target prediction algorithm using the Intention-Driven Dynamics Model (IDDM) (Wang et al., 2012b, 2013). The IDDM is a discrete-time dynamics model for movement modeling and intention inference. In robotics scenarios, we often rely on noisy and high-dimensional sensor data. However, the intrinsic states are typically not observable, and may have lower dimensions. Therefore, we seek a latent state representation of the relevant information in the data, and then model how the intention governs the dynamics in the latent states $\mathbf{x}_t$, as shown in Figure 4.3. The resulting model jointly learns both the latent state representation and the dynamics in the state space.

Designing a parametric dynamics model is difficult due to the complexity of nonlinear and stochastic human movements. Hence, the IDDM uses Gaussian processes to handle both the

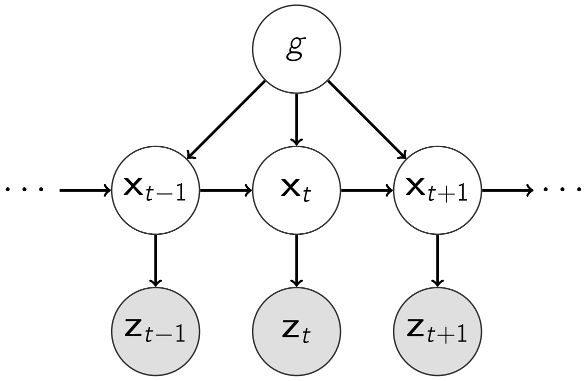


Figure 4.3: The graphical model of the IDDM in an online manner, where we denote the intended target by g, state by $\mathbf{x}_t$, and observation by $\mathbf{z}_t$. The proposed model explicitly incorporates the intention as an input to the transition function (Wang et al., 2013). Here, we use gray nodes for the observed variables and white nodes for the latent variables.

transition model $p(\mathbf{x}_{t+1}|\mathbf{x}_t, g)$ in the latent state space and the observation model $p(\mathbf{z}_t|\mathbf{x}_t)$ from the latent states to the observations. The IDDM considers the dynamics of latent states $\mathbf{x}$ to follow an unknown function $\mathbf{f}$, given by

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, g) + \mathbf{n}_{x,t}, \quad \mathbf{n}_{x,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_x).$$

The latent state $\mathbf{x}_{t+1}$ at time $t+1$ depends on the latent state $\mathbf{x}_t$ at time t as well as on the intention g, as demonstrated in the graphical model shown in Figure 4.3. A GP prior $\mathcal{GP}(m_x(\cdot), k_x(\cdot,\cdot))$ is placed on every dimension of $\mathbf{f}$ with shared mean and covariance functions. Subsequently, the predictive distribution of the latent state $\mathbf{x}_{t+1}$ conditioned on the current state $\mathbf{x}_t$ and intention g is a Gaussian distribution given by $\mathbf{x}_{t+1} \sim \mathcal{N}(\mathbf{m}_x([\mathbf{x}_t, g]), \boldsymbol{\Sigma}_x([\mathbf{x}_t, g]))$ based on training inputs $\mathbf{X}_x$ and outputs $\mathbf{Y}_x$. Similarly, the measurement mapping function $\mathbf{h}$ from latent state $\mathbf{x}$ to observations $\mathbf{z}$, given by

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{n}_{z,t}, \quad \mathbf{n}_{z,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_z),$$

is modeled by another set of GPs. The predictive probability of the observations $\mathbf{z}_t$ is given by a Gaussian distribution $\mathbf{z}_t \sim \mathcal{N}(\mathbf{m}_z(\mathbf{x}_t), \boldsymbol{\Sigma}_z(\mathbf{x}_t))$, where the predictive mean and covariance are computed based on training inputs $\mathbf{X}_z$ and outputs $\mathbf{Y}_z$. For more details of the IDDM, we refer the reader to Wang et al. (2013).

Assuming that the dynamics of the human player's racket is driven by the intended target g, we can apply the IDDM to predict the target g given a time series of observations $\mathbf{z}_{1:t}$ that are generated from corresponding latent states $\mathbf{x}_{1:t}$. While exact inference of the intention g and states $\mathbf{x}_t$ is not tractable, Wang et al. (2013) presented an efficient online inference algorithm to update the belief $p(g, \mathbf{x}_t|\mathbf{z}_{1:t})$, i.e., the posterior probability of the intention g and the latent states $\mathbf{x}_t$ once a new observation $\mathbf{z}_t$ is obtained. Figure 3.9 showed that the predictive uncertainties decrease as the human player finishes the hitting movement. To summarize, the IDDM provides an estimate of the transition model $\mathcal{T}$ and of the measurement model Ω in the considered POMDP with Gaussian processes, which are used for updating the belief.

4.2.3 Policy Learning with Belief MDP

A POMDP can be transformed to an equivalent fully observable MDP by using the belief state $\boldsymbol{\theta}_t \in \Theta$ as the observable state, which is the belief of the unobserved state $\mathbf{s}_t = [\mathbf{x}_t, g]$ given the observation history $\mathbf{z}_{1:t}$. Subsequently, we can apply reinforcement learning algorithms to learn and improve a policy π, mapping a belief state $\boldsymbol{\theta}_t$ to the action that maximizes the expected reward. Estimating the transition model $\mathscr{P}(\boldsymbol{\theta}'|\boldsymbol{\theta}, a)$ is difficult mainly due to the complicated behavior of the nonparametric dynamics model employed. Hence, we need model-free reinforcement learning algorithms that do not rely on an estimate of the transition model for belief states. Specifically, we consider the Q-learning algorithm (Sutton and Barto, 1998), and learn a state-action value function $Q^\pi(\boldsymbol{\theta}, a)$ of a policy π. The value function $Q^\pi(\boldsymbol{\theta}, a)$ measures the expected future reward when taking action a according to the current belief $\boldsymbol{\theta}$ and following the policy π thereafter. We can write the value function according to the Bellman equation, given by

$$Q^\pi(\boldsymbol{\theta}, a = 0) = \int \mathscr{P}(\boldsymbol{\theta}'|\boldsymbol{\theta}, a = 0) \int \pi(a'|\boldsymbol{\theta}') Q^\pi(\boldsymbol{\theta}', a') da' d\boldsymbol{\theta}', \tag{36}$$

for the waiting action $a = 0$, and

$$\forall a \neq 0, \qquad Q^\pi(\boldsymbol{\theta}, a) = \mathscr{R}(\boldsymbol{\theta}, a),$$

for any stopping action.

Since the belief states are continuous, the value function Q^π cannot be written in a tabular form. A common way to deal with large or infinite state space is value function approximation by a linear combination of basis functions $\boldsymbol{\phi}(\boldsymbol{\theta}, a)$, given by

$$Q^\pi(\boldsymbol{\theta}, a) \approx \hat{Q}^\pi(\boldsymbol{\theta}, a; \mathbf{w}) = \boldsymbol{\phi}(\boldsymbol{\theta}, a)^T \mathbf{w},$$

with parameters $\mathbf{w}$. Taking into account these factors, we choose the Least-Square Policy Iteration (LSPI) algorithm (Lagoudakis and Parr, 2003), a model-free reinforcement learning algorithm with the function approximation. The LSPI has been used successfully to solve several large scale problems.

The LSPI algorithm consists of a policy evaluation step, which estimates the value function $\hat{Q}^\pi$ for the current policy π, and a policy improvement step, which improves the policy π by fixing the obtained value function $\hat{Q}^\pi$. Using the linear function approximation, the policy evaluation step boils down to estimating the parameters $\mathbf{w}$. The integration in Eq. (36) is intractable due to the unknown transition model $\mathscr{P}$, and is replaced by sampling in the Q-learning framework. Derived from MDP with finite state space, the update rule of the approximation parameters $\mathbf{w}$ can be straightforwardly applied for the considered continuous state space. The approximation parameters $\mathbf{w}$ can be obtained from a finite set of samples $\mathscr{D} = \{(\boldsymbol{\theta}_i, a_i, r_i, \boldsymbol{\theta}'_i)|i = 1, \ldots, L\}$, where $\boldsymbol{\theta}_i$ is the output of the online inference algorithm (Wang et al., 2013). Given a policy π, the estimates

$$\hat{\mathbf{A}} = \frac{1}{L} \sum_{l=1}^{L} \boldsymbol{\phi}(\boldsymbol{\theta}_i, a_i)(\boldsymbol{\phi}(\boldsymbol{\theta}_i, a_i) - \boldsymbol{\phi}(\boldsymbol{\theta}'_i, \pi(\boldsymbol{\theta}'_i)))^T,$$

$$\hat{\mathbf{b}} = \frac{1}{L} \sum_{l=1}^{L} \boldsymbol{\phi}(\boldsymbol{\theta}_i, a_i) r_i,$$

Algorithm 4.1: The LSPI algorithm, which iteratively updates the approximated value function and the optimal policy.

```
Input  : Obtained samples $\mathscr{D}$
Output: Approximation parameters $\mathbf{w}$
$\mathbf{w}' \leftarrow 0$;
repeat
    $\mathbf{w} \leftarrow \mathbf{w}'$;
    foreach $(\boldsymbol{\theta}_i, a_i, r_i, \boldsymbol{\theta}'_i) \in \mathscr{D}$ do
        $a' \leftarrow \pi(\boldsymbol{\theta}'_i) = \arg\max_a \boldsymbol{\phi}(\boldsymbol{\theta}'_i, a)^T \mathbf{w}$;
        $\hat{\mathbf{A}} \leftarrow \hat{\mathbf{A}} + \frac{1}{L}\boldsymbol{\phi}(\boldsymbol{\theta}_i, a_i)(\boldsymbol{\phi}(\boldsymbol{\theta}_i, a_i) - \boldsymbol{\phi}(\boldsymbol{\theta}'_i, a'))^T$;
        $\hat{\mathbf{b}} \leftarrow \hat{\mathbf{b}} + \frac{1}{L}\boldsymbol{\phi}(\boldsymbol{\theta}_i, a_i) r$;
    $\mathbf{w}' \leftarrow (\hat{\mathbf{A}} + \delta \mathbf{I})^{-1}\hat{\mathbf{b}}$;
until convergence ;
```

are used to update the approximation parameters

$$\mathbf{w} = (\hat{\mathbf{A}} + \delta^2 \mathbf{I})^{-1}\hat{\mathbf{b}},$$

where the sufficient small δ^2 is used to avoid numerical error in the inversion of $\hat{\mathbf{A}}$ (Lagoudakis and Parr, 2003).

In the policy improvement step, we improve the policy π by a new policy π' that maximizes the expected reward according to the estimated value function $\hat{Q}^\pi$. The optimal policy π' greedily chooses the action that maximizes the corresponding value function $\hat{Q}^\pi$. Therefore, we obtain an improved policy

$$\pi'(\boldsymbol{\theta}) = \arg\max_a \hat{Q}^\pi(\boldsymbol{\theta}, a).$$

We can obtain the optimal policy by iteratively executing the policy evaluation and improvement steps, as summarized in Algorithm 4.1.

4.2.4 Monte-Carlo Planning with POMDP

The LSPI algorithm as described above employs a model-free approach to policy learning, using the Intention-Driven Dynamics Model as a black box for updating the belief $\boldsymbol{\theta}_t$ given a history of observations $\mathbf{z}_{1:t}$. Besides the capability of the belief update, the IDDM in fact provides a transition model in the state space, estimated from its training data, which has not been exploited by the LSPI algorithm. Here, we present Monte-Carlo Planning (MCP), a model-based approach to action selection as an alternative to the LSPI algorithm.

