Probabilistic Movement Modeling for Intention Inference in Human-Robot Interaction

Zhikun Wang[1,2], Katharina Mülling[1,2], Marc Peter Deisenroth[2],
Heni Ben Amor[2], David Vogt[3],
Bernhard Schölkopf[1], and Jan Peters[1,2]

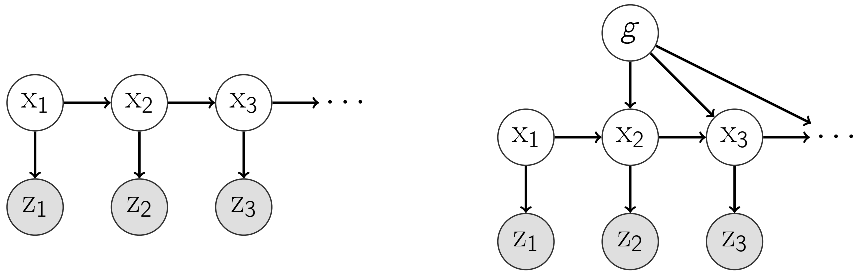
[1]Max Planck Institute for Intelligent Systems
Spemannstr. 38, 72076 Tübingen, Germany.
[2]Technische Universität Darmstadt
Hochschulstr. 10, 64289 Darmstadt, Germany.
[3]Technical University Bergakademie Freiberg
Bernhard-von-Cotta-Str. 2, 09596 Freiberg, Germany.

Abstract

Intention inference can be an essential step toward efficient human-robot interaction. For this purpose, we propose the Intention-Driven Dynamics Model (IDDM) to probabilistically model 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 used to evaluate the performance of our inference algorithm. In both intention inference tasks, the proposed algorithm achieves substantial improvements over support vector machines and Gaussian processes.

1 Introduction

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., 2011), which can enhance the robot's ability to interact with humans. However, effective perception alone may not be sufficient for Human-Robot



(a) GPDM

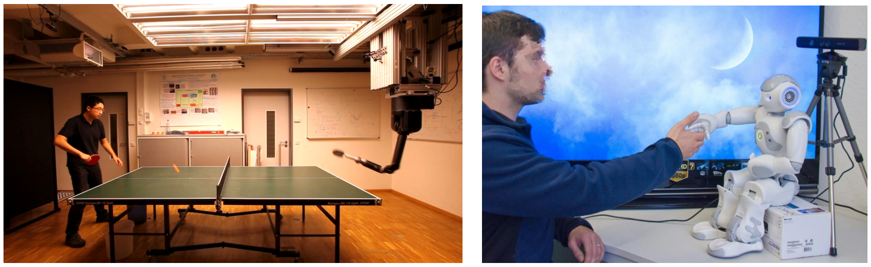
(b) IDDM

Figure 1: Graphical models of the Gaussian process dynamical 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 (Wang et al., 2012b).

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 article, 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 Fig. 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 Dynamical Model (GPDM) proposed in (Wang et al., 2008) uses GPs for modeling the



(a) Robot table tennis.

(b) Interactive humanoid robot.

Figure 2: Two examples of HRI scenarios where intention inference plays an important role: (a) target prediction in robot table tennis games, and (b) action recognition for human-robot interaction.

generative process of human motion with a nonlinear dynamical system, as shown in Fig. 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 states are driven by the intention of the human action/behavior, as shown in Fig.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 article is organized as follows. First, in this section, we illustrate the considered scenarios (Section 1.1) and discuss the related work (Section 1.2). Subsequently, we present the Intention- Driven Dynamics Model (IDDM) and address the problem of its training in Section 2. In Section 3, we study approximate algorithms for intention inference and extend them to online inference in Section 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 5 and 6. Finally, we summarize our contributions and discuss properties of the IDDM in Section 7.

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 Fig. 2a. 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 incomining balls due to a lack of time caused by acceleration and torque limits for the biomimetic robot table tennis player described in (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 for the whole kinematically reachable hitting plane, 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 utilizing 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 Fig. 2b, 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 our previous method in (Wang et al., 2012b).

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.

1.2.1 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., 2012), 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.

1.2.2 Gaussian Process Dynamical 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 Dynamical 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).

2 Intention-Driven Dynamics Model

We propose the Intention-Driven Dynamics Model (IDDM), which is an extension of the GPDM (Wang et al., 2008). The GPDM is a nonparametric approach to learning the transition function in the latent state space and the measurement mapping from states to observations simultaneously. As shown in Fig. 1a, the transition function in the GPDM is only determined by the latent state. However, in the applications considered in this paper, the underlying intention, as an important drive of human movements, can hardly be discovered directly from the observations. Considering that the dynamics can be substantially different when the actions are based on different intentions, we propose the Intention-Driven Dynamics Model. As shown in Fig. 1b, the IDDM explicitly incorporates the intention into the transition function in the latent state space. This dynamics model was inspired by the hypothesis that the human action is directed by the goal (Baker et al., 2009; Friesen and Rao, 2011). For example, in table tennis, the player swings the racket in order to return the ball to an intended target. The target is, hence, a driving factor in the dynamics of the racket.

