Artificial Intelligence 247 (2017) 399–414

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Anticipatory action selection for human–robot table tennis

Zhikun Wang [a,b,*], Abdeslam Boularias [c], Katharina Mülling [c], Bernhard Schölkopf [a], Jan Peters [a,b]

[a] MPI for Intelligent Systems, Spemannstr. 38, 72076 Tübingen, Germany
[b] TU Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany
[c] Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Received in revised form 26 November 2014
Accepted 29 November 2014
Available online 4 December 2014

Keywords:
Anticipation
Intention-driven dynamics model
Partially observable Markov decision process
Policy iteration
Monte-Carlo planning

ABSTRACT

Anticipation can enhance the capability of a robot in its interaction with humans, where the robot predicts the humans' intention for selecting its own action. We present a novel framework of anticipatory action selection for human–robot interaction, which is capable to handle nonlinear and stochastic human behaviors such as table tennis strokes and allows the robot to choose the optimal action based on prediction of the human partner's intention with uncertainty. The presented framework is generic and can be used in many human–robot interaction scenarios, for example, in navigation and human–robot co-manipulation. In this article, we conduct a case study on human–robot table tennis. Due to the limited amount of time for executing hitting movements, a robot usually needs to initiate its hitting movement before the opponent hits the ball, which requires the robot to be anticipatory based on visual observation of the opponent's movement. Previous work on Intention-Driven Dynamics Models (IDDM) allowed the robot to predict the intended target of the opponent. In this article, we address the problem of action selection and optimal timing for initiating a chosen action by formulating the anticipatory action selection as a Partially Observable Markov Decision Process (POMDP), where the transition and observation are modeled by the IDDM framework. We present two approaches to anticipatory action selection based on the POMDP formulation, i.e., a model-free policy learning method based on Least-Squares Policy Iteration (LSPI) that employs the IDDM for belief updates, and a model-based Monte-Carlo Planning (MCP) method, which benefits from the transition and observation model by the IDDM. Experimental results using real data in a simulated environment show the importance of anticipatory action selection, and that POMDPs are suitable to formulate the anticipatory action selection problem by taking into account the uncertainties in prediction. We also show that existing algorithms for POMDPs, such as LSPI and MCP, can be applied to substantially improve the robot's performance in its interaction with humans.

© 2014 Elsevier B.V. All rights reserved.

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 [38]. Such ability of anticipation is also crucial in Human–Robot Interaction (HRI), where an anticipatory robot needs to coordinate for its actions while inter-

* Corresponding author.
E-mail address: zhikun.wang@gmail.com (Z. Wang).

http://dx.doi.org/10.1016/j.artint.2014.11.007
0004-3702/© 2014 Elsevier B.V. All rights reserved.

acting with humans [18], 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" [36]. Predicting the underlying intention and the future actions of the human partners has been extensively studied in robotics (e.g. [23,17,22,53,5]). This prediction capability offers the promise of anticipatory robots that incorporate *anticipatory action selection* in their planning.

In this article, we focus on the anticipatory action selection based on a prediction model. 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 movements, is obtained. However, waiting for a confident prediction causes delayed action selection and reduces the available time for the robot to execute its action [53]. 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 approaches to efficiently choose the robot's optimal action and decide the timing to initiate action. The framework is based on Intention-Driven Dynamics Model (IDDM), a nonparametric Bayesian model for human movements that allows to infer the intention of human partner while he is executing his action [53].

The presented framework is generic and can be used in many human–robot interaction scenarios. A typical application would be one where the robot observes the actions of a human, makes predictions about the human's intention with the IDDM, and reacts accordingly by either waiting for more observations or by executing a physical action. For example, a robot that navigates in an indoor environment alongside humans needs to constantly predict their trajectories in order to avoid collisions and to optimize its path [56]. In such environments, humans tend to walk to a target that belongs to a small set of specific predefined targets such as doors, chairs, stairs, etc. Therefore, the human's intention can be discretized and IDDM can be used to compute a belief on where the human is heading based on the observed images and on the images used for training. The robot can either adjust its path or slow down and wait for more observations, by using the Monte Carlo POMDP framework to forecast the effect of each decision. It is interesting to note here that humans use seemingly similar mechanisms of stopping, or slowing down, and belief updating in order to avoid collisions with pedestrians.

Another example is the problem of cooperative human–robot co-manipulation that is receiving increasing attention in the domain of intelligent prosthetics. Modern prosthetics contain semi-autonomous robotic joints that react to neurological signals [48]. These signals are high-dimensional and inherently noised. One can use IDDM to learn to predict the joint movement intended by the human user and to execute it. However, before executing the joint movement, it is also important to wait until the system is sufficiently certain that the human wants that movement. This is also a clearly optimal stopping movement that can be casted in the proposed framework [40]. Maybe a better example of human–robot co-manipulation is the scenario where the human uses a separate, fully autonomous, robotic arm for lifting objects [20]. In this case, the human simply reaches to an object, the robot perceives the human's arm trajectory and predicts the intended object and reaches to it. During the lifting action, the robot also needs to predict the direction that the human is trying to follow for moving the object, by using the force and torque feedback, as well as observed trajectories with IDDM. This is also an anticipatory action selection problem that can be solved in the proposed framework.

In this article, 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 [52]. This choice is mainly motivated by the need of fast perception and decision-making in table tennis, in addition to the high level of noise encountered in this type of applications.

1.1. Anticipation in human–robot table tennis

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 [1,12], fast movement generation [3,29], learning [25,27,28], and many other research problems in robotics. Inspired by the fact that human players reconstruct and improve their dynamics during matches [39], Mülling et al. [28] 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 often needs a repertoire of actions to cover the potential hitting points in its entire accessible workspace. For example, the biomimetic robot player [29] used in this article employs three actions, namely, forehand, default, and backhand hitting movements, as shown in Fig. 6. Each hitting movement is parametrized by duration, amplitude, and final goal position. Although 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 Fig. 1, which consist of four stages [34]. 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 shortly after the ball bounces of the table and the human has enough information to decide for a hitting state. The racket moves towards this hitting state and hits the ball at the end of the hitting