Rather than estimating the value function Q^π given a policy π, we directly consider the value function Q for the optimal policy. As the stopping actions terminate the decision process immediately, the value function for the stopping actions $\forall a \in \mathscr{A} \setminus \{0\} : Q^\pi(\boldsymbol{\theta}, a) = Q(\boldsymbol{\theta}, a)$ holds for any belief state $\boldsymbol{\theta}$. We can reuse the same value function for those stopping actions estimated by LSPI. The key difference is in the value function for the waiting action $Q(\boldsymbol{\theta}_t, a_t = 0)$, given by the expected future reward

$$Q(\boldsymbol{\theta}_t, a_t = 0) = \mathbb{E}\big[\max_{a_{t+1}} Q(\boldsymbol{\theta}_{t+1}, a_{t+1})\big]$$

Algorithm 4.2: The particle projection algorithm that estimates the value function of postponing the decision for one time step.

```
Input  : Current belief $\boldsymbol{\theta}_t$
Input  : Number of samples $I$
Output: Estimate of value function $Q(\boldsymbol{\theta}_t, a_t = 0)$
Collection of sampled rewards $\Phi = \emptyset$ ;
for $i \leftarrow 1, \ldots, I$ do
    Sample current state and intention $\mathbf{s}_t = [\mathbf{x}_t, g]$ according to belief $\boldsymbol{\theta}_t$ ;
    Sample subsequent state $\mathbf{x}_{t+1} \sim P(\mathbf{x}_{t+1}|\mathbf{x}_t, g)$ using transition model of IDDM ;
    Sample subsequent observation $\mathbf{z}_{t+1} \sim P(\mathbf{z}_{t+1}|\mathbf{x}_{t+1})$ using measurement model of IDDM
    ;
    Update belief $\boldsymbol{\theta}_{t+1}$ provided observation $\mathbf{z}_{t+1}$ using IDDM ;
    Compute maximal expected reward for stopping $r^i = \max_{a \neq 0} \hat{Q}(\boldsymbol{\theta}_{t+1}, a; \mathbf{w})$ ;
    Update collection of sampled rewards $\Phi = \Phi \cup \{r^i\}$ ;
Return $Q(\boldsymbol{\theta}_t, a_t = 0) \approx \frac{1}{I} \sum_{r^i \in \Phi} r^i$ ;
```

with respect to the subsequent belief state $\boldsymbol{\theta}_{t+1}$.

The value function $Q(\boldsymbol{\theta}_t, a_t = 0)$ measures the expected reward of waiting for more observation. While computing the exact expectation is intractable, the value function can be estimated using Monte-Carlo approximation (Thrun, 2000), where we replace the expectation operator by an empirical average over sampled belief states. To estimate the value function, each time we draw a sample of the current state $\mathbf{s}_t$, the subsequent state $\mathbf{s}_{t+1}$, and the subsequent observation $\mathbf{z}_{t+1}$, compute the subsequent belief state $\boldsymbol{\theta}_{t+1}$ based on the IDDM, and estimate $\max_{a_{t+1}} Q(\boldsymbol{\theta}_{t+1}, a_{t+1})$ recursively. One can see that estimating the value function for waiting at the next time step $Q(\boldsymbol{\theta}_{t+1}, a_{t+1} = 0)$ again relies on the Monte-Carlo approximation, and, hence, that the number of sampled decision trees grows exponentially with the horizon. Although the horizon is often finite for the anticipatory action selection, e.g., the optimal hitting action can be chosen straightforwardly once the human player has hit the ball, we need to limit the depth of sampled decision trees in consideration of restrictive time constraints in robotics. Here, we only plan for one step ahead; namely, we estimate the value function of postponing the decision for one time step, given by

$$Q(\boldsymbol{\theta}_t, a_t = 0) = \mathbb{E}\big[\max_{a_{t+1} \neq 0} \hat{Q}(\boldsymbol{\theta}_{t+1}, a_{t+1}; \mathbf{w})\big].$$

This estimation can be achieved by using particle projection routine (Thrun, 2000), as described in Algorithm 4.2.

The particle projection may still be too time-consuming to be applicable to online planning, as the Monte-Carlo approximation requires a certain amount of samples to achieve a reliable estimate. Nevertheless, consider the online planning that finds the optimal action

$$a_t = \operatorname*{argmax}_{a \in \mathscr{A}} Q(\boldsymbol{\theta}_t, a),$$

where one only needs to know if the waiting action $a = 0$ leads to higher expected reward than all the stopping actions, rather than the actual expected reward of waiting. We can terminate the

Algorithm 4.3: The MCP algorithm with early termination according to the estimate of confidence interval.

```
Input : Current belief $\boldsymbol{\theta}_t$
Input : Number of samples $I$
Input : Function approximation parameters $\mathbf{w}$
Input : Confidence level $\alpha$
Output: Action $a_t$
Collection of sampled rewards $\Phi = \emptyset$ ;
for $i \leftarrow 1, \dots, I$ do
    Sample current state and intention $\mathbf{s}_t = [\mathbf{x}_t, g]$ ;
    Sample subsequent state $\mathbf{x}_{t+1} \sim P(\mathbf{x}_{t+1} | \mathbf{x}_t, g)$ ;
    Sample subsequent observation $\mathbf{z}_{t+1} \sim P(\mathbf{z}_{t+1} | \mathbf{x}_{t+1})$ ;
    Update belief $\boldsymbol{\theta}_{t+1}$ provided observation $\mathbf{z}_{t+1}$ ;
    Compute expected reward $r^i = \max_{a \neq 0} \hat{Q}(\boldsymbol{\theta}_{t+1}, a; \mathbf{w})$ ;
    Update collection of sampled rewards $\Phi = \Phi \cup \{r^i\}$ ;
    if Number of samples $|\Phi|$ sufficiently large then
        Compute upper confidence bound $U$ given sample $\Phi$;
        if $U < \max_{a \neq 0} Q(\boldsymbol{\theta}_t, a)$ then
            Return the optimal stopping action $a_t = \text{argmax}_{a \neq 0} Q(\boldsymbol{\theta}_t, a)$ ;
        Compute lower confidence bound $L$ given sample $\Phi$ ;
        if $L > \max_{a \neq 0} Q(\boldsymbol{\theta}_t, a)$ then
            Return the waiting action $a_t = 0$ ;
    Expected reward for waiting $Q(\boldsymbol{\theta}_t, a_t = 0)$ is the mean of sampled rewards in $\Phi$ ;
    Return the optimal action $a_t = \text{argmax}_a Q(\boldsymbol{\theta}_t, a)$ ;
```

particle projection routine if the expected reward of waiting is very likely to be higher or lower than that of the optimal stopping action $\max_{a \neq 0} Q(\boldsymbol{\theta}_t, a)$. Inspired by the *upper confidence bound* algorithms (Auer, 2003), we use the confidence interval estimate of the expected reward for waiting to terminate the particle projection routine before the Monte-Carlo sampling completes. To obtain a confidence interval estimator, we assume that the future reward for waiting at time step t is Gaussian distributed. Given a set of sampled rewards $\Phi = \{r^1, \dots, r^n\}$, the confidence interval given a confidence level α is

$$\left[\bar{r} - \frac{cs}{\sqrt{n}}, \bar{r} + \frac{cs}{\sqrt{n}}\right],$$

where $n = |\Phi|$ is the number of samples, $\bar{r}$ the sample mean, s the sample standard deviation, and c is the α percentile of a Student's t-distribution with $n-1$ degrees of freedom. Algorithm 4.3 describes the resulting Monte-Carlo planning algorithm.

In comparison to the model-free LSPI method, the MCP algorithm exploits the estimated transition model in the IDDM, and is expected to be more sample-efficient.

4.2.5 Basis Functions

The presented LSPI and MCP methods both employ function approximation, replying on a set of basis functions of the belief state $\boldsymbol{\theta}$. The belief state obtained by the IDDM is represented by a vector that consists of the mean and covariance of the belief on the latent state $\mathbf{x}$ and a discretized histogram over the intended target g. The discretized histogram is illustrated in Figure 3.9.

We consider a set of radial basis functions for approximating the value function. We collected all the encountered belief states on the training data, and chose K centers $\bar{\boldsymbol{\theta}}_1, \ldots, \bar{\boldsymbol{\theta}}_K$ using K-means clustering. The basis functions are given by

$$\boldsymbol{\phi}(\boldsymbol{\theta}, a) = [\delta_{a,0}\boldsymbol{\phi}'(\boldsymbol{\theta})^T, \delta_{a,1}\boldsymbol{\phi}'(\boldsymbol{\theta})^T, \ldots, \delta_{a,|\mathscr{A}|}\boldsymbol{\phi}'(\boldsymbol{\theta})^T]^T,$$

where δ is the Kronecker delta and we consider the radial basis functions

$$\boldsymbol{\phi}'(\boldsymbol{\theta}) = [\exp\{-\eta||\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_1||^2\}, \ldots, \exp\{-\eta||\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_K||^2\}]^T \tag{37}$$

for the belief states.

4.3 Application in Human-Robot Table Tennis

We evaluated the LSPI and the MCP algorithms for anticipatory action selection in human-robot table tennis setup. The presented action selection methods were evaluated on the biomimetic robot table tennis player (Mülling et al., 2011), as this setup allowed exhibiting how much of an advantage such an anticipation system may offer. We expect that the system will help similarly when deployed within the skill learning framework (Mülling et al., 2013) as well as many of the recent table tennis learning systems (Huang et al., 2013; Yang et al., 2010; Matsushima et al., 2005).

4.3.1 Robot Player

To quantitatively evaluate the performance of the proposed methods in terms of success rates, we used the SL framework (Schaal, 2009), which consisted of a real-world setup and a sufficiently realistic simulation. The setup included a Barrett WAM$^{\text{TM}}$ arm with seven degrees of freedom, which is capable of high speed motion. A racket was attached to the end-effector. Table, racket and ball were compliant with the international rules of table tennis. We used a vision system of seven cameras to collect real data during a table tennis game between two human players (Wang et al., 2011b), recording the players' movement and ball's trajectory. The collected data was used by the SL simulation for the following experiments, as demonstrated in Figure 4.4. Previous work showed that the SL framework is sufficiently realistic to simulate the robot table tennis setting[9], and that similar performance can be expected when the real robot is used (Mülling et al., 2011; Mülling et al., 2013; Wang et al., 2013).

In the considered setting, the robot always hits the ball on a virtual hitting plane 80 cm behind the table, as shown in Figure 4.2. We defined the human's intended target as the intersubsection

[9] See `http://robot-learning.de/Research/ProbabilisticMovementModeling` for a demonstration of the real-robot setup.

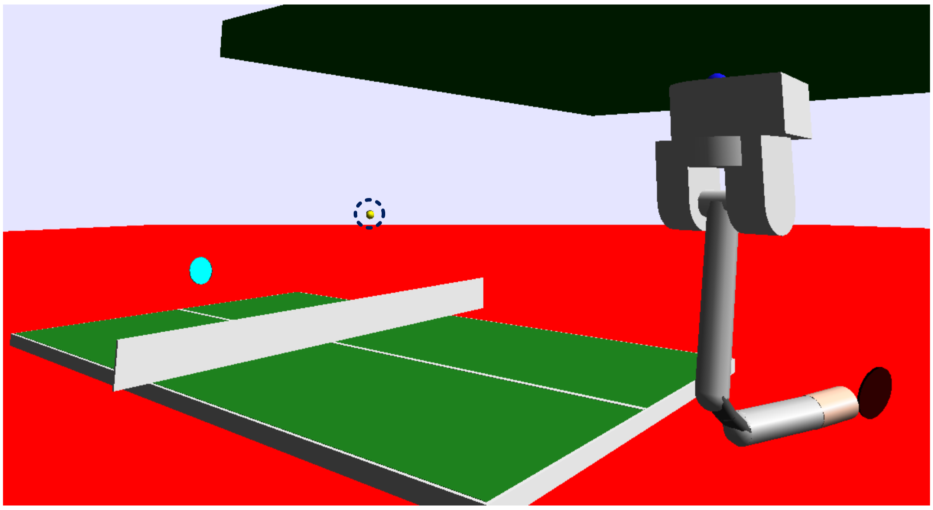
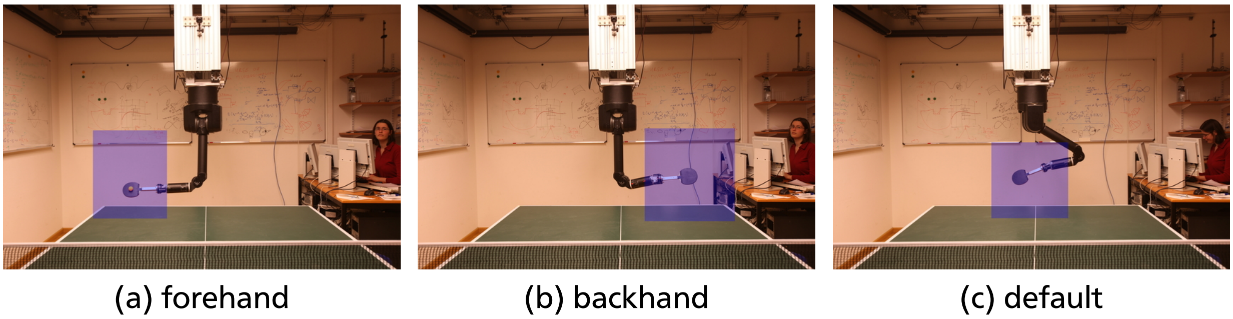


Figure 4.4: The SL simulated environment with the state of the robot arm and the information obtained from the vision system, including states of the opponent's racket the ball.