We present the proposed model and address the problem of its training in this section. Later, in Section 3, we study approximate algorithms for intention inference, and extend it for online inference in Section 4.

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.

This article extensively uses properties of the Gaussian processes, e.g., predictive distribution and marginal likelihood. We refer to (Rasmussen and Williams, 2006) for a comprehensive introduction to GPs.

2.1.1 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

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

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

$$\mathbf{m}_z(\mathbf{x}_t) = \mathbf{Y}_z\mathbf{K}_z^{-1}\mathbf{k}_z(\mathbf{x}_t), \tag{2}$$

$$\mathbf{\Sigma}_z(\mathbf{x}_t) = \sigma_z^2(\mathbf{x}_t)\mathbf{I}, \tag{3}$$

$$\sigma_z^2(\mathbf{x}_t) = k_z(\mathbf{x}_t, \mathbf{x}_t) - \mathbf{k}_z(\mathbf{x}_t)^T\mathbf{K}_z^{-1}\mathbf{k}_z(\mathbf{x}_t)\,, \tag{4}$$

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 $\mathbf{X}_z$.

2.1.2 Transition model

We consider first-order Markov transition model, see Fig. 1b, with a latent transition function $\mathbf{f}$, such that

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, g) + \mathbf{n}_{x,t}, \quad \mathbf{n}_{x,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_x)\,. \tag{5}$$

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

$$\mathbf{m}_x([\mathbf{x}_t, g]) = \mathbf{Y}_x\mathbf{K}_x^{-1}\mathbf{k}_x([\mathbf{x}_t, g]), \tag{6}$$

$$\mathbf{\Sigma}_x([\mathbf{x}_t, g]) = \sigma_x^2([\mathbf{x}_t, g])\mathbf{I}, \tag{7}$$

$$\sigma_x^2([\mathbf{x}_t, g]) = k_x([\mathbf{x}_t, g], [\mathbf{x}_t, g]) - \mathbf{k}_x([\mathbf{x}_t, g])^T\mathbf{K}_x^{-1}\mathbf{k}_x([\mathbf{x}_t, g])\,, \tag{8}$$

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.

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.,

$$\begin{aligned} k_x([\mathbf{x}_i, g_i], [\mathbf{x}_j, g_j]; \boldsymbol{\alpha}) &= k_x(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\alpha}) k_x(g_i, g_j; \boldsymbol{\alpha}) + k_{\text{noise}} \\ &= \alpha_1 \exp\left(-\tfrac{\alpha_2}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2 - \tfrac{\alpha_3}{2}(g_i - g_j)^2\right) + \alpha_4 \delta_{ij} \,, \end{aligned} \tag{9}$$

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

$$k_z(\mathbf{x}, \mathbf{x}'; \boldsymbol{\beta}) = \exp\left(-\tfrac{\beta_1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) + \beta_2 \delta_{\mathbf{x},\mathbf{x}'}, \tag{10}$$

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. (1). In the context of target prediction in table tennis games, we use the linear kernel

$$k_z(\mathbf{x}, \mathbf{x}'; \boldsymbol{\beta}) = \mathbf{x}^T \mathbf{x}' + \beta_1 \delta_{\mathbf{x},\mathbf{x}'}, \tag{11}$$

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

2.3 Learning the IDDM

The proposed IDDM can be learned from a training data set $\mathcal{D} = \{\mathbf{Z}, g\}$ of J movements and corresponding intentions. Each movement $\mathbf{Z}^j$ consists of a time series of observations given by $\mathbf{Z}^j = [\mathbf{z}_1^j, \ldots, \mathbf{z}_T^j]^T$. We construct the overall observation matrix $\mathbf{Z}$ by vertically concatenating the observation matrices $\mathbf{Z}^1, \ldots, \mathbf{Z}^J$, and the overall intention matrix g from $g^1, \ldots, g^J$. In the robot table tennis example, one movement corresponds to a stroke of the opponent, represented by a time series of observed racket and ball configurations. We assume the intention g can be obtained for training, for example by post-processing the data. In the robot table tennis example, the observed intention corresponds to the target where the opponent returns the ball to (see Fig. 5 for an illustration). In the table-tennis training data, we can obtained the target's coordinates by post-processing. In the action recognition, the label of action is provided directly in the training data.