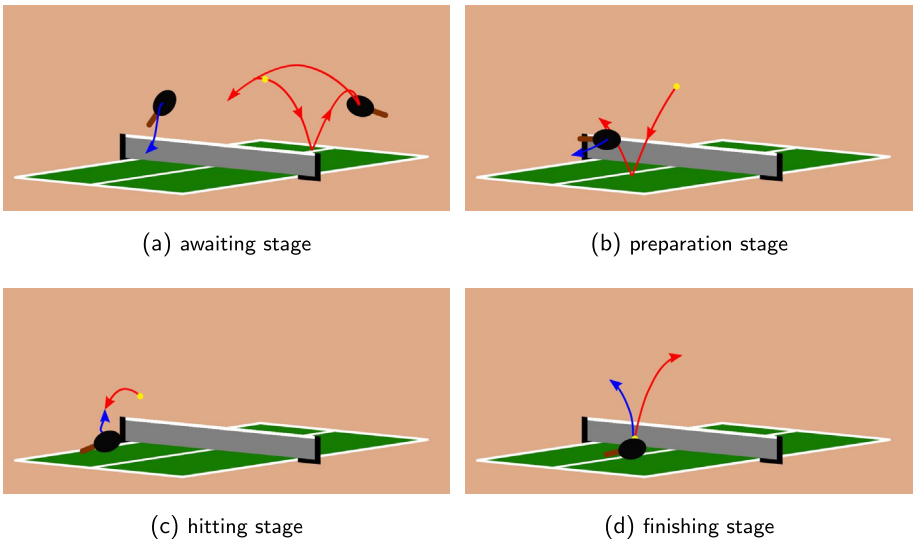


Fig. 1. The four stages of a typical table tennis ball rally, where the trajectories in red represent the ball trajectories and the trajectories in blue represent the racket's movements of the robot player. Figure adapted from Mülling et al. [29]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

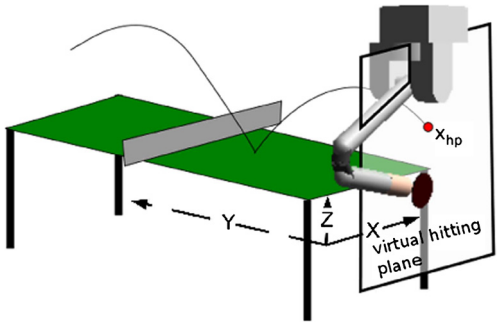


Fig. 2. Demonstration of the robot's hitting plane and the human opponent's target. The hitting point is the intersection x_{hp} of the coming ball's trajectory and the virtual hitting plane 80 cm behind the table, which is considered the *target* where the human opponent intents to shoot the ball. Our goal is to select an action according to the prediction of the target x_{hp}. Figure adapted from Mülling et al. [29].

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 80 ms. Even against a slower opponent, which allows less than 300 ms for the robot's hitting movement, this short time often does not suffice for initializing the racket to the preparation position and then moving it to reach the hitting state. Therefore, many desired hitting movements are not feasible due to time limits; the robot needs to initiate its hitting movement during the awaiting stage, that is, before the human opponent returns the ball.

To gain sufficient time for initializing and executing a hitting movement, the robot needs to be anticipatory about its human opponent's intention, which human players rely heavily on [2]. The prediction of the human partner's intention can be realized by modeling how the intention directs the dynamics of hitting movement [53]. 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 intends to shoot the ball [53], as shown in Fig. 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 Fig. 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 [53]. 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.

1.2. Related work and contributions

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 [30].

Online learning of intentional motion patterns and prediction of intentions based on HMMs was proposed by Vasquez et al. [46], which allows efficient inference in real time. The HMM can be learned incrementally to cope with new motion patterns in parallel with prediction [47].

Anticipation is important in many human–robot interaction scenarios (e.g., [56,10,51]). Decision making can also be achieved jointly with intention inference, in scenarios such as autonomous driving [5], control [17], or navigation in human crowds [22]. For example, when the state space is finite, the problem can be formulated as a partially observable Markov decision process [23] and solved efficiently [49]. The issue of planning a robot's actions while predicting actions of humans has also been tackled in the literature of human-aware planning. For example, Cirillo et al. [8] presented a task planner for robots that recognizes activities of humans in a shared living space (e.g. kitchen). In the presented system, an HMM is used for activity recognition, based on partial observations. The planner computes a probability distribution on the actions of the humans, and finds the best robot plan accordingly. The planner also adjusts to detected changes. Compared to our work, the aforementioned work is based on the same concept of combining predictions of human behavior with planning. Our work is however tailored to predict human motions and trajectories, instead of general plans or activities. Sisbot et al. [42] also proposed a human-aware motion planner. The proposed system allows robots to take into account human partners by reasoning about their accessibility, vision field and preferences. The proposed system did not include, however, a dynamical model of human movements.

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., [40]). 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 [33,14,11], using reinforcement learning to obtain an optimal stopping policy. The optimal stopping problem has also been studied under partial observation [55], and in many applications, such as quality control [21] and finance [9,35].

POMDPs have been used to formulate planning for human–robot interaction, for example, to assist elderly individuals with mild cognitive and physical impairments [32,6]. Previous work in human–robot interaction focused on POMDPs with representational restrictions, such as finite states [16] or time-aggregated state space [6,7]. Such restrictions make the model difficult to represent complicated and nonlinear human behaviors. This article presents a novel framework that formulates action selection in human–robot interaction with continuous-state POMDPs and handles non-linearity and stochasticity in human movements with Bayesian nonparametric models. This formulation is sufficiently expressive for complicated human movements such as table tennis strokes.

Human–robot table tennis has been used as a benchmark task for high-speed vision [1,12], fast movement generation [3, 29], learning [25,27,28], and many other research problems in robotics. However, previous work in robot table tennis focused on reactive action planning solely based on the ball's trajectory. This article presents a novel proactive planning framework by modeling and predicting the behavior of human opponent, such that the robot can start planning even before the human has hit the ball. To the best of our knowledge, this is the first work on human–robot table tennis that makes use of perceived human movements.

In addition, this article substantially extends previous work on the Intention-Driven Dynamics Models (IDDM), which was originally a dynamics model for intention inference and for predicting future observations [53]. We propose an original formulation of the action selection problem in robot table tennis as POMDPs, wherein a Monte-Carlo planning algorithm that makes use of the IDDM was presented to address the trade-off between prediction accuracy and reaction delay. We show that this novel formulation leads to a significant improvement of the robot's performance.

At a more general level, the most important contribution of this paper to Artificial Intelligence is presenting a well developed study on the combination of anticipation and planning. The anticipation module forecasts future states of a dynamical system based on past observations. Often, the dynamical system corresponds to human actions and intentions that are complex and difficult to model. Therefore, the anticipation module is typically learned from examples and used as a black-box by the planning module. The planning module relates to the robot's actions. Starting from a given state, the planner inquirers the anticipation module about the probabilities of different states in the future, and chooses its actions accordingly. These two modules are complementary because they solve different problems. The two modules can be developed separately. For instance, the anticipation module might have been developed for analyzing table tennis games between human players, while the planner was developed using a simulator or playing against a simple automatic ball launcher. Yet, the two modules can be combined with a minimum effort. We believe that this type of anticipation-planning combinations will play a crucial role in the development of future intelligent robots, as robots are moving out of their traditional controlled environments and increasingly interacting with humans.

The remainder of this article is organized as follows. We formulate the anticipatory action selection using POMDPs, and present approaches to policy learning and decision making in Section 2. In Section 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 contributions in Section 4.

2. Anticipatory action selection

The essence of anticipatory action selection is decision making given a time series of predictions, where two fundamental issues have to be addressed. First, uncertainties in the prediction and outcome need to be considered. For example in the robot table tennis setup, the uncertainty in prediction is mainly due to the fact that the opponent may still change the target before his 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.