(a) forehand

(b) backhand

(c) default

Figure 4.5: Three types of hitting movement (actions) of the robot table tennis player. Each action was optimized for hitting points in a specific region as shown by the shaded rectangle.

of the returned ball's trajectory with the robot's virtual hitting plane. As the x-coordinate (see Figure 4.2) was most important for choosing the type of hitting movements (Wang et al., 2013), the intention g considered here was the x-coordinate of the hitting point. Physical limitations of the robot restricted the x-coordinate to the range of $\pm 1.2\,\mathrm{m}$ from the robot's base (the table is $1.52\,\mathrm{m}$ wide).

The robot player can execute three types of hitting movement (actions) that were refined and optimized for hitting points in a specific region, as shown in Figure 4.5. Hence, the action set $\mathscr{A}$ consisted of one waiting action and three stopping actions, each stopping action corresponding to a type hitting movement. Note that the action selection was only used to a hitting type, e.g., default, forehand, or backhand. Fine-tuning of the robot's movement can be done when the robot has initiated an action and once the returned ball can be reliably predicted from the ball's trajectory alone. However, returning the ball outside the corresponding hitting region is difficult once the robot has initiated the chosen action (Wang et al., 2013).

We used a data set with recorded 270 trials. Each trial started when the ball passed over the net while flying towards the human player, and ended when the ball returned by the human

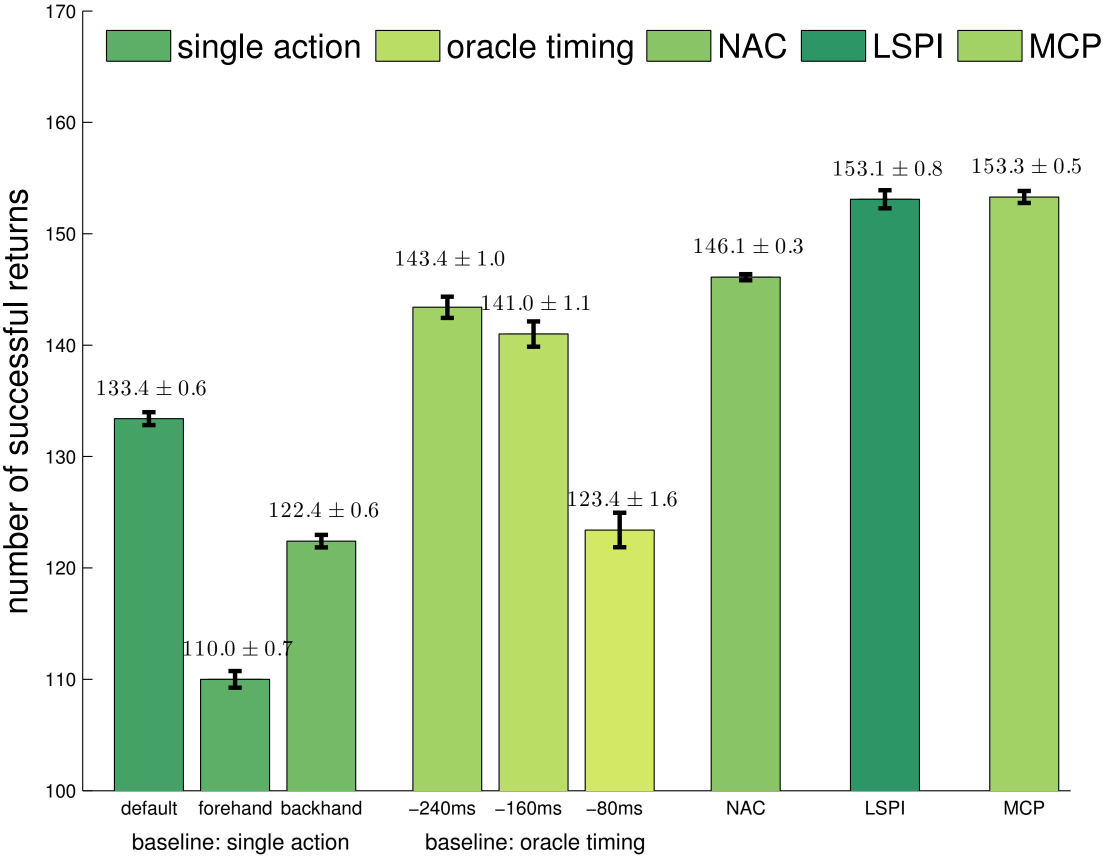


Figure 4.6: Performance of different methods evaluated in ten repetitions on the test data with 207 valid trials, based on enumerated training episodes. For each method, we evaluated the performance in terms of averaged number of successful returns, using cross-validation in each repetition. The first group of bars showed the baseline performance using only a single action (default, forehand, or backhand hitting movement) for all the test trials. The second group of bars showed the performance with oracle timing, where the action was always selected at a specific time (240ms, 160ms, and 80ms) before the opponent hit the ball. The last three bars showed the performance of the NAC algorithm, the LSPI algorithm, and the MCP algorithm. The error bars correspond to standard error of the mean estimated from ten repetitions.

player reached the robot's hitting plane (see Figure 4.2), including the trajectories of the ball and the player's racket. We excluded the trials where the ball was shot outside the overall hitting region of the robot, as shown in Figure 4.5, and evaluated the performance of the policy learning algorithm on the remaining 207 trials. For the basis functions used for function approximation in Equation (37), we chose 100 centers in the experiments.

4.3.2 Experimental Results

We evaluated the algorithms using ten-fold cross-validation. We repeated each round of the cross-validation for ten times to reduce the randomness in the robot player. In the first experiment, for each round of the cross-validation, we obtained sampled episodes $\mathscr{D} = \{(\boldsymbol{\theta}_i, a_i, r_i, \boldsymbol{\theta}'_i) | i = 1, \ldots, L\}$ on the nine subsamples for training, where all valid combinations of action and time to initiate were enumerated on each trial. For each episode, the reward, i.e., success of the robot's return, was obtained from the SL simulation, where a reward of 1 was given for successful return of the ball and -1 for failing to return. For the LSPI algorithm, the policy was learned from the sample $\mathscr{D}$ using Algorithm 4.1, and evaluated on the one subsample for test. The anticipatory action selection following the learned policy by LSPI led to successful returns for the robots in 153.1 ± 0.8 times, with an average success rate of 74%. We subsequently evaluated the MCP algorithm in the same manner. The action selection following Algorithm 4.3 led to successful returns for the robots in 153.3 ± 0.6 times, an average success rate of 74%.

For comparison, we evaluated a baseline that exclusively used a single type of hitting movement. Every type of hitting movement yielded a relatively high success rate in its designated regions. However, the overall rate on the entire data set was considerably reduced due to its poor performance in the other regions. Figure 4.6 showed the number of successful returns for using each action initiated at 240ms before the opponent hits the ball, such that the robot had sufficient time to complete the movement. The robot player without anticipation would achieve an average success rate of 64% for using only the default hitting movement, and 53% and 59% for using only the forehand and backhand movements, respectively. Therefore, both the LSPI algorithm and the MCP algorithm substantially improved the performance of the robot player by learning which action should be executed.

In addition, we considered another baseline with oracle timing, i.e., the action was always chosen at a specific time before the opponent has hit the ball. Without the waiting action, the LSPI algorithm and the MCP algorithm were equivalent, and both achieved the success return 143.4 ± 1.0 times at 240ms before the opponent has hit the ball, 141.0 ± 1.1 times at 160ms, and 123.4 ± 1.6 times at 80ms, where the performance tended to decline as the reaction delay was increased. One can see that the waiting action played an important role in trading off the prediction accuracy and the reaction delay. Note that these time steps were only used for this baseline method with oracle timing, and that the other evaluated algorithms were not aware of the hitting time of the opponent in advance. Nevertheless, both the LSPI algorithm and the MCP algorithm substantially outperformed this baseline performance by learning when a chosen action should be initiated.

Furthermore, we also compared to policy gradient methods (Sutton et al., 1999). Specifically, we adopted episodic Natural Actor-Critic (NAC) algorithm (Peters and Schaal, 2008), and a policy based on softmax action selection (Sutton and Barto, 1998). The NAC is a policy search method that directly estimates the gradient of the value function, which can be easier to do than estimating the value function itself. Moreover, one can use available domain knowledge to choose the type of policies that are searched. However, as an on-policy approach, the NAC is not very efficient in the use of sampled data. In addition, as in all gradient methods, the NAC can suffer from a premature convergence to a local optimum depending on the chosen step-size. In this application, the LSPI and MCP algorithms both outperformed the NAC algorithm, as shown in Figure 4.6.

In the second experiment, we compared the LSPI and the MCP in terms of sample efficiency. We obtained samples $\mathscr{D}$ from the training data by following a uniform policy, which chose the waiting action with a probability of $\frac{1}{2}$ and each of the three stopping actions with a probability of $\frac{1}{6}$. The first group of bars in Figure 4.7 was obtained by sampling one episode for each trial in

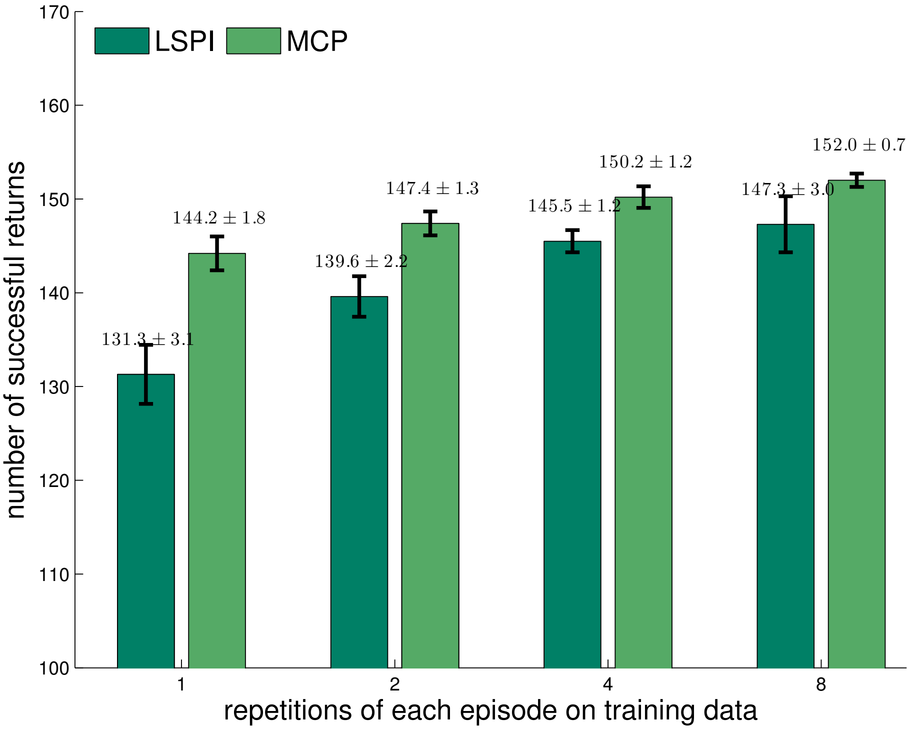


Figure 4.7: Performance of the LSPI and MCP methods evaluated in ten repetitions on the test data with 207 valid trials, based on sampled episodes on the training data. The numbers on the X-axis showed the number of sampled episodes on the training data.

the training data. This amount of data was insufficient for the LSPI to acquire a reliable estimate of the value function, especially for the waiting action. While the LSPI method performed even worse than the baseline that only chose the default hitting movement, the MCP method already achieved a substantial improvement by taking advantage of the estimated transition model in the IDDM. As we increased the number of sampled episode for each trial, the performance was improved for both methods. The MCP method always outperformed the LSPI method, although the advantages became smaller as more samples were available. The experimental results shown in Figure 4.7 verified that the MCP can be more sample-efficient than the LSPI. Due to the fact that sampling is expensive in many real-robot applications, we advocate the use of the MCP method for anticipatory action selection.

4.4 Conclusions of Chapter 4

In this chapter, we introduced new approaches to anticipatory action selection. We formulated the anticipatory action selection as optimal stopping in a partially observable Markov decision processes. We first presented a policy learning approach using the Lease-Square Policy Iteration algorithm. However, the LSPI can be sample-inefficient, as it does not exploit the transition model in the Intention-Driven Dynamics Models. Consequently, we presented the Monte-Carlo Planning approach, which benefits from the transition model estimated by the IDDM framework. Experimental results using real data and a simulated environment show that the anticipatory action selection can be used for a robot table tennis player to enhance its performance against human players, where the robot decided the timing to initiate a selected hitting movement according to the prediction of the human opponent. We also showed that both the LSPI and

MCP algorithms substantially improved the performance over the baselines, and that the MCP algorithm is more sample-efficient than the LSPI and, hence, more applicable to robot table tennis scenarios.

5 Opponent Modeling and Strategy Learning

In this chapter, we consider decision making solely based on the preference of the human partners, where observations are not sufficient for reliable intention inference. We formulate it as a repeated game and present a learning approach to safe strategies that exploit the humans' preferences. This chapter is based on Wang et al. (2011a).

5.1 Prologue

Opponent modeling allows a player to exploit the opponents' preferences and weaknesses in repeated games. Approaches that discard opponent modeling usually need to make worst-case assumptions, e.g. following a minimax strategy. Such approaches are considered safe as their expected payoff is lower-bounded by the minimax payoff. However, they lack the ability to exploit non-competitive or imperfect opponents in general-sum games. In contrast, the idea of fictitious play (Brown, 1951) has been extensively used for sixty years. Assuming the opponents are playing stationary strategies, fictitious play consists of modeling them by the empirical probabilities of the observed actions and then optimally responding according to the model. Using a sufficiently accurate model of a stationary opponent, fictitious play yields a higher expected payoff than any approach without modeling. However, even when the stationarity assumption holds, inaccurate models are inevitable for a limited number of observations, which exposes the player to adopting overly risky strategies.

Online learning algorithms (e.g., Zinkevich, 2003; Bowling, 2005) were developed to guarantee bounded regret in the worst case. However, the opponents are usually suboptimal (Simon, 1991) or not completely competitive (e.g. in general-sum games) in many real-world situations. A practical criterion (Powers et al., 2007) is concerned with learning to play optimally against stationary opponents or opponents whose strategies converge to a stationary one. For example, WoLF-IGA (Bowling and Veloso, 2002) and AWESOME (Conitzer and Sandholm, 2007) learn the best-response against stationary opponents, and converge in self-plays. However, these algorithms may adapt to unsafe counter-strategies due to inaccurate estimates of the opponents' strategies. To address the safety issue in opponent modeling, Markovitch and Reger (2005) proposed to infer a weakness model instead of estimating the precise model. McCracken and Bowling (2004) proposed an ϵ-safe learning algorithm that chooses the best counter-strategy from a safe set of strategies. Strategies in the corresponding set do not lose more than ϵ in the worst case. In more recent work, Johanson and Bowling (2009) proposed robust learners by considering restricted Nash responses and data-biased responses for imperfect information games.

In this chapter, we propose a different approach to modeling an opponent's strategy with a set of possible strategies that contains the actual strategy with a high probability. For simplicity, we limit the discussion in this chapter to the two-player games. Nevertheless, the modeling technique can be extended to multi-player games. Given a parameter δ that controls the safety-exploitability trade-off, a stationary opponent's strategy is modeled with a set of possible strategies that contains the actual one with probability no less than $1-\delta$. These possible strategies are chosen according to their consistency with the observations. We propose an algorithm that provides a counter-strategy with a lower-bound above the minimax payoff with probability no less than $1-\delta$. Such a strategy is said to be δ-safe, where δ is a parameter indicating the trade-off between the safety and the exploitability. The model of possible strategies shrinks along with the increased number of observations, and converges to the actual opponent's strategy. Therefore, the algorithm ensures the convergence of the counter-strategy to the actual best-response.

Unfortunately, the stationarity assumption is often unrealistic. A more reasonable assumption is local stationarity, i.e., the opponent's strategy is assumed to be stationary within a number of consecutive repetitions. A statistical hypothesis tests is used to detect significant changes in the opponent's strategy, so that the algorithm can efficiently adapt to them. Given a sequence of consecutive observations that are likely to be drawn from a locally stationary strategy, the proposed algorithm computes a locally δ-safe counter-strategy.

The proposed modeling technique allows a robot to improve its response to balls served by human opponents. The used robot setting (Mülling et al., 2011) has three possible high-level actions, i.e. setting to either a forehand, backhand, or middle hitting movement while the opponent serves. Each action has a relatively high success rate when the ball is served to its corresponding region. However, the robot is limited in its acceleration, resulting in low success rate for incoming balls far away from the preparation pose. Assuming that the opponent uses a stationary strategy, this algorithm generates counter-strategies such that the robot is more likely to successfully return the served ball. We use the low-level planner to evaluate the expected success rate of the learned strategies.

5.2 Strategy Learning in Normal-Form Games

The two participants are indicated by Player i and Player j, and the algorithm plays on behalf of Player i. A repeated game consists of several repetitions of a base game. Such a base game can either be a normal-form game or a stochastic game. A normal-form game is represented by reward matrices for both participants, among which the reward matrix for Player i is denoted by $\mathbf{R}$. In each game, two players choose their own actions a_i, a_j independently from action spaces $\mathscr{A}_i, \mathscr{A}_j$ respectively. The reward for Player i is given by $\mathbf{R}_{a_i,a_j}$ depending on their joint action.

For normal-form games, the strategies are probability distributions over all possible actions a_i and a_j, which we denote by $\pi_i(a_i) \in \Delta^{|\mathscr{A}_i|}$ and $\pi_j(a_j) \in \Delta^{|\mathscr{A}_j|}$, where $|\mathscr{A}_i|, |\mathscr{A}_j|$ are the size of $\mathscr{A}_i, \mathscr{A}_j$, and Δ^n is the n-simplex set $\{\pi \in \mathbb{R}^n | \pi^T \mathbf{1} = 1 \text{ and } \pi \succeq \mathbf{0}\}$. Consider the case when Player i has played the game N times and observed the opponent's actions $\{a_j^k | k = 1 \ldots N\}$. Assuming the actual opponent's strategy π_j^* is stationary during these N repetitions, the goal of the player is to learn a best-response strategy π_i against the opponent's possible strategies.

Fictitious play uses the empirical distribution $\tilde{\pi}_j(a_j) = N_{a_j}/N$ to model the opponent, where N_{a_j} is the number of times action a_j was observed. Given the opponent's model, Player i wants to learn a best-response strategy that maximizes its expected payoff. In normal-form games, the expected payoff for strategies π_i and π_j is computed as $\pi_i^T \mathbf{R} \pi_j$. When the model is a single estimate $\tilde{\pi}_j$, the best-response strategy is given by $\mathrm{BR}(\tilde{\pi}_j) = \arg\max_{\pi_i} \pi_i^T \mathbf{R} \tilde{\pi}_j$. The best-response strategy may tend to always take the same action, which can be unsafe when the opponent's model is not sufficiently precise.

To ensure the safety of the learned counter-strategy, we propose a δ-safe model of the opponent with a set of possible strategies instead of a single estimate. Furthermore, we compute the generalized best-response strategy against the proposed opponent's model, resulting in a δ-safe counter-strategy.

5.2.1 δ-Safe Opponent Modeling

Assume π_j^* is the true opponent's strategy, according to which the opponent's actions are drawn, and denote by $\tilde{\pi}_j$ the empirical distribution. According to Cover and Thomas (12.2.1 in 1991), the Kullback-Leibler (KL) divergence between $\tilde{\pi}_j$ and π_j^* is bounded by the following inequality with probability no less than $1-\delta$,

$$\mathrm{KL}(\tilde{\pi}_j||\pi_j^*) \leq \varepsilon(\delta) \triangleq \frac{(|\mathscr{A}_j|-1)\ln(N+1)-\ln(\delta)}{N}.$$

The opponent's possible strategies are selected by their distance to the empirically observed behavior, where the KL divergence serves as the natural measure of distance between probability distributions. The model of the opponent is given by

$$\Omega(\delta) \triangleq \{\pi_j \in \Delta^{|\mathscr{A}_j|} | \mathrm{KL}(\tilde{\pi}_j||\pi_j) \leq \varepsilon(\delta)\}.$$

The true opponent's strategy π_j^* lies in the set $\Omega(\delta)$ with probability no less than $1-\delta$.

In practice, the algorithms is not very sensitive to the choice of the parameter if $\delta \leq 0.5$. We set $\delta = C/(N+1)$, where $C \leq 1$ is a constant, so that the algorithm has an upper-bound on its regret.

5.2.2 δ-Safe Strategy Learning

With probability no less than $1-\delta$, the true opponent's strategy is inside $\Omega(\delta)$. However, the player has no further information to choose the real opponent's strategy among all possible strategies in this set. To learn a safe counter-strategy, we generalize the best-response strategy to be the counter-strategy that has the maximal expected payoff in the worst case. This problem can be formulated as:

$$\mathrm{BR}(\Omega(\delta)) \triangleq \arg\max_{\pi_i \in \Delta^{|\mathscr{A}_i|}} \min_{\pi_j \in \Omega(\delta)} \pi_i^T \mathbf{R} \pi_j. \tag{38}$$

Finding the best-response solution in Problem (38) is a convex optimization problem that can be solved efficiently using sub-gradient methods. When the opponent's model contains the true strategy π_j^*, the expected payoff of the best-response strategy has a lower-bound above the minimax payoff. Therefore, the learned strategy is safe with probability no less than $1-\delta$.

The resulting counter-strategy eventually converges to $\mathrm{BR}(\pi_j^*)$ if the opponent's strategy converges to a stationary strategy π_j^*. As the number of observations increases infinitely, the bound ε on the KL divergence converges to zero. For an infinite amount of data, the set of possible opponent's strategies shrinks to only one element which is the empirical estimate $\tilde{\pi}_j$, and the empirical distribution $\tilde{\pi}_j$ converges to π_j^*. As a result, the counter-strategy will eventually converge to the best-response against π_j^*.

The learned counter-strategy also converges sufficiently fast to the best-response against a stationary opponent. We now assume that the opponent uses a stationary strategy π_j^* and the game has been played t times. The *regret* is given by the difference of expected payoffs between the learned strategy and the best-response. As shown in the appendix, the expected regret for the next game is upper-bounded as $\mathbb{E}[r_t] \leq 4\beta\sqrt{2\varepsilon} + \delta\tau$, where β and τ are constants and $\varepsilon = (|\mathscr{A}_j|\ln(t+1) - \ln C)/t$. Thus, the expected accumulated regret in the first T games is upper-bounded as

$$\mathscr{R}_T = \sum_{t=1}^{T} \mathbb{E}[r_t] \leq \sum_{t=1}^{T} \left(c_1\sqrt{\ln(t+1)/t} + c_2/t\right),$$

where c_1 and c_2 are constants. As the partial sums of harmonic series have logarithmic growth rate and the partial sums of the series $\sqrt{\ln t/t}$ have growth rate of $\mathscr{O}(\sqrt{T \ln T})$, the accumulated regret bound has a growth rate of $\mathscr{O}(\sqrt{T \ln T})$. Therefore, the algorithm has zero average regret and converges fast.

5.2.3 Adaptive Strategy Update

Given N observed actions from consecutive games, the learned strategy is δ-safe only if these actions are drawn according to a same strategy π_j^*. However, the opponent may also update its strategy during the games. An adaptive learning algorithm is required to deal with the changes in the opponent's strategy and to learn against locally stationary strategies.

We propose an algorithm that accumulates observations and updates the counter-strategy when the opponent's strategy is locally stationary. It detects changes in the strategy of the opponent and recomputes the counter-strategy accordingly. The proposed algorithm, as outlined in Algorithm 5.1, maintains two sets of observed actions: a set $\mathbf{X}$ that contains observed actions for learning a counter-strategy, and a set $\mathbf{Y}$ for testing if the strategy has changed. The algorithm can update the counter-strategy π_i while the opponent is also adjusting its strategy against π_i.