Partially Observable Markov Decision Processes (POMDPs) are suitable to model the uncertainties in prediction and outcome and formulate anticipatory action selection as a Markov decision process under uncertainty. 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-Squares Policy Iteration (LSPI) for policy learning. However, 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, where actions are chosen according to the value function estimated using the Monte-Carlo method [45]. The considered optimal stopping problem is a special case of POMDPs; only one action, i.e., waiting, needs to be explored. We applied upper confidence bounds [4] to avoid unnecessary exploration of the waiting action and effectively prune search trees. The resulting algorithm is equivalent to a tailored POMCP [41] algorithm for optimal stopping.

2.1. Optimal stopping under partial observability

The anticipatory action selection is a generalization of optimal stopping under partial observation [26], and, similarly, can be formulated as a POMDP. A discrete-time POMDP is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{Z}, \mathcal{P}, \Omega, \mathcal{R})$, where $\mathcal{S}$ is a state space, $\mathcal{A} = \{0, \ldots, n\}$ is a set of actions, and $\mathcal{Z}$ is a observation space. The states evolve following a Markov transition model, governed by $\mathcal{P}$, where $\mathcal{P}(\mathbf{s}'|\mathbf{s}, a)$ 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 $\mathcal{R}(\mathbf{s}, a)$ represents the expected immediate reward obtained for taking action a in state $\mathbf{s}$.

The set of actions $\mathcal{A}$ consists of a waiting action $a = 0$ and a set of stopping actions $a \in \mathcal{A} \setminus \{0\}$. 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 $\mathbf{z} \in \mathcal{Z}$ 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 with indefinite horizon [15], given by

$$J = \mathbb{E}\big[r(\mathbf{s}_T, a_T)\big], \tag{1}$$

where the variable T denotes the stopping time, and a reward is obtained only by taking a stopping action a_T. Note that, generally, the stopping time T is unknown and determined by when a stopping action is taken. However, in many scenarios, there exists a maximum length of episode. For example in the case of human–robot table tennis, the robot's hitting point can usually be precisely predicted 120 ms after the human player hits the ball [53], and, thus, an optimal policy would stop waiting and initiate a hitting movement. In practice, we enforce that the robot takes a stopping action when the robot's hitting point can be reliably predicted by the vision system.

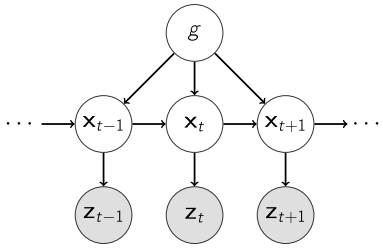


Fig. 3. The graphical model of the IDDM in an online manner, where we denote the intended target by g, state by $\mathbf{x}_t$, and observation by $\mathbf{z}_t$. The proposed model explicitly incorporates the intention as an input to the transition function [53]. Here, we use gray nodes for the observed variables and white nodes for the latent variables.

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). 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 Fig. 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 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

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

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 Fig. 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

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t) + \mathbf{n}_{z,t}, \quad \mathbf{n}_{z,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_z), \tag{3}$$

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 $\mathbf{Y}_z$.

The IDDM is a special case of the Hierarchical Gaussian Process Dynamics Model [50], where the intention is assumed to be time-invariant when learning the model. Nevertheless, this assumption does not limit the model's capability to infer time-varying intention. In fact, the online inference method [53] is able to deal with the change of intention. For more technical details of learning and approximate inference, we refer the reader to Wang et al. [53] and Wang [50].

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. [53] 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. Fig. 4 shows 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.

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 $\mathcal{P}(\boldsymbol{\theta}'|\boldsymbol{\theta}, a)$ 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 [43], 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

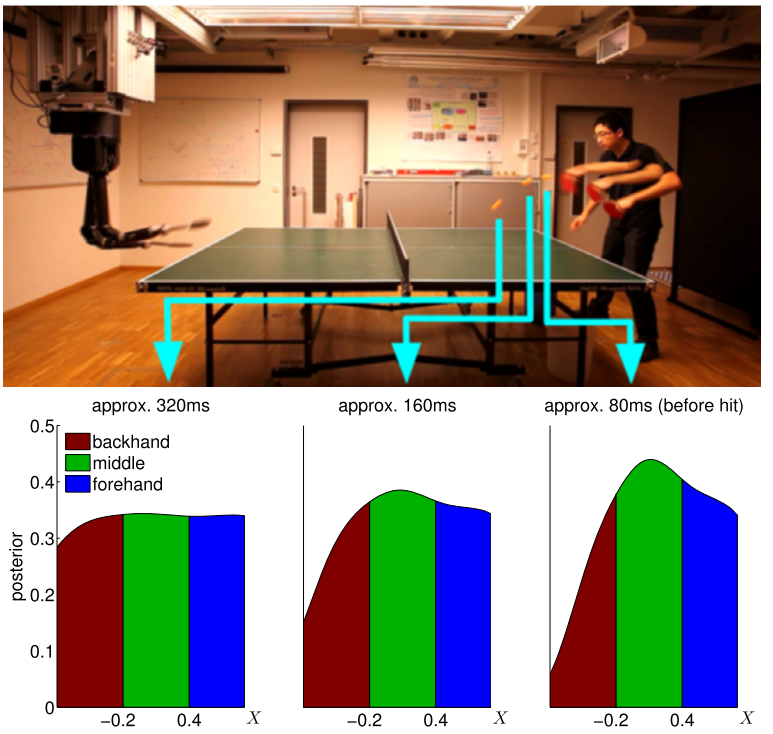


Fig. 4. Bar plots show the distribution of the target (X coordinate) at approximately 320 ms, 160 ms, and 80 ms before the player hits the ball. The prediction became increasingly confident as the player finishes the hitting movement, and the robot later chose the default hitting movement accordingly. Figure adapted from Wang et al. [53].