In Step 8 of the algorithm, we test the hypothesis that the probability of executing action a_j is the same in local strategies that generated the sample set $\mathbf{X}$ and the validation set $\mathbf{Y}$. The sampled actions are converted into binary variables indicating if they are equal to a_j. Then, the empirical mean of these binary variables is tested with the hypotheses: $H_0 : p_X(a_j) = p_Y(a_j)$, vs. $H_1 : p_X(a_j) \neq p_Y(a_j)$.

Let $\mu = (\tilde{p}_X(a_j) + \tilde{p}_Y(a_j)/(|\mathbf{X}| + |\mathbf{Y}|)$, and $\sigma^2 = \mu(1-\mu)$. With large sample sets $\mathbf{X}$ and $\mathbf{Y}$, the statistic

$$z_0 = \frac{\tilde{p}_X(a_j) - \tilde{p}_Y(a_j)}{\sigma\sqrt{1/|\mathbf{X}| + 1/|\mathbf{Y}|}}$$

is approximately normally distributed, where $\tilde{p}_X(a_j)$ and $\tilde{p}_Y(a_j)$ are empirical probabilities of action a_j in $\mathbf{X}$ and $\mathbf{Y}$. The test of H_0 fails if $|z_0| > z_{\alpha/2}$, where α is the significance level of the test.

Algorithm 5.1: Adaptive algorithm for learning counter-strategies in normal-form games.

```
$\mathbf{X} := \emptyset$.;
Initialize $\pi_i$ to be the minimax strategy;
repeat
    while $|\mathbf{X}| < N_{\min}$ do
        Draw an action $a_i$ from the current strategy $\pi_i$;
        Observe a new sample $a_j$;
        $\mathbf{X} := \mathbf{X} \cup \{a_j\}$;
        Update $\pi_i$ by solving Problem (38);
    repeat
        Get $N_{\min}$ new samples as $\mathbf{Y}$;
        Perform two-sample tests for each action $a_j$ in $\mathbf{X}$ and $\mathbf{Y}$ ;
        if all the tests were passed then
            $\mathbf{X} := \mathbf{X} \cup \mathbf{Y}$;
            Update $\pi_i$ by solving Problem (38);
    until test failed or game is over ;
    $\mathbf{X} := \emptyset$;
    Reset $\pi_i$ to be the minimax strategy;
until the game is over. ;
```

5.3 Extension to Finite Stage Stochastic Games

For stochastic games, we only consider those with a finite number of stages, which can be represented by a tree structure of states $s \in \mathscr{S}$. The game starts from an initial state s_0 at Stage 1. At each state s, the players choose their actions a_i^s and a_j^s from action spaces $\mathscr{A}_i^s$ and $\mathscr{A}_j^s$ respectively. Let Children(s) denote the set of states that possibly follow after the state s. The game transfers to a subsequent state $s_k \in$ Children(s) with probability $P_s(s_k|a_i^s, a_j^s)$ based on the joint action, and Player i receives an immediate reward $\mathbf{R}^s_{a_i^s, a_j^s}$. The game terminates when a leaf state s_l is reached/ The cumulative reward for Player i is the sum of all rewards that it obtained in all stages. For stochastic games, a strategy consists of local strategies at every state.

Algorithm 5.1 can be extended to repeated stochastic games with a finite number of stages, where the proposed opponent modeling technique is used at every state. We model the opponent's strategy at state s by the set $\Omega^s(\delta) \triangleq \{\pi_j^s \in \Delta^{|\mathscr{A}_j^s|} | \mathrm{KL}(\tilde{\pi}_j^s || \pi_j^s) \leq \varepsilon(\delta, N_s)\}$, where the function $\varepsilon(\delta, N_s) = ((|\mathscr{A}_j^s| - 1)\ln(N_s + 1) - \ln(\delta / |\mathscr{S}|))/N_s$ depends on N_s, i.e. the number of observations available at state s.

A different model of the opponent is used at every state. The counter-strategy is recursively computed, starting with the states in the last stage. At a state in the last stage, only the immediate reward is considered. Therefore, learning the counter-strategy is the same as in normal-form games, resulting in a δ-safe estimate of expected payoff at this state. The estimated payoff is back propagated to the previous stage as a lower-bound of the future expected reward. The reward at the previous stage is given by the sum of the immediate reward and the estimated future reward. Then, the counter-strategy at the previous stage is learned with the updated

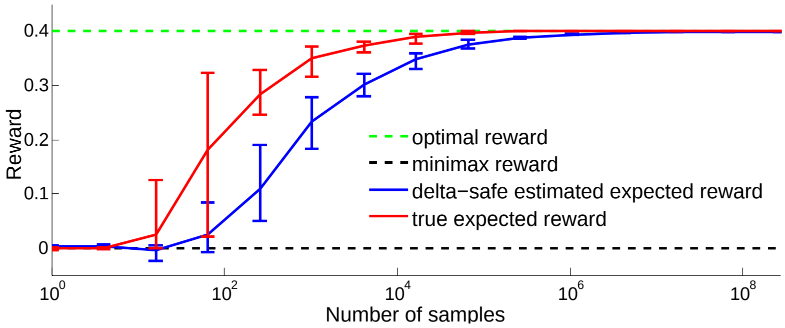


Figure 5.1: For a stationary opponent's strategy, the expected reward grew with the increasing number of samples, and converged to the optimal reward. The error bars showed the maximal and minimal reward in 20 repeated tests.

reward matrices. The counter-strategies at every state of each stage are obtained by traversing the state tree in a bottom-up order.

5.4 Evaluation

We first evaluate our algorithms in rock-paper-scissors games. Then we apply it to a table tennis robot to improve the performance of returning served balls.

5.4.1 Rock-Paper-Scissors

Rock-paper-scissors is a zero-sum normal-form game. The reward matrix for Player i is defined in Table 5.1. We represent strategies π_i and π_j by vectors with elements corresponding to the probabilities of taking Rock, Paper, and Scissor, respectively. The minimax strategy in the game draws actions uniformly, i.e. $\pi_i = [1/3, 1/3, 1/3]$. If Player i follows this uniform strategy, its expected reward is zero regardless of what the opponent's strategy is.

First, we evaluated the opponent modeling technique by playing against a stationary opponent's strategy $\pi_j^* = [0.6, 0.2, 0.2]$. The performance of the algorithm was tested on samples ranging exponentially from 1 to 10^8 observations. For each sample size, the experiment was repeated 20 times. The plot in Figure 5.1 showed the estimated expected reward and true expected reward. The true reward was the expected reward when the learned strategy was played against the true opponent's strategy. It was lower-bounded by the estimated reward with probability no less than $1-\delta$, which was the expected reward when the learned strategy was played against the worst-case opponent's strategy in the model. The reward gradually converged to the optimal reward, showing that the algorithm could exploit the opponent with sufficient observed data.

	Rock	Paper	Scissor
Rock	0	-1	1
Paper	1	0	-1
Scissor	-1	1	0

Table 5.1: The payoff matrix in rock-paper-scissors games.

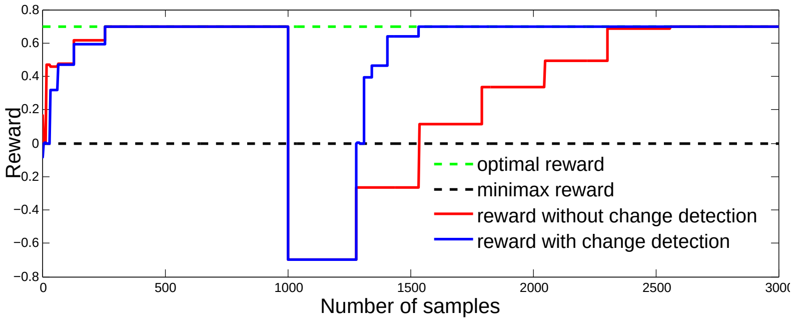


Figure 5.2: Comparison of true reward by algorithms with or without change detection. The algorithm with change detection could adapt to strategy switch more efficiently. The step curve was caused by the minimal sample size $N_{\text{min}} = 256$.

Then, we evaluated the change detection mechanism in Algorithm 5.1. The game had three thousand repetitions, where the opponent used a stationary strategy $[0.8, 0.1, 0.1]$ for the first thousand actions and changed to another stationary strategy $[0.1, 0.1, 0.8]$ for the remaining two thousand actions. After the first thousand repetitions, the learned strategy π_i tended to take Paper more frequently than other actions. Therefore, the sudden switch of the opponent's strategy caused a considerable loss in the expected reward. As illustrated by Figure 5.2, the algorithm showed its efficiency in adjusting to such situations.

Subsequently, considering that human players changed their strategies based on previous games in practice, we used a two-stage form of the regular rock-paper-scissors as an example of stochastic games with a finite number of stages. We chose the reward matrix in Stage 2 that was twice as much as it is in Stage 1. The game states could be represented by a two-level tree of states. The root state s_0 corresponded to a regular rock-paper-scissors game at Stage 1, and the game transferred to one following states according to the executed joint action.

We designed an opponent with a stationary strategy [0.6, 0.2, 0.2] in the first stage. Its preferences in Stage 2 were shown in Table 5.2, as well as the conditions for transferring to a subsequent state s_i.

The performance of the learning algorithm was evaluated by its expected reward. As shown in Figure 5.3, it converged to the globally optimal reward.

s_i	$a_i^{s_0}$	$a_j^{s_0}$	$\pi_j^{s_i}$	s_i	$a_i^{s_0}$	$a_j^{s_0}$	$\pi_j^{s_i}$
s_1	R	R	1/3,1/3,1/3	s_2	R	P	1/5,3/5,1/5
s_3	R	S	2/5,2/5,1/5	s_4	P	R	1/5,2/5,2/5
s_5	P	P	1/3,1/3,1/3	s_6	P	S	1/5,1/5,3/5
s_7	S	R	3/5,1/5,1/5	s_8	S	P	2/5,1/5,2/5
s_9	S	S	1/3,1/3,1/3				

Table 5.2: Local strategies for the designed opponent. (i) It tended to avoid taking the previous action if it lost the first stage. (ii) It preferred to take the previous action if it won the first stage. (iii) It drew actions uniformly if the first stage was a tie game.

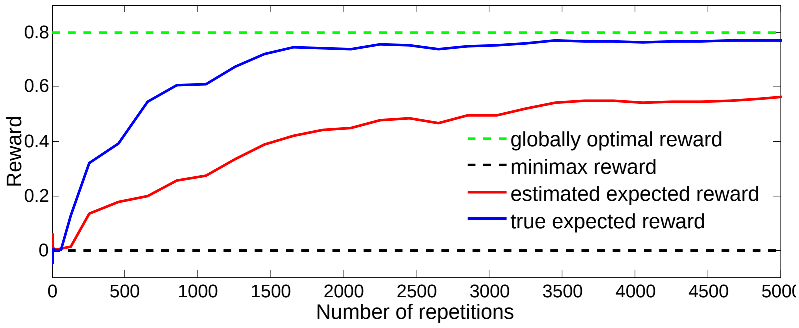
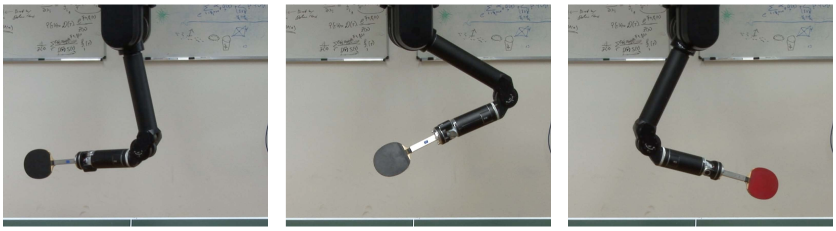


Figure 5.3: In two-stage rock-paper-scissors games, both the estimated expected reward and true expected reward converged to the globally optimal reward.



(a) forehand pose (b) middle pose (c) backhand pose

Figure 5.4: Three hitting movements offered by the robot. Each hitting movement was optimized for hitting points its specific region.

5.4.2 Returning Served Table Tennis Balls

The proposed modeling technique was used to learn strategies, which allowed the robot to improve its response to table tennis balls served by human opponents. The used table tennis robot offered three high-level actions, namely, forehand, backhand, or middle hitting movements. Each action had a relatively high success rate when the ball was served to its corresponding region, see Figure 5.4. However, the robot was limited by its torque and acceleration. It had a low success rate if the served ball was far from the chosen preparation pose. The proposed algorithm allowed generating strategies such that the robot would be more likely to successfully return the balls. We used the low-level planner to evaluate the learned strategies (Mülling et al., 2011).

The robot chose its preparation pose while the opponent served. We formulated the problem using a repeated two-player game. The empirical reward matrix for the robot could be estimated using the low-level planner with 600 recorded served ball trajectories from different human players (Mülling et al., 2011). As shown in Table 5.3, we knew how well a robot's action could return balls in those three regions. According to this reward matrix, the minimax play followed the strategy of $[0.2088, 0, 0.7912]$, whose elements corresponded to choosing the forehand, backhand and middle hitting movements respectively. It assumed a competitive opponent and preferred to choose the middle position. However, it was not the optimal strategy if an opponent tended to serve the ball to the right region more frequently.

We recruited three volunteers to repeatedly serve the ball. The experiment with each volunteer consisted in over one hundred trials. For each trial, the low-level planer provided three

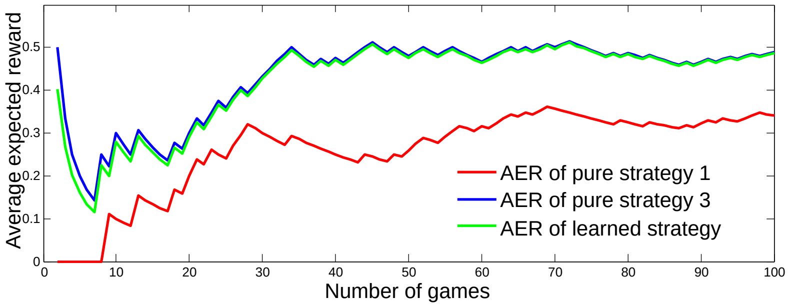


Figure 5.5: The curves showed the average expected reward against Volunteer 1. The pure strategy 1 always chose the forehand action. The pure strategy 3 always chose the middle hitting movement, which was the optimal strategy.

binary outputs, indicating whether a high-level action was successfully return the served ball according to its trajectory. The learned stochastic strategy at each trial was evaluated by the expected reward, i.e., the success rate. To analyze the performance of the algorithms, we showed the curve of its Average Expected Reward (AER), which was the sum of its expected reward divided by the number of trials.

Volunteer 1 served the balls with approximately a uniform distribution over the three regions. Therefore, *pure strategy* 3, which always chose action 3, led to the maximal average expected reward. The results were shown in Figure 5.5. The performance of the δ-safe strategies were slightly below that of the pure strategy 3 in the beginning, as it started from playing the minimax strategy, yet quickly converged to the optimal strategy.

Volunteer 2 had a strong preference to serve the balls to the right region. According to the results in Figure 5.6, the learned strategies gradually shifted towards *pure strategy* 1, which was the optimal counter-strategy against this opponent.

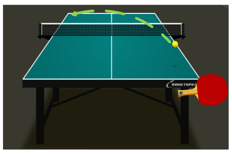
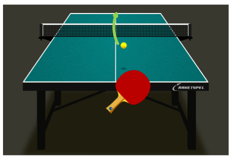
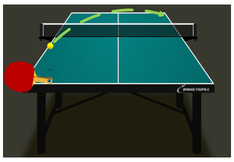
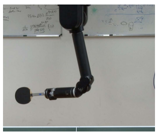
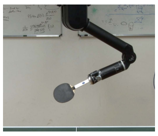
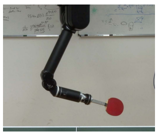
			
	0.65	0.52	0.06
	0.41	0.74	0.56
	0.04	0.12	0.39

Table 5.3: The empirical reward matrix obtained from recorded robot table tennis games.

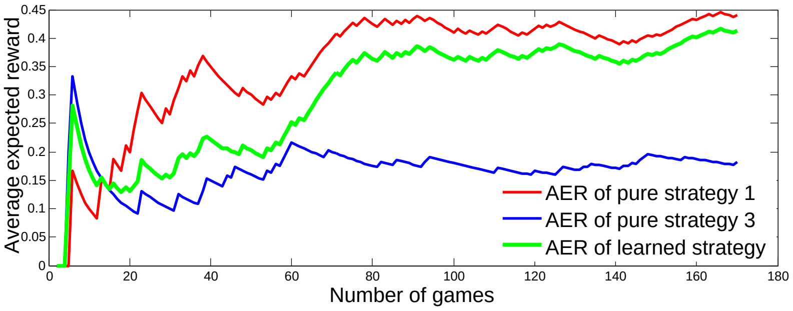


Figure 5.6: The curves showed the average expected reward against Volunteer 1. The pure strategy 1 was the optimal counter-strategy.

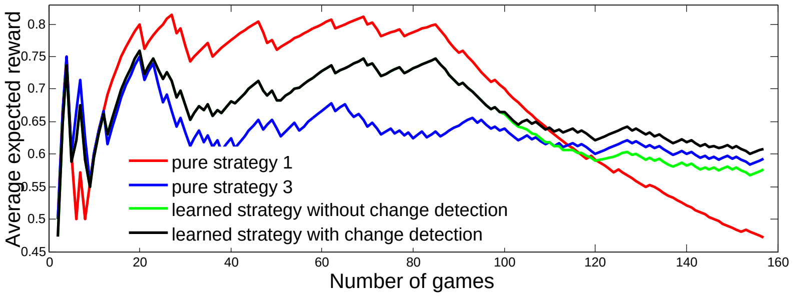


Figure 5.7: The algorithm with change detection outperformed other strategies against Volunteer 3.

Volunteer 3 started with a strategy that preferred to serve the ball to the middle/right side, and intentionally switched his strategy after around 85 trials to preferring the middle/left side. Pure strategies 1 and 3 were respectively the optimal counter-strategy before and after the switch. We compared the Algorithm 5.1 to its simplified version without change detection in this case. As shown in Figure 5.7, both algorithms had decreased performance immediately after the switch happens. The proposed algorithm successfully detected the strategy switch after playing one hundred trials and adapted to it by re-initializing to the minimax strategy. It outperformed other strategies after the switch, and resulted in the best average expected reward for all trials.

The algorithm ran efficiently in real table tennis games. We used the projected sub-gradient method to find the optimal counter-strategy in each repetition of the game, initialized with the counter-strategy used in the previous game. In the table tennis setup, the algorithm took less than 50ms on average to compute a response for each serve.

5.5 Conclusions of Chapter 5

We introduced a new opponent modeling technique, which models the opponent by a set of strategies whose KL divergence from the empirical distribution is bounded. Based on it, we proposed δ-safe algorithms for both normal-form games and stochastic games with a finite number of stages. The algorithms learn strategies whose expected reward has a lower-bound of minimax payoff with probability no less than $1 - \delta$. We showed that the learned strategies converged to the best-response strategy and had bounded regret. We also employed two-sample tests to deal with locally stationary opponents. We evaluated the performance of our algorithms in rock-paper-scissors games and the robot table tennis setting. The experimental results showed that the proposed algorithms could balance safety and exploitability in opponent modeling, and could adapt to changes in the opponent's strategy.

5.A Derivation of Expected Regret Bound

The expected regret for the game $t+1$ is $\mathbb{E}[r_t] = \boldsymbol{\pi}_i^{*T}\mathbf{R}\boldsymbol{\pi}_j^* - \boldsymbol{\pi}_i^{tT}\mathbf{R}\boldsymbol{\pi}_j^*$, where $\boldsymbol{\pi}_i^*$ is the best-response against $\boldsymbol{\pi}_j^*$. There are two situations.

(1): When $\boldsymbol{\pi}_j^* \in \Omega(\delta)$, $\|\boldsymbol{\pi}_j^* - \boldsymbol{\pi}_j^t\| \leq \|\boldsymbol{\pi}_j^t - \tilde{\boldsymbol{\pi}}_j^t\| + \|\boldsymbol{\pi}_j^* - \tilde{\boldsymbol{\pi}}_j^t\| \leq \sqrt{2\mathrm{KL}(\tilde{\boldsymbol{\pi}}_j^t||\boldsymbol{\pi}_j)} + \sqrt{2\mathrm{KL}(\tilde{\boldsymbol{\pi}}_j^t||\boldsymbol{\pi}_j^*)} \leq 2\sqrt{2\varepsilon}$, where $\tilde{\boldsymbol{\pi}}_j^t$ is the empirical distribution obtained during those t games. $\mathbb{E}[r_t] = \boldsymbol{\pi}_i^{*T}\mathbf{R}\boldsymbol{\pi}_j^* - \boldsymbol{\pi}_i^{tT}\mathbf{R}\boldsymbol{\pi}_j^*$. As $\boldsymbol{\pi}_i^t$ is a best-response strategy against $\boldsymbol{\pi}_j^t$, $\boldsymbol{\pi}_i^{tT}\mathbf{R}\boldsymbol{\pi}_j^t \geq \boldsymbol{\pi}_i^{*T}\mathbf{R}\boldsymbol{\pi}_j^t$.

$$
\begin{aligned}
\mathbb{E}[r_t] &= \boldsymbol{\pi}_i^{*T}\mathbf{R}\boldsymbol{\pi}_j^* - \boldsymbol{\pi}_i^{tT}\mathbf{R}\boldsymbol{\pi}_j^t + \boldsymbol{\pi}_i^{tT}\mathbf{R}\boldsymbol{\pi}_j^t - \boldsymbol{\pi}_i^{tT}\mathbf{R}\boldsymbol{\pi}_j^* \\
&\leq (\boldsymbol{\pi}_i^{*T}\mathbf{R}\boldsymbol{\pi}_j^* - \boldsymbol{\pi}_i^{*T}\mathbf{R}\boldsymbol{\pi}_j^t) \\
&\quad + (\boldsymbol{\pi}_i^{tT}\mathbf{R}\boldsymbol{\pi}_j^t - \boldsymbol{\pi}_i^{tT}\mathbf{R}\boldsymbol{\pi}_j^*) \\
&= \boldsymbol{\pi}_i^{*T}\mathbf{R}(\boldsymbol{\pi}_j^* - \boldsymbol{\pi}_j^t) + \boldsymbol{\pi}_i^{tT}\mathbf{R}(\boldsymbol{\pi}_j^t - \boldsymbol{\pi}_j^*) \\
&\leq \left(\max\{|\boldsymbol{\pi}_i^{*T}\mathbf{R}|\} + \max\{|\boldsymbol{\pi}_i^{tT}\mathbf{R}|\}\right) \|\boldsymbol{\pi}_j^t - \boldsymbol{\pi}_j^*\| \\
&\leq 4\beta\sqrt{2\varepsilon},
\end{aligned}
$$

where $\beta \triangleq \max_{a_i,a_j} \mathbf{R}_{a_i,a_j}$.

(2): When $\boldsymbol{\pi}_j^* \notin \Omega(\delta)$,

$$
r_t \leq \tau \triangleq \max_{a_i,a_j} \mathbf{R}_{a_i,a_j} - \min_{a_i,a_j} \mathbf{R}_{a_i,a_j}.
$$

Since the situation (1) happens with probability no less than $1-\delta$ and it has lower expected regret than the situation (2), the expected regret is bounded $\mathbb{E}[r_t] \leq 4\beta\sqrt{2\varepsilon} + \delta\tau$.

6 Conclusions and Future Directions

This thesis presented a machine-learning approach to intention inference and decision making, the key components towards the anticipatory robots. We proposed the Hierarchical Gaussian Process Dynamics Model in Chapter 2 for complex human movements that are directed by exogenous driving factors, such as the goal, intention, or gait style. We introduced a variational Bayesian method for inferring the exogenous variables and the latent states simultaneously. When the exogenous variables are the underlying intention of human movements, the H-GPDM leads the Intention-Driven Dynamics Model for intention inference. To fulfill the real-time requirements in many human-robot interaction scenarios, we proposed an online intention inference method in Chapter 3. Based on the prediction obtained by the online inference method, the anticipatory robot can choose its action in a proactive manner. We formulated the anticipatory action selection with a POMDP, and addressed the trade-off between prediction accuracy and reaction delay involved in such decision making processes using both model-free policy learning and Monte-Carlo planning. In addition, we considered decision making solely based on the preference of the human partners. We formulated it as a repeated game and presented a learning approach to safe strategies that exploited the humans' preferences. Finally, we built a prototype robot table tennis player to demonstrate the merit of the proposed approach to intention inference and decision making.

6.1 Summary of the Thesis

This thesis contribute to the developments of anticipatory robots by proposing novel models and developing practical methods for the prediction and the proactive action selection.