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

$$Q^{\pi}(\boldsymbol{\theta}, a = \text{waiting}) = \int \mathcal{P}(\boldsymbol{\theta}'|\boldsymbol{\theta}, a = \text{waiting}) \int \pi(a'|\boldsymbol{\theta}') Q^{\pi}(\boldsymbol{\theta}', a') da' d\boldsymbol{\theta}', \tag{4}$$

for the waiting action, and

$$\forall a \neq 0, \quad Q^{\pi}(\boldsymbol{\theta}, a) = \mathcal{R}(\boldsymbol{\theta}, a), \tag{5}$$

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

$$Q^{\pi}(\boldsymbol{\theta}, a) \approx \hat{Q}^{\pi}(\boldsymbol{\theta}, a; \mathbf{w}) = \boldsymbol{\phi}(\boldsymbol{\theta}, a)^T \mathbf{w}, \tag{6}$$

with parameters $\mathbf{w}$. Taking into account these factors, we choose the Least-Squares Policy Iteration (LSPI) algorithm [24], 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. (4) is intractable due to the unknown transition model $\mathcal{P}$, 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 $\mathcal{D} = \{(\boldsymbol{\theta}_i, a_i, r_i, \boldsymbol{\theta}'_i) | i = 1, \ldots, L\}$, where $\boldsymbol{\theta}_i$ is the output of the online inference algorithm [53]. Given a policy π, the estimates

$$\hat{\mathbf{A}} = \frac{1}{L} \sum_{l=1}^{L} \boldsymbol{\phi}(\boldsymbol{\theta}_i, a_i) \left(\boldsymbol{\phi}(\boldsymbol{\theta}_i, a_i) - \boldsymbol{\phi}(\boldsymbol{\theta}'_i, \pi(\boldsymbol{\theta}'_i)) \right)^T, \tag{7}$$

$$\hat{\mathbf{b}} = \frac{1}{L} \sum_{l=1}^{L} \boldsymbol{\phi}(\boldsymbol{\theta}_i, a_i) r_i, \tag{8}$$

Algorithm 1: The LSPI algorithm, which iteratively updates the approximated value function and the optimal policy.

Input : Obtained samples $\mathcal{D}$
Output: Approximation parameters $\mathbf{w}$
$\mathbf{w}' \leftarrow 0$;
repeat
  $\mathbf{w} \leftarrow \mathbf{w}'$;
  foreach $(\boldsymbol{\theta}_i, a_i, r_i, \boldsymbol{\theta}_i') \in \mathcal{D}$ **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

$$\mathbf{w} = \left(\hat{\mathbf{A}} + \delta^2\mathbf{I}\right)^{-1}\hat{\mathbf{b}}, \tag{9}$$

where the sufficient small δ^2 is used to avoid numerical error in the inversion of $\hat{\mathbf{A}}$ [24].

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

$$\pi'(\boldsymbol{\theta}) = \arg\max_a \hat{Q}^{\pi}(\boldsymbol{\theta}, a). \tag{10}$$

We can obtain the optimal policy by iteratively executing the policy evaluation and improvement steps, as summarized in Algorithm 1.

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 belief $\boldsymbol{\theta}_t$ given a history of observations $\mathbf{z}_{1:t}$. Besides the capability of belief update, the IDDM in fact provides a transition model in 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 an optimal policy. The considered optimal stopping problem is a special case of POMDPs; there is only one single waiting action that needs to be explored. As a stopping action terminates the decision process immediately, the value function for the stopping actions $\forall a \in \mathcal{A} \setminus \{0\} : Q^{\pi}(\boldsymbol{\theta}, a) = Q(\boldsymbol{\theta}, a)$ holds for any belief state $\boldsymbol{\theta}$ and any policy π. It is sensible to estimate the value function for stopping actions offline, as running simulation repeatedly, especially with real-time simulator (e.g., simulated robot table tennis), during planning is expensive. We, hence, reuse the estimated value function for stopping actions by LSPI, and focus on the value function for the waiting action $Q(\boldsymbol{\theta}_t, a_t = \text{waiting})$, given by the expected future reward

$$Q(\boldsymbol{\theta}_t, a_t = \text{waiting}) = \mathbb{E}\Big[\max_{a_{t+1}} Q(\boldsymbol{\theta}_{t+1}, a_{t+1})\Big] \tag{11}$$

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

The value function $Q(\boldsymbol{\theta}_t, a_t = \text{waiting})$ measures the expected reward if the agent chooses to wait for more observation. While computing the exact expectation is intractable, the value function can be estimated using Monte-Carlo approximation [45], where we replace the expectation operator by an empirical average over sampled belief states. 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 $Q(\boldsymbol{\theta}_{t+1}, a_{t+1} = \text{waiting})$ 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

$$Q(\boldsymbol{\theta}_t, a_t \text{ waiting}) = \mathbb{E}\Big[\max_{a_{t+1} \neq 0} \hat{Q}(\boldsymbol{\theta}_{t+1}, a_{t+1}; \mathbf{w})\Big]. \tag{12}$$

This estimation can be achieved by using particle projection routine [45], as described in Algorithm 2.

Algorithm 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 $Q(\boldsymbol{\theta}_t, a_t = \text{waiting})$
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 $Q(\boldsymbol{\theta}_t, a_t = \text{waiting}) \approx \frac{1}{I} \sum_{r^i \in \Phi} r^i$;

Algorithm 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 α
Output: Action a_t
Collection of sampled rewards $\Phi = \emptyset$;
for $i \leftarrow 1, \ldots, I$ **do**
  Sample current state and intention $\mathbf{s}_t = [\mathbf{x}_t, g]$;
  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 Φ;
    if $U < \max_{a \neq 0} Q(\boldsymbol{\theta}_t, a)$ **then**
      Return the optimal stopping action $a_t = \arg\max_{a \neq 0} Q(\boldsymbol{\theta}_t, a)$;
    Compute lower confidence bound L given sample Φ ;
    if $L > \max_{a \neq 0} Q(\boldsymbol{\theta}_t, a)$ **then**
      Return the waiting action $a_t = 0$;
  Expected reward for waiting $Q(\boldsymbol{\theta}_t, a_t = \text{waiting})$ is the mean of sampled rewards in Φ ;
  Return the optimal action $a_t = \arg\max_a Q(\boldsymbol{\theta}_t, a)$;

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

$$a_t = \arg\max_{a \in \mathcal{A}} Q(\boldsymbol{\theta}_t, a), \tag{13}$$

where one only needs to know if the waiting action leads to higher expected reward than all the stopping actions, rather than the actual expected reward of waiting. We can terminate the 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 upper confidence bound algorithms [4], 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 $\Phi = \{r^1, \ldots, r^n\}$, the confidence interval given a confidence level α is

$$\left[\bar{r} - \frac{cs}{\sqrt{n}}, \bar{r} + \frac{cs}{\sqrt{n}}\right], \tag{14}$$

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 3 describes the resulting Monte-Carlo planning algorithm. Note that this pruning algorithm can be seen as a Partially Observable Monte-Carlo Planning (POMCP) algorithm ([41] Algorithm 1) tailored for optimal stopping problems, where the search procedure can be even more effectively pruned before reaching timeout. The differences between the presented algorithm and POMCP are twofold. First, simulation is often computationally demanding in many robotics applications; it is sensible to estimate the value function of stopping actions offline, as stopping actions terminate the POMDP immediately and no exploration is needed. Therefore, the presented

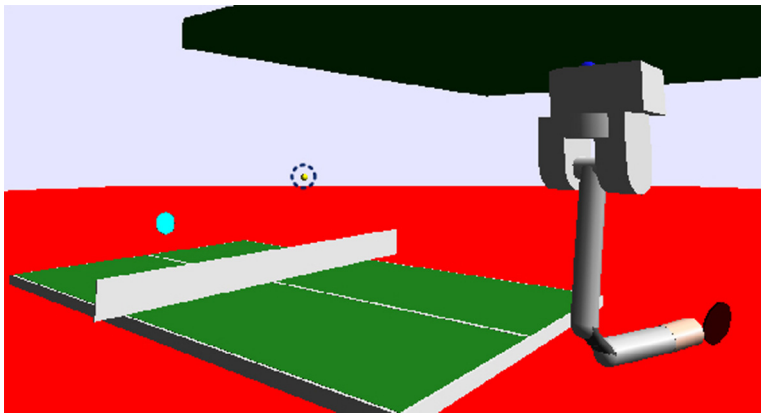


Fig. 5. The SL simulated environment with the state of the robot arm and the information obtained from the vision system, including states of the opponent's racket the ball.

algorithm estimates the value of stopping actions using all samples obtained from simulation, whereas the vanilla POMCP algorithm explores the stopping action during planning, which is not realistic for real-time simulators such as simulated robot table tennis [29]. Second, only the waiting action needs to be explored in each step, and the search procedure in POMCP can stop before timeout when either (1) the upper confidence bound of expected reward for waiting is lower than expected reward for any stopping action or (2) the lower confidence bound is higher than expected reward for all stopping actions.

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.

2.5. Basis functions

The presented LSPI and MCP methods both employ function approximation, relying 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 Fig. 4.

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

$$\boldsymbol{\phi}(\boldsymbol{\theta}, a) = \left[\delta_{a,0}\boldsymbol{\phi}'(\boldsymbol{\theta})^T, \delta_{a,1}\boldsymbol{\phi}'(\boldsymbol{\theta})^T, \ldots, \delta_{a,|\mathcal{A}|}\boldsymbol{\phi}'(\boldsymbol{\theta})^T\right]^T, \tag{15}$$

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

$$\boldsymbol{\phi}'(\boldsymbol{\theta}) = \left[\exp\left\{-\eta\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_1\|^2\right\}, \ldots, \exp\left\{-\eta\|\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_K\|^2\right\}\right]^T \tag{16}$$

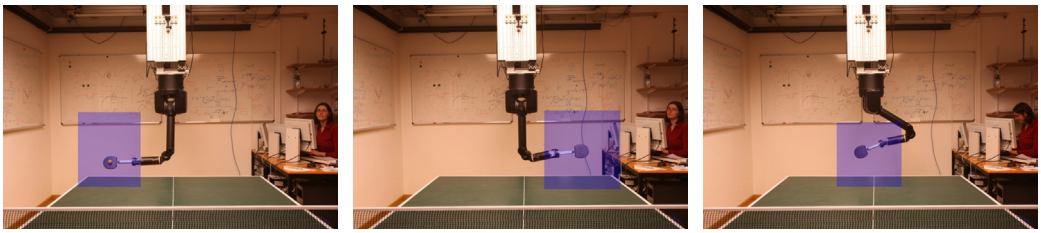
for the belief states.

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 [29], 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 [28] as well as many of the recent table tennis learning systems [19,54,25].

3.1. Robot player

To quantitatively evaluate the performance of the proposed methods in terms of success rates, we used the SL framework [37], which consisted of a real-world setup and a sufficiently realistic simulation. The setup included a Barrett WAM^{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 [52], recording the players' movement and ball's trajectory. The collected data was used by the SL simulation for the following experiments, as demonstrated in Fig. 5. Previ-



(a) forehand (b) backhand (c) default

Fig. 6. 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.

ous work showed that the SL framework is sufficiently realistic to simulate the robot table tennis setting,[1] and that similar performance can be expected when the real robot is used [29,28,53].

In the considered setting, the robot always hits the ball on a virtual hitting plane 80 cm behind the table, as shown in Fig. 2. We defined the human's intended target as the intersection of the returned ball's trajectory with the robot's virtual hitting plane. As the x-coordinate (see Fig. 2) was most important for choosing the type of hitting movements [53], 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 ± 1.2 m from the robot's base (the table is 1.52 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 Fig. 6. Hence, the action set $\mathcal{A}$ 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 choose a hitting type, i.e., 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 [53].

When the robot's hitting point can be precisely predicted, an optimal policy would take a stopping action to maximize the time for executing planned movement. Therefore, we enforce that the robot takes a stopping action when the robot's hitting point can be reliably predicted by the vision system, which typically happens within 120 ms after the human player returns the ball [53].

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 player reached the robot's hitting plane (see Fig. 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 Fig. 6, and evaluated the performance of the policy learning algorithm on the remaining 207 trials. For the basis functions used for function approximation in Eq. (16), we chose 100 centers in the experiments. Note that the IDDM model had been trained with 100 additional trials [53], and this set of data was not used in the experiments in this article.

3.2. Experimental results

We evaluated the algorithms using ten-fold cross-validation. We carried out 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 $\mathcal{D} = \{(\boldsymbol{\theta}_i, a_i, r_i, \boldsymbol{\theta}'_i) | i = 1, \ldots, L\}$ 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 $\mathcal{D}$ using Algorithm 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 3 led to successful returns for the robots in 153.3 ± 0.6 times, an average success rate of 74%.

To demonstrate the importance of action selection, we evaluated a baseline that exclusively used a single action, i.e., the robot always use the same 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. Fig. 7 showed the number of successful returns for using each action initiated at 240 ms 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

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

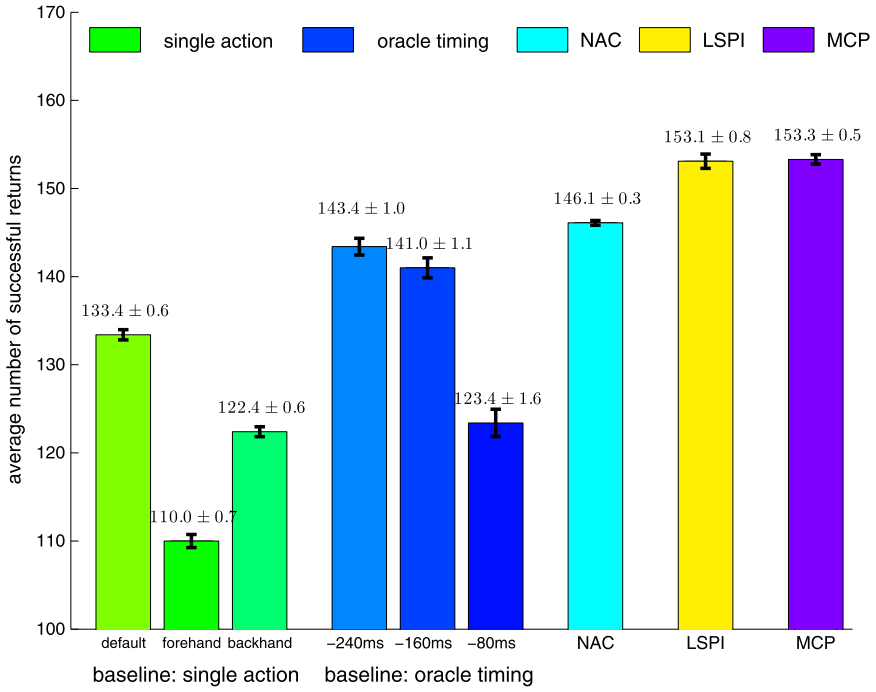


Fig. 7. Performance of different methods evaluated in ten repetitions on the test data with 207 valid trials, based on enumerated training episodes. For each method, we evaluated the performance in terms of averaged number of successful returns, using cross-validation in each repetition. The first group 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 (240 ms, 160 ms, and 80 ms) 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.