The H-GPDMs, introduced in Chapter 2, are a flexible class of latent-variable models for representing complex nonlinear human movements that are driven by exogenous driving factors. We incorporated the exogenous variables in the dynamics model to improve the interpretation, analysis, and prediction of human movements. The H-GPDM can be learned from observed movements in both supervised and unsupervised settings. We applied the H-GPDM to jointly infer the exogenous variables and the missing observations. While exact inference is analytically intractable, we introduced a variational inference method. We analyzed the performance of our proposed VB inference in three applications, i.e., target prediction from human-robot table tennis, character recognition and recovery from handwriting trajectories, and gait recognition using motion capture data. The experimental results demonstrated the merit of both the H-GPDM and the inference algorithm.

We have discussed the Intention-Driven Dynamics Model (IDDM) in Chapter 3, a special case of the H-GPDMs, where the intention of the human is considered as the exogenous driving factor. We introduced efficient online inference algorithms that allowed real-time inference. We verified the proposed model in two human-robot interaction scenarios, namely, target inference in robot table tennis and action recognition for interactive robots. In these two scenarios, we showed that modeling the intention-driven dynamics achieved better predictions than algorithms without modeling the dynamics. Hence, we advocate the use of IDDM when the movement is driven by the intention (or target to predict), as the IDDM captures the causal relationship of the intention and the observed movements.

We introduced approaches to anticipatory action selection based on the prediction of the opponent's intention in Chapter 4. By formulating the anticipatory action selection as optimal stopping in POMDP, we presented two methods, namely, the least-square policy iteration algorithm for policy learning and the Monte-Carlo planning algorithm for decision making. Experimental results showed that the anticipatory action selection could be used for a robot table tennis player to enhance its performance against human players, where the robot decided timing for initiating a selected hitting movement according to the prediction of the human opponent.

We introduced a new strategy learning technique in Chapter 5, which modeled the opponent's preference by a set of strategies whose KL divergence from the empirical distribution is bounded. We proposed δ-safe algorithms for both normal-form games and stochastic games with a finite number of stages. The algorithms learned strategies whose expected reward has a lower-bound of minimax payoff with probability no less than $1-\delta$. We showed that the learned strategies converged to the best-response strategy and had bounded regret. We evaluated the performance of our algorithms in rock-paper-scissors games and the robot table tennis setting. The experimental results showed that the proposed algorithms could balance safety and exploitability in opponent modeling.

6.2 Open Problems

Looking beyond the presented work, there are several potential directions that one can explore in the future.

The presented H-GPDM has no intra-layer connection between the exogenous variables. In practice, there may be correlation among the top-level exogenous variables. For example in the table tennis scenario, the striking styles such as forehand or backhand and the intended target to hit are often correlated. In addition, the exogenous variables can evolve temporally, which could correspond to moving targets, time-varying walking styles, or transition between different gait dynamics. Such a complex graphical model renders inference even more challenging than that of the presented H-GPDMs. One needs to introduce novel Bayesian inference methods to handle the correlated exogenous variables.

The presented idea and methods of intention inference are also applicable to other scenarios of human-robot or human-computer interaction. For example, the IDDM can be applied to tracking a subject's intended movement from ECoG data (Pistohl et al., 2008). Here, tracking relies on a transition model of the subject's state, e.g., hand position and velocity. Such transition model depends on the target that the subject's hand intends to reach, as reaching difference targets corresponds to difference dynamics. Intention inference and tracking are dependent tasks and should be treated jointly. The IDDMs are applicable to simultaneously tracking the hand's state and inferring the intended target. Similarly, the proposed intention inference methods have potential applications for tracking pedestrians in motion (Ziebart et al., 2009), pointing target prediction for intelligent interface design (Ziebart et al., 2012), and activity recognition with smart phones (Frank et al., 2011). Proactive planning is also of great interest for building anticipatory systems. Extending the presented anticipatory action selection methods, one can deal with more complex planning problems where continuous action space is involved (Spaan and Vlassis, 2005; Antos et al., 2007).

6.3 Publications

The research presented in this thesis has been partly published in or has lead to the following publications.

Zhikun Wang, Katharina Mülling, Marc P. Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. Probabilistic Movement Modeling for Intention Inference in Human-Robot Interaction. *International Journal of Robotics Research*, 32(7):841–858, 2013.

Zhikun Wang, Marc P. Deisenroth, Kun Zhang, Abdeslam Boularias, Bernhard Schölkopf, and Jan Peters. Hierarchical gaussian process dynamics models for human movement analysis. *Journal of Machine Learning Research*, submitted .

Zhikun Wang, Abdeslam Boularias, Katharina Mülling, Bernhard Schölkopf, and Jan Peters. Anticipatory action selection in human-robot table tennis. *Artificial Intelligence*, submitted .

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

Zhikun Wang, Marc P. Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schölkopf, and Jan Peters. Probabilistic Modeling of Human Movements for Intention Inference. In *Proceedings of Robotics: Science and Systems (R:SS)*, 2012.

Zhikun Wang, Christoph H Lampert, Katharina Mulling, Bernhard Scholkopf, and Jan Peters. Learning anticipation policies for robot table tennis. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 332–337, 2011.

Zhikun Wang, Abdeslam Boularias, Katharina Mülling, and Jan Peters. Balancing safety and exploitability in opponent modeling. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2011.

References

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-first International Conference on Machine Learning*.

Acosta, L., Rodrigo, J., Mendez, J., Marichal, G., and Sigut, M. (2003). Ping-pong player prototype. *IEEE Robotics and Automation Magazine*, 10(4):44–52.

Aglioti, S. M., Cesari, P., Romani, M., and Urgesi, C. (2008). Action anticipation and motor resonance in elite basketball players. *Nature Neuroscience*, 11(9):1109–1116.

Alexander, M. and Honish, A. (2009). Table tennis: a brief overview of biomechanical aspects of the game for coaches and players. Technical report, Faculty of Kinesiology and Recreation Management, University of Manitoba.

Anderson, R. (1988). *A Robot Ping-Pong Player: Experiment in Real-time Intelligent Control*. MIT Press.

Ángel, L., Sebastián, J., Saltarén, R., Aracil, R., and Gutiérrez, R. (2005). RoboTenis: design, dynamic modeling and preliminary control. In *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 747–752.

Antos, A., Szepesvári, C., and Munos, R. (2007). Fitted Q-iteration in continuous action-space MDPs. In *Advances in Neural Information Processing Systems 20*, pages 9–16.

Auer, P. (2003). Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422.

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

Baker, C. L., Tenenbaum, J. B., and Saxe, R. (2006). Bayesian models of human action understanding. In *Advances in Neural Information Processing Systems 18*, pages 99–106, Cambridge, MA. MIT Press.

Bandyopadhyay, T., Won, K. S., Frazzoli, E., Hsu, D., Lee, W. S., and Rus, D. (2013). Intention-aware motion planning. In *Algorithmic Foundations of Robotics X*, pages 475–491. Springer.

Billingsley, J. (1984). Machineroe joins new title fight. *Practical Robotics*, pages 14–16.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Bitzer, S. and Vijayakumar, S. (2009). Latent spaces for dynamic movement primitives. In *Proceedings of IEEE-RAS International Conference on Humanoid Robots*, pages 574–581. IEEE.

Bowling, M. (2005). Convergence and no-regret in multiagent learning. In *Advances in Neural Information Processing Systems 17*, page 209.

Bowling, M. and Veloso, M. (2002). Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250.

Brown, G. (1951). Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376.

Chang, C. and Lin, C. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27.

Conitzer, V. and Sandholm, T. (2007). AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1):23–43.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley & Sons.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society.

Damianou, A. C., Titsias, M., and Lawrence, N. D. (2011). Variational Gaussian Process Dynamical Systems. In *Advances in Neural Information Processing Systems 24*, pages 2510–2518.

Davies, D. and Armstrong, M. (1989). *Psychological factors in competitive sport*. Psychology Press.

Décamps, J.-P., Mariotti, T., and Villeneuve, S. (2005). Investment timing under incomplete information. *Mathematics of Operations Research*, 30(2):472–500.

Deisenroth, M. P. (2010). *Efficient Reinforcement Learning using Gaussian Processes*. KIT Scientific Publishing.

Deisenroth, M. P., Huber, M. F., and Hanebeck, U. D. (2009). Analytic moment-based Gaussian process filtering. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 225–232. ACM.

Deisenroth, M. P. and Mohamed, S. (2012). Expectation Propagation in Gaussian Process Dynamical Systems. In *Advances in Neural Information Processing Systems 25*, pages 2618–2626.

Deisenroth, M. P. and Ohlsson, H. (2011). A general perspective on Gaussian filtering and smoothing. In *Proceedings of American Control Conference 2011*, pages 1807–1812. IEEE.

Deisenroth, M. P., Turner, R. D., Huber, M. F., Hanebeck, U. D., and Rasmussen, C. E. (2012). Robust filtering and smoothing with Gaussian processes. *IEEE Transactions on Automatic Control*, 57(7):1865–1871.

Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(02):185–205.

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

Dulac-Arnold, G., Denoyer, L., Preux, P., and Gallinari, P. (2012). Sequential approaches for learning datum-wise sparse representations. *Machine Learning*, 89(1-2):87–122.

Farley, C. T. and McMahon, T. A. (1992). Energetics of walking and running: insights from simulated reduced-gravity experiments. *Journal of Applied Physiology*, 73(6):2709–2712.

Fässler, H., Beyer, H., and Wen, J. (1990). A robot ping pong player: optimized mechanics, high perfromance 3d vision, and intelligent sensor control. *Robotersysteme*, 6(3):161–170.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Fleet, D. J. (2011). Motion Models for People Tracking. *Visual Analysis of Humans*, pages 171–198.

Frank, J., Mannor, S., and Precup, D. (2011). Activity recognition with mobile phones. In *Proceedings of Machine Learning and Knowledge Discovery in Databases*, pages 630–633. Springer.

Friesen, A. L. and Rao, R. P. (2011). Gaze following as goal inference: A Bayesian model. In *Proceedings of Annual Conference of the Cognitive Science Society*.

Gaudel, R. and Sebag, M. (2010). Feature selection as a one-player game. In *Proceedings of the 27th International Conference on Machine Learning*.

Ghahramani, Z. and Roweis, S. (1999). Learning nonlinear dynamical systems using an em algorithm. In *Advances in Neural Information Processing Systems 12*.

Girard, A. (2004). *Approximate methods for propagation of uncertainty with Gaussian process models*. PhD thesis, University of Glasgow.

Girard, A., Rasmussen, C. E., Quiñonero-Candela, J., and Murray-Smith, R. (2002). Gaussian Process Priors with Uncertain Inputs—Application to Multiple-step ahead Time Series Forecasting. In *Advances in Neural Information Processing Systems 15*, pages 529–536.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press.

Hauser, K. (2012). Recognition, prediction, and planning for assisted teleoperation of freeform tasks. In *Proceedings of Robotics: Science and Systems*.

Hoffman, G. and Breazeal, C. (2007). Cost-based anticipatory action selection for human–robot fluency. *IEEE Transactions on Robotics*, 23(5):952–961.

Huang, Y., Xu, D., Tan, M., and Su, H. (2013). Adding active learning to LWR for ping-pong playing robot. *IEEE Transactions on Control Systems Technology*, 21(4):1489 – 1494.

Ijspeert, A., Nakanishi, J., and Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proceedings of IEEE International Conference on Robotics and Automation*.

Jenkins, O., Serrano, G., and Loper, M. (2007). Interactive human pose and action recognition using dynamical motion primitives. *International Journal of Humanoid Robotics*, 4(2):365–386.

Jensen, U. and Hsu, G.-H. (1993). Optimal stopping by means of point process observations with applications in reliability. *Mathematics of Operations Research*, 18(3):645–657.

Johanson, M. and Bowling, M. (2009). Data biased robust counter strategies. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*.

Khan, M., Mohamed, S., Marlin, B., and Murphy, K. (2012). A stick-breaking likelihood for categorical data analysis with latent Gaussian models. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*.

Ko, J. and Fox, D. (2009). GP-BayesFilters: Bayesian Filtering using Gaussian Process Prediction and Observation Models. *Autonomous Robots*, 27(1):75–90.

Ko, J. and Fox, D. (2011). Learning GP-BayesFilters via Gaussian process latent variable models. *Autonomous Robots*, 30(1):3–23.