MCP algorithm significantly improved the performance of the robot player (p-value 0.03, using chi-squared test) by learning which action should be executed.

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. This hypothetical setting allows to demonstrate the importance of choosing an optimal stopping time. 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 240 ms before the opponent hits the ball, 141.0 ± 1.1 times at 160 ms, and 123.4 ± 1.6 times at 80 ms, 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 outperformed this hypothetical baseline by learning when a chosen action should be initiated.

In addition, we consider a naive baseline where the action is chosen only based on the most likely prediction without considering predictive uncertainty, i.e., the belief state $\boldsymbol{\theta}$ is represented only by the mean of latent state and the most likely intention. We first consider the setting with oracle timing. The robot successfully returned the ball 131.0 ± 4.9, 128.8 ± 5.5, and $111.7.4 \pm 5.8$ times when the hitting movement was chosen at 240 ms, 160 ms, and 80 ms, respectively, before the opponent hits the ball. The performance is not better than simply using a single action, as it is crucial to take into account the uncertainty in prediction for action selection. We then consider the setting where the timing for action selection is decided by the learned policy using LSPI. The robot successfully returned the ball 127.2 ± 4.2 times. The significantly degenerated performance (p-value 0.006) indicates that the uncertainty in belief state is also crucial for optimal timing. The presented formulation with POMDPs in this article is suitable to represent such optimal stopping problems and propagate the uncertainty in prediction. We do not consider Monte-Carlo planning algorithm in this baseline setting, as the uncertainty in belief state are needed for sampling future states.

Furthermore, we also compared to policy gradient methods [44]. Specifically, we adopted episodic Natural Actor-Critic (NAC) algorithm [31] and formulated the policy by soft-max function [43]. 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 Fig. 7.

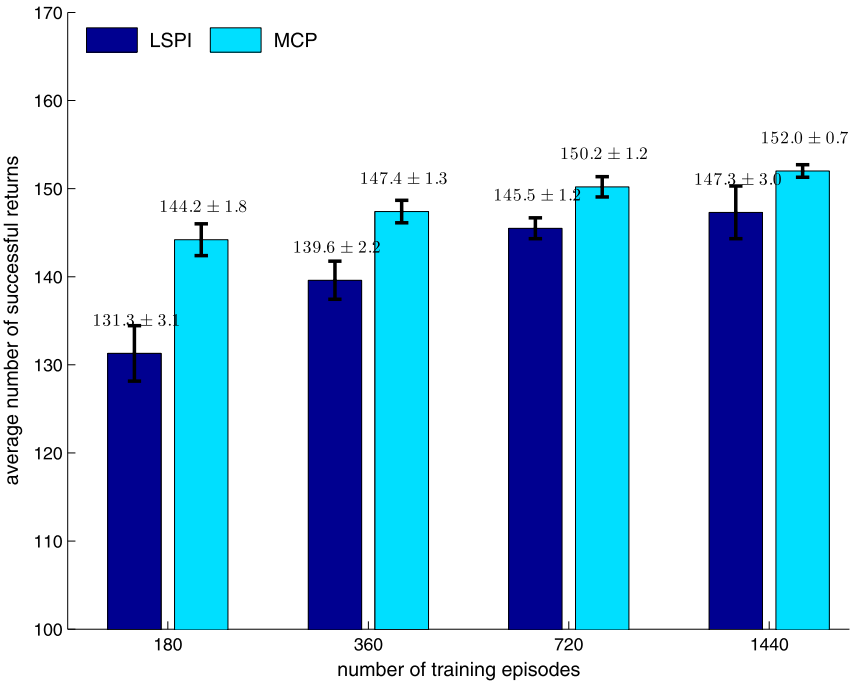


Fig. 8. Performance of the LSPI and MCP methods evaluated in ten repetitions on the test data with 207 valid trials, based on sampled episodes on the training data. The numbers on the X-axis showed the number of sampled episodes on the training data.

In the second experiment, we compared LSPI and MCP in terms of sample efficiency. We obtained samples $\mathcal{D}$ 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 Fig. 8 was obtained by sampling one episode for each trial in the training data. This amount of data was insufficient for 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 Fig. 8 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.

3.3. Discussion

We conducted a case study with a simplified human–robot table tennis scenario. We assume that the recruited non-professional players do not deliberately mislead the opponent by changing their intended target during the execution of stroke. More generally, IDDM assumes that the intention is time-invariant and the main driving factor when learning the dynamics model. For example, the speed and spin of the table tennis ball, which also affect the dynamics of strokes, are not explicitly considered in the model.[2] This assumption does not necessarily limit the capability of IDDM for intention inference [53] and planning. Moreover, this assumption can be further relaxed by taking into account other driving factors as exogenous variables in the dynamics model, leading to Hierarchical Gaussian Process Dynamics Models (H-GPDMs), discussed by [50]. Note that, however, the method is not applicable for adversarial planning, as the assumption of time-invariant intention is violated in adversarial scenarios. Incorporation of game-theoretic perspectives in the framework of H-GPDMs is a future direction of this work.

Despite the simplification, the case study clearly showed the importance of combining anticipation and planning and the feasibility of presented methods. We discuss implications of the experimental results in the general context of human–robot interaction as follows. *Action selection is important*. Robot players with action selection (baseline: oracle timing, NAC, LSPI, and MCP) outperformed that using a single action (baseline: single action), as shown in Fig. 7. *Anticipatory action selection substantially improves the robot's performance*. The baseline (oracle timing) in Fig. 7 shows that it is important for the robot to be anticipatory and to choose optimal action even before the human's intention can be accurately predicted. *Optimal*

[2] The speed and spin of the ball are implicitly modeled by the data-driven, non-parametric dynamics model.

stopping is essential in anticipatory action selection. Comparison between methods for optimal stopping (NAC, LSPI, and MCP) and baseline (oracle timing) shows that it is important to trade off between confidence in prediction and time for executing action. POMDPs are suitable formulation of such optimal stopping problems, and existing algorithms for POMDPs can be applied to solve the optimal stopping problem. *Uncertainties in prediction needs to be taken into account for optimal stopping*. The baseline that only used predictive mean of latent state and most likely intention has substantially degenerated performance, indicating that uncertainties in prediction are important and informative for optimal stopping. *Model-based planning can be more sample-efficient than model-free policy learning*. Comparison between LSPI (model-free policy learning) and MCP (model-based planning) in Fig. 8 shows that model-based planning tends to be more sample-efficient than model-free policy learning.

Besides the considered scenario, the method is generic and can be used in other human–robot interaction scenarios. Human–robot co-manipulation is another type of problems that can benefit from the proposed approach. In this problem, a robot assists a human in lifting and moving objects. Co-manipulation is needed in industrial settings where the objects are often too heavy, it is also increasingly used in robotic prosthetics where a robotic arm replaces a lost human arm. In these scenarios, the robot should predict which object among several ones the human is trying to reach and whether the human will need help. Upon becoming certain about the intended target, the robotic arm's trajectory should be planned and the robotic hand should start moving toward the intended object. Adapting our approach to this problem seems straightforward. We only need to re-define the intention $\mathbf{g}$ as the coordinates of the intended object in a three-dimensional space. Observations $\mathbf{z}_{1..t}$ can remain the observed trajectory of the human's arm. The robot decides to initiate a movement toward the intended object, to wait until it becomes more certain, or to do nothing if the object can be lifted by the human without any help. The presented method is applicable when the intention is multi-dimensional variables, in which case MCP is a preferred tool for solving POMDPs; it is non-trivial for LSPI to model and parametrize the distribution of multi-dimensional and potentially multi-modal intention.

Note that, however, the proposed method does not straightforwardly work for structured intention prediction such as predicting trajectories of cyclist. There are two potential approaches based on the proposed method for dealing with structured intention. Take the avoidance of a cyclist as an example. Rather than predicting the trajectory, we can formulate the intention as the direction where the cyclist wants to go in a short period of time. To take into account the intention change over time, one can model the dynamics of the cyclist with a H-GPDM [50], where the intention g_t follows a Markov chain. The other potential approach is to design a new covariance function $k_x(\cdot,\cdot)$ in the transition function in Eq. (2) for structured inputs [13], for example measuring the similarity of cyclist trajectories.

4. Conclusions

In this article, we introduced novel formulation and methods for anticipatory action selection for human–robot interaction, combining anticipation and planning. We formulated the anticipatory action selection as optimal stopping in a partially observable Markov decision processes. The presented formulation, using latent state variables and Bayesian nonparametric model, is expressive for nonlinear and stochastic human behaviors, such as table tennis strokes. We first presented a policy learning approach using the Least-Squares 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. The presented framework is generic and can be used in many other human–robot interaction scenarios, for example, in navigation and human–robot co-manipulation. A typical application would be one where the robot observes the actions of a human, makes predictions about the human's intention with the IDDM, and reacts accordingly by either waiting for more observations or by executing a physical action, where the uncertainties in prediction needs to be considered in deciding when and how to plan the robot's own actions.

In this article, we motivated and evaluated the proposed framework with human–robot table tennis games, where a bottleneck is the limited amount of time for the robot to execute a hitting movement. Movement initiation requires an early decision on the type of action, such as a forehand or backhand movement, before the opponent has hit the ball. Experimental results using real data and a simulated environment showed 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 concluded that anticipatory action selection substantially improves the robot's performance, and that uncertainties in prediction needs to be taken into account for optimal stopping. POMDPs had been shown to be suitable to formulate the anticipatory action selection problem by taking into account the uncertainties in prediction. We also showed that existing algorithms for POMDPs, such as LSPI and MCP, can be applied to substantially improve the robot's performance in its interaction with humans.

References

[1] L. Acosta, J. Rodrigo, J. Mendez, G. Marichal, M. Sigut, Ping-pong player prototype, IEEE Robot. Autom. Mag. 10 (4) (2003) 44–52.
[2] M. Alexander, A. Honish, Table tennis: a brief overview of biomechanical aspects of the game for coaches and players, Tech. report, Faculty of Kinesiology and Recreation Management, University of Manitoba, 2009, http://umanitoba.ca/faculties/kinrec/research/media/table_tennis.pdf.
[3] L. Ángel, J. Sebastián, R. Saltarén, R. Aracil, R. Gutiérrez, RoboTenis: design, dynamic modeling and preliminary control, in: Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2005, pp. 747–752.