Kuderer, M., Kretzschmar, H., Sprunk, C., and Burgard, W. (2012). Feature-based prediction of trajectories for socially compliant navigation. In *Proceedings of Robotics: Science and Systems*.

Kurniawati, H., Du, Y., Hsu, D., and Lee, W. (2011). Motion planning under uncertainty for robotic tasks with long time horizons. *International Journal of Robotics Research*, 30(3):308–323.

Lagoudakis, M. and Parr, R. (2003). Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107–1149.

Lampert, C. and Peters, J. (2012). Real-time detection of colored objects in multiple camera streams with off-the-shelf hardware components. *Journal of Real-Time Image Processing*, 7(1):31–41.

Lawrence, N. (2005). Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816.

Lawrence, N. D. (2004). Gaussian Process Latent Variable Models for Visualization of High Dimensional Data. In *Advances in Neural Information Processing Systems 16*, pages 585–591.

Lawrence, N. D. and Moore, A. J. (2007). Hierarchical Gaussian Process Latent Variable Models. In *Proceedings of the 24th international conference on Machine learning*, pages 481–488. ACM.

Li, R., Tian, T.-P., and Sclaroff, S. (2012). Divide, Conquer and Coordinate: Globally Coordinated Switching Linear Dynamical System. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:654–669.

Liao, L., Patterson, D. J., Fox, D., and Kautz, H. (2007). Learning and Inferring Transportation Routines. *Artificial Intelligence*, 171(5):311–331.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of IEEE International Conference on Computer Vision*, page 1150.

Markovitch, S. and Reger, R. (2005). Learning and exploiting relative weaknesses of opponent agents. *Autonomous Agents and Multi-Agent Systems*, 10(2):103–130.

Matsushima, M., Hashimoto, T., Takeuchi, M., and Miyazaki, F. (2005). A learning approach to robotic table tennis. *IEEE Transactions on Robotics*, 21(4):767–771.

Mazziotto, G. (1986). Approximations of the optimal stopping problem in partial observation. *Journal of Applied Probability*, 23:341–354.

McCracken, P. and Bowling, M. (2004). Safe strategies for agent modelling in games. In *Proceedings of AAAI Fall Symposium on Artificial Multi-agent Learning*, pages 103–110.

Miyazaki, F., Matsushima, M., and Takeuchi, M. (2005). Learning to dynamically manipulate: A table tennis robot controls a ball and rallies with a human being. *Advances in Robot Control*, pages 3137–341.

Møller, M. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525–533.

Mülling, K., Kober, J., Kroemer, O., and Peters, J. (2013). Learning to select and generalize striking movements in robot table tennis. *International Journal of Robotics Research*, 32(3):263–279.

Mülling, K., Kober, J., and Peters, J. (2011). A biomimetic approach to robot table tennis. *Adaptive Behavior*, 19(5):359–376.

Parker, S. K., Bindl, U. K., and Strauss, K. (2010). Making things happen: A model of proactive motivation. *Journal of Management*, 36(4):827–856.

Pentland, A. and Liu, A. (1999). Modeling and prediction of human behavior. *Neural Computation*, 11(1):229–242.

Peters, J. and Schaal, S. (2008). Natural actor-critic. *Neurocomputing*, 71(7):1180–1190.

Pezzulo, G., Butz, M. V., Castelfranchi, C., and Falcone, R. (2008). *The challenge of anticipation: A unifying framework for the analysis and design of artificial cognitive systems*. Springer.

Pistohl, T., Ball, T., Schulze-Bonhage, A., Aertsen, A., and Mehring, C. (2008). Prediction of arm movement trajectories from ECoG-recordings in humans. *Journal of Neuroscience Methods*, 167(1):105–114.

Póczos, B., Abbasi-Yadkori, Y., Szepesvári, C., Greiner, R., and Sturtevant, N. (2009). Learning when to stop thinking and do something! In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 825–832. ACM.

Powers, R., Shoham, Y., and Vu, T. (2007). A general criterion and an algorithmic framework for learning in multi-agent systems. *Machine Learning*, 67(1):45–76.

Quiñonero-Candela, J., Girard, A., Larsen, J., and Rasmussen, C. (2003). Propagation of uncertainty in Bayesian kernel models-application to multiple-step ahead forecasting. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959.

Ramanantsoa, M. and Durey, A. (1994). Towards a stroke construction model. *International Journal of Table Tennis Science*, 2:97–114.

Rao, R., Shon, A., and Meltzoff, A. (2004). A Bayesian model of imitation in infants and robots. In *Proceedings of Imitation and Social Learning in Robots, Humans, and Animals*, pages 217–247.

Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rishel, R. and Helmes, K. (2006). A variational inequality sufficient condition for optimal stopping with application to an optimal stock selling problem. *SIAM Journal on Control and Optimization*, 45(2):580–598.

Rosen, R., Rosen, J., Kineman, J., and Nadin, M. (2012). *Anticipatory Systems: Philosophical, Mathematical, and Methodological Foundations*. IFSR International Series on Systems Science and Engineering. Springer.

Schaal, S. (2009). The SL simulation and real-time control software package. Technical report, University of Southern California.

Schölkopf, B. and Smola, A. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

Sebanz, N., Bekkering, H., and Knoblich, G. (2006). Joint action: bodies and minds moving together. *Trends in Cognitive Sciences*, 10(2):70–76.

Shiryaev, A. N. (2007). *Optimal stopping rules*, volume 8. Springer.

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R. (2013). Real-time Human Pose Recognition in Parts from Single Depth Images. *Communications of the ACM*, 56(1):116–124.

Simon, H. (1991). Bounded rationality and organizational learning. *Organization Science*, 2(1):125–134.

Simon, M. A. (1982). *Understanding Human Action: Social Explanation and the Vision of Social Science*. State University of New York Press.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian Processes using Pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT press.

Snelson, E. L. (2007). *Flexible and Efficient Gaussian Process Models for Machine Learning*. PhD thesis, University of London.

Spaan, M. T. and Vlassis, N. (2005). Planning with continuous actions in partially observable environments. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 3458–3463. IEEE.

Sutton, C. and McCallum, A. (2007). An introduction to conditional random fields for relational learning. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning*. MIT Press.

Sutton, R. and Barto, A. (1998). *Reinforcement learning: An introduction*. The MIT press.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057–1063.

Taylor, G. W., Hinton, G. E., and Roweis, S. T. (2011). Two Distributed-State Models For Generating High-Dimensional Time Series. *Journal of Machine Learning Research*, 12:1025–1068.

Thrun, S. (2000). Monte Carlo POMDPs. In *Advances in Neural Information Processing Systems 12*, pages 1064–1070. MIT Press.

Titsias, M. K. and Lawrence, N. D. (2010). Bayesian Gaussian process latent variable model. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, volume 9, pages 844–851.

Turner, R., Deisenroth, M., and Rasmussen, C. (2010). State-space inference and learning with Gaussian processes. In *Proceedings of International Conference on Artificial Intelligence and Statistics*.

Tzikas, D. G., Likas, A. C., and Galatsanos, N. P. (2008). The Variational Approximation for Bayesian Inference. *Signal Processing Magazine, IEEE*, 25(6):131–146.

Urtasun, R., Fleet, D. J., and Fua, P. (2006). 3D People Tracking with Gaussian Process Dynamical Models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 238–245.

Urtasun, R., Fleet, D. J., Geiger, A., Popović, J., Darrell, T. J., and Lawrence, N. D. (2008). Topologically-Constrained Latent Variable Models. In *Proceedings of the 25th international conference on Machine learning*, pages 1080–1087. ACM.

van der Maaten, L., Postma, E., and van den Herik, J. (2009). Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10:1–41.

Vasquez, D., Fraichard, T., Aycard, O., and Laugier, C. (2008). Intentional motion on-line learning and prediction. *Machine Vision and Applications*, 19(5):411–425.

Vasquez, D., Fraichard, T., and Laugier, C. (2009). Growing hidden Markov models: An incremental tool for learning and predicting human and vehicle motion. *International Journal of Robotics Research*, 28(11-12):1486–1506.

Veeraraghavan, A., Roy-Chowdhury, A. K., and Chellappa, R. (2005). Matching shape sequences in video with applications in human movement analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1896–1909.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305.

Wang, J. M., Fleet, D. J., and Hertzmann, A. (2007). Multifactor Gaussian Process Models for Style-Content Separation. In *Proceedings of the 24th International Conference on Machine Learning*, pages 975–982. ACM.

Wang, J. M., Fleet, D. J., and Hertzmann, A. (2008). Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):283–298.

Wang, Y., Won, K., Hsu, D., and Lee, W. (2012a). Monte Carlo Bayesian reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning*.

Wang, Z., Boularias, A., Mülling, K., and Peters, J. (2011a). Balancing safety and exploitability in opponent modeling. In *Proceedings of AAAI Conference on Artificial Intelligence*.

Wang, Z., Boularias, A., Mülling, K., Schölkopf, B., and Peters, J. (submitteda). Anticipatory action selection in human-robot table tennis. *Artificial Intelligence*.

Wang, Z., Deisenroth, M. P., Amor, H. B., Vogt, D., Schölkopf, B., and Peters, J. (2012b). Probabilistic Modeling of Human Movements for Intention Inference. In *Proceedings of Robotics: Science and Systems (R:SS)*.

Wang, Z., Deisenroth, M. P., Zhang, K., Boularias, A., Schölkopf, B., and Peters, J. (submittedb). Hierarchical gaussian process dynamics models for human movement analysis. *Journal of Machine Learning Research*.

Wang, Z., Lampert, C. H., Mulling, K., Scholkopf, B., and Peters, J. (2011b). Learning anticipation policies for robot table tennis. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 332–337.

Wang, Z., Mülling, K., Deisenroth, M. P., Amor, H. B., Vogt, D., Schölkopf, B., and Peters, J. (2013). Probabilistic Movement Modeling for Intention Inference in Human-Robot Interaction. *International Journal of Robotics Research*, 32(7):841–858.

Williams, A., Ward, P., Knowles, J., and Smeeton, N. (2002). Anticipation skill in a real-world task: Measurement, training, and transfer in tennis. *Journal of Experimental Psychology*, 8(4):259.

Williams, B. H., Toussaint, M., and Storkey, A. J. (2006). Extracting motion primitives from natural handwriting data. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 634–643.

Williams, B. H., Toussaint, M., and Storkey, A. J. (2008). Modelling motion primitives and their timing in biologically executed movements. In *Advances in Neural Information Processing Systems 20*, pages 1609–1616.

Yang, P., Xu, D., Wang, H., and Zhang, Z. (2010). Control system design for a 5-DOF table tennis robot. In *Proceedings of International Conference on Control Automation Robotics and Vision*, pages 1731–1735.

Yao, A., Gall, J., Gool, L. V., and Urtasun, R. (2011). Learning Probabilistic Non-Linear Latent Variable Models for Tracking Complex Activities. In *Advances in Neural Information Processing Systems 24*, pages 1359–1367.

Zhou, E. (2013). Optimal stopping under partial observation: Near-value iteration. *IEEE Transactions on Automatic Control*, 58(2):500–506.

Ziebart, B., Dey, A., and Bagnell, J. (2012). Probabilistic pointing target prediction via inverse optimal control. In *Proceedings of ACM International Conference on Intelligent User Interfaces*.

Ziebart, B., Maas, A., Bagnell, J., and Dey, A. (2008). Maximum entropy inverse reinforcement learning. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 1433–1438.

Ziebart, B. D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J. A., Hebert, M., Dey, A. K., and Srinivasa, S. (2009). Planning-based prediction for pedestrians. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*.

List of Figures

List of Tables

List of Algorithms

Curriculum Vitae

Educational Background

2009–2013	**Ph.D. Student** Technische Universität Darmstadt, Darmstadt, Germany and Max Planck Institute, Tübingen, Germany
2007–2009	**Master's Degree in Computer Science and Technology** Tsinghua University, Beijing, China
2003–2007	**Bachelor's Degree in Computer Science and Technology** Tsinghua University, Beijing, China

Work Experience

2008	**Research Intern** Google China
2007	**Research Intern** Microsoft Research Asia
2004–2005	**Scientific Committee Member** National Olympiad in Informatics, China

Selected Awards

2009	Excellent Master Thesis Award of Tsinghua University
2004	Second place, ACM International Collegiate Programming Contest Beijing Site
2001, 2002	First Prize twice, National Olympiad in Informatics