[4] P. Auer, Using confidence bounds for exploitation–exploration trade-offs, J. Mach. Learn. Res. 3 (2003) 397–422.
[5] T. Bandyopadhyay, K.S. Won, E. Frazzoli, D. Hsu, W.S. Lee, D. Rus, Intention-aware motion planning, in: Algorithmic Foundations of Robotics X, in: Springer Tracts in Advanced Robotics, vol. 86, Springer, Berlin/Heidelberg, 2013, pp. 475–491.
[6] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie, A. Mihailidis, A decision-theoretic approach to task assistance for persons with dementia, in: Proceedings of the 19th International Joint Conference on Artificial intelligence, 2005, pp. 1293–1299.
[7] F. Broz, I. Nourbakhsh, R. Simmons, Planning for human–robot interaction in socially situated tasks, Int. J. Soc. Robot. 5 (2) (2013) 193–214.
[8] M. Cirillo, L. Karlsson, A. Saffiotti, Human-aware task planning: an application to mobile robots, ACM Trans. Intell. Syst. Technol. 1 (2) (2010) 15.
[9] J.-P. Décamps, T. Mariotti, S. Villeneuve, Investment timing under incomplete information, Math. Oper. Res. 30 (2) (2005) 472–500.
[10] A.D. Dragan, S.S. Srinivasa, Formalizing assistive teleoperation, in: Proceedings of Robotics: Science and Systems, 2012.
[11] G. Dulac-Arnold, L. Denoyer, P. Preux, P. Gallinari, Sequential approaches for learning datum-wise sparse representations, Mach. Learn. 89 (1–2) (2012) 87–122.
[12] H. Fässler, H. Beyer, J. Wen, A robot ping pong player: optimized mechanics, high pheromones 3d vision, and intelligent sensor control, Robotersysteme 6 (3) (1990) 161–170.
[13] T. Gärtner, A survey of kernels for structured data, ACM SIGKDD Explor. Newsl. 5 (1) (2003) 49–58.
[14] R. Gaudel, M. Sebag, Feature selection as a one-player game, in: Proceedings of the 27th International Conference on Machine Learning, 2010, pp. 359–366.
[15] E.A. Hansen, Indefinite-horizon POMDPs with action-based termination, in: Proceedings of the National Conference on Artificial Intelligence, vol. 22, AAAI Press/MIT Press, Menlo Park, CA/Cambridge, MA/London, 2007, p. 1237.
[16] E.A. Hansen, R. Zhou, Synthesis of hierarchical finite-state controllers for POMDPs, in: Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling, 2003, pp. 113–122.
[17] K. Hauser, Recognition, prediction, and planning for assisted teleoperation of freeform tasks, in: Proceedings of Robotics: Science and Systems, 2012.
[18] G. Hoffman, C. Breazeal, Cost-based anticipatory action selection for human–robot fluency, IEEE Trans. Robot. 23 (5) (2007) 952–961.
[19] Y. Huang, D. Xu, M. Tan, H. Su, Adding active learning to LWR for ping-pong playing robot, IEEE Trans. Control Syst. Technol. 21 (4) (2013) 1489–1494.
[20] N. Jarrassé, J. Paik, V. Pasqui, G. Morel, How can human motion prediction increase transparency?, in: IEEE International Conference on Robotics and Automation, IEEE, 2008, pp. 2134–2139.
[21] U. Jensen, G.-H. Hsu, Optimal stopping by means of point process observations with applications in reliability, Math. Oper. Res. 18 (3) (1993) 645–657.
[22] M. Kuderer, H. Kretzschmar, C. Sprunk, W. Burgard, Feature-based prediction of trajectories for socially compliant navigation, in: Proceedings of Robotics: Science and Systems, 2012.
[23] H. Kurniawati, Y. Du, D. Hsu, W. Lee, Motion planning under uncertainty for robotic tasks with long time horizons, Int. J. Robot. Res. 30 (3) (2011) 308–323.
[24] M. Lagoudakis, R. Parr, Least-squares policy iteration, J. Mach. Learn. Res. 4 (2003) 1107–1149.
[25] M. Matsushima, T. Hashimoto, M. Takeuchi, F. Miyazaki, A learning approach to robotic table tennis, IEEE Trans. Robot. 21 (4) (2005) 767–771.
[26] G. Mazziotto, Approximations of the optimal stopping problem in partial observation, J. Appl. Probab. 23 (1986) 341–354.
[27] F. Miyazaki, M. Matsushima, M. Takeuchi, Learning to dynamically manipulate: a table tennis robot controls a ball and rallies with a human being, in: Advances in Robot Control, 2005, pp. 3137–3341.
[28] K. Mülling, J. Kober, O. Kroemer, J. Peters, Learning to select and generalize striking movements in robot table tennis, Int. J. Robot. Res. 32 (3) (2013) 263–279.
[29] K. Mülling, J. Kober, J. Peters, A biomimetic approach to robot table tennis, Adapt. Behav. 19 (5) (2011) 359–376.
[30] A. Pentland, A. Liu, Modeling and prediction of human behavior, Neural Comput. 11 (1) (1999) 229–242.
[31] J. Peters, S. Schaal, Natural actor–critic, Neurocomputing 71 (7) (2008) 1180–1190.
[32] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, S. Thrun, Towards robotic assistants in nursing homes: challenges and results, Robot. Auton. Syst. 42 (3) (2003) 271–281.
[33] B. Póczos, Y. Abbasi-Yadkori, C. Szepesvári, R. Greiner, N. Sturtevant, Learning when to stop thinking and do something!, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 825–832.
[34] M. Ramanantsoa, A. Durey, Towards a stroke construction model, Int. J. Table Tennis Sci. 2 (1994) 97–114.
[35] R. Rishel, K. Helmes, A variational inequality sufficient condition for optimal stopping with application to an optimal stock selling problem, SIAM J. Control Optim. 45 (2) (2006) 580–598.
[36] R. Rosen, J. Rosen, J. Kineman, M. Nadin, Anticipatory Systems: Philosophical, Mathematical, and Methodological Foundations, IFSR International Series on Systems Science and Engineering, Springer, 2012.
[37] S. Schaal, The SL simulation and real-time control software package, Tech. rep., University of Southern California, 2009.
[38] N. Sebanz, H. Bekkering, G. Knoblich, Joint action: bodies and minds moving together, Trends Cogn. Sci. 10 (2) (2006) 70–76.
[39] C. Sève, J. Saury, J. Theureau, M. Durand, Activity organization and knowledge construction during competitive interaction in table tennis, Cogn. Syst. Res. 3 (3) (2002) 501–522.
[40] A.N. Shiryaev, Optimal Stopping Rules, vol. 8, Springer, 2007.
[41] D. Silver, J. Veness, Monte-Carlo planning in large POMDPs, in: Advances in Neural Information Processing Systems, vol. 23, 2010, pp. 2164–2172.
[42] E.A. Sisbot, L.F. Marin-Urias, R. Alami, T. Simeon, A human aware mobile robot motion planner, IEEE Trans. Robot. 23 (5) (2007) 874–883.
[43] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, The MIT Press, 1998.
[44] R.S. Sutton, D.A. McAllester, S.P. Singh, Y. Mansour, Policy gradient methods for reinforcement learning with function approximation, in: Advances in Neural Information Processing Systems, vol. 12, 1999, pp. 1057–1063.
[45] S. Thrun, Monte Carlo POMDPs, Advances in Neural Information Processing Systems, vol. 12, MIT Press, 2000, pp. 1064–1070.
[46] D. Vasquez, T. Fraichard, O. Aycard, C. Laugier, Intentional motion on-line learning and prediction, Mach. Vis. Appl. 19 (5) (2008) 411–425.
[47] D. Vasquez, T. Fraichard, C. Laugier, Growing hidden Markov models: an incremental tool for learning and predicting human and vehicle motion, Int. J. Robot. Res. 28 (11–12) (2009) 1486–1506.
[48] M. Velliste, S. Perel, M.C. Spalding, A.S. Whitford, A.B. Schwartz, Cortical control of a prosthetic arm for self-feeding, Nature 453 (7198) (2008) 1098–1101.
[49] Y. Wang, K. Won, D. Hsu, W. Lee, Monte Carlo Bayesian reinforcement learning, in: Proceedings of the 29th International Conference on Machine Learning, 2012, pp. 1135–1142.
[50] Z. Wang, Intention inference and decision making with hierarchical gaussian process dynamics models, Ph.D. thesis, TU Darmstadt, 2013, http://tuprints.ulb.tu-darmstadt.de/3617/.
[51] Z. Wang, M.P. Deisenroth, H.B. Amor, D. Vogt, B. Schölkopf, J. Peters, Probabilistic modeling of human movements for intention inference, in: Proceedings of Robotics: Science and Systems, 2012, (R:SS).
[52] Z. Wang, C.H. Lampert, K. Mulling, B. Scholkopf, J. Peters, Learning anticipation policies for robot table tennis, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2011, pp. 332–337.
[53] Z. Wang, K. Mülling, M.P. Deisenroth, H.B. Amor, D. Vogt, B. Schölkopf, J. Peters, Probabilistic movement modeling for intention inference in human–robot interaction, Int. J. Robot. Res. 32 (7) (2013) 841–858.

[54] P. Yang, D. Xu, H. Wang, Z. Zhang, Control system design for a 5-DOF table tennis robot, in: Proceedings of International Conference on Control Automation Robotics and Vision, 2010, pp. 1731–1735.
[55] E. Zhou, Optimal stopping under partial observation: near-value iteration, IEEE Trans. Autom. Control 58 (2) (2013) 500–506.
[56] B.D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J.A. Bagnell, M. Hebert, A.K. Dey, S. Srinivasa, Planning-based prediction for pedestrians, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2009, pp. 3931–3936.