Similar to the GPDM (Wang et al., 2008), we find maximum a posteriori (MAP) estimates of the latent states $\mathbf{X}$. Alternative learning methods and an empirical comparison can be found in (Turner et al., 2010; Damianou et al., 2011). Given the model hyperparameters, the posterior distribution of latent states $\mathbf{X}$ can be decomposed into the probability of the observations given the states and the probability of the states given the intention, i.e.,

$$p(\mathbf{X}|\mathbf{Z}, g, \boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{W}) \propto p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\beta}, \mathbf{W})p(\mathbf{X}|g, \boldsymbol{\alpha}), \tag{12}$$

both obtained by the GP marginal likelihood (Rasmussen and Williams, 2006). The GP marginal probability of the observations $\mathbf{Z}$ given the latent states $\mathbf{X}$ is given by a Gaussian distribution

$$p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\beta}, \mathbf{W}) = \frac{|\mathbf{W}|^M}{\sqrt{(2\pi)^{MD_z}|\mathbf{K}_z|^{D_z}}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_z^{-1}\mathbf{Z}\mathbf{W}\mathbf{W}^\top\mathbf{Z}^T\right)\right), \tag{13}$$

where $M \triangleq JT$ is the length of observations $\mathbf{Z}$, and $\mathbf{K}_z$ is the kernel matrix computed by the kernel function $k_z(\cdot, \cdot)$. Given the intention g, the sequence of latent states $\mathbf{X}$ has a Gaussian probability

$$\begin{aligned} p(\mathbf{X}|g, \boldsymbol{\alpha}) &= p(\mathbf{X}_1)p(\mathbf{X}_{2:T}|\mathbf{X}_{1:T-1}, g, \boldsymbol{\alpha}) \\ &= \frac{p(\mathbf{X}_1)}{\sqrt{(2\pi)^{mD_x}|\mathbf{K}_x|^{D_x}}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_x^{-1}\mathbf{X}_{2:T}\mathbf{X}_{2:T}^T\right)\right), \end{aligned} \tag{14}$$

where $\mathbf{X}_t, t \in \{1, \ldots, T\}$ is constructed by vertically concatenating state matrices $\mathbf{x}_t^1, \ldots, \mathbf{x}_t^J$, $m \triangleq J(T-1)$ is the length of $\mathbf{X}_{2:T}$, and $\mathbf{K}_x$ is the kernel matrix of $\mathbf{X}_{2:T}$, computed by the kernel function $k_x(\cdot, \cdot)$. We use a Gaussian prior distribution on the initial states $\mathbf{X}_1$.

Based on Eqs. (13)–(14), the MAP estimates of the states are obtained by maximizing the posterior in Eq. (12). In practice, we minimize the negative log-posterior

$$\begin{aligned} \mathcal{L}(\mathbf{X}) =& \frac{D_z}{2}\log|\mathbf{K}_z| + \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_z^{-1}\mathbf{Z}\mathbf{W}\mathbf{W}^\top\mathbf{Z}^T\right) - M\log|\mathbf{W}| \\ &+ \frac{D_x}{2}\log|\mathbf{K}_x| + \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_x^{-1}\mathbf{X}_{2:T}\mathbf{X}_{2:T}^T\right) + \frac{1}{2}\mathrm{tr}\left(\mathbf{X}_1\mathbf{X}_1^T\right) + \mathrm{const} \end{aligned} \tag{15}$$

with respect to the states $\mathbf{X}$, using the Scaled Conjugate Gradient (SCG) method (Møller, 1993).

2.4 Learning Hyperparameters

A reliable approach to learning the hyperparameters $\boldsymbol{\Theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{W}\}$ is to maximize the marginal likelihood

$$p(\mathbf{Z}|g, \boldsymbol{\Theta}) = \int p(\mathbf{Z}, \mathbf{X}|g, \boldsymbol{\Theta})d\mathbf{X}, \tag{16}$$

which can be achieved approximately by using the Expectation-Maximization (EM) algorithm (Bishop, 2006). The EM algorithm computes the posterior distribution of states $q(\mathbf{X}) = p(\mathbf{X}|\mathbf{Z}, g, \boldsymbol{\Theta})$, given in Eq. (12), in the Expectation (E) step and updates the hyperparameters by maximizing the expected data likelihood $\mathbb{E}_q[p(\mathbf{Z}, \mathbf{X}|g, \boldsymbol{\Theta})]$ in the Maximization (M) step. However, the posterior distribution $q(\mathbf{X})$ is difficult to compute in the IDDM. Following (Wang et al., 2008), we draw samples of the states

Algorithm 1: Learning the model hyperparameters $\boldsymbol{\alpha}, \boldsymbol{\beta}$, and $\mathbf{W}$ by maximizing the marginal likelihood, using the Monte Carlo EM algorithm.

```
Input  : Data: $\mathcal{D} = \{\mathbf{Z}, g\}$
Input  : Number of EM iterations: $L$
Output: Model hyperparameters: $\boldsymbol{\Theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{W}\}$
for $l \leftarrow 1$ to $L$ do
    for $i \leftarrow 1$ to $I$ do
        Initialize $\mathbf{X}$ by its MAP estimate ;
        Draw sample $\mathbf{X}^{(i)}$ from $p(\mathbf{X}|\mathbf{Z}, g, \boldsymbol{\Theta})$ using HMC;
    Maximize $\frac{1}{I}\sum_{i=1}^{I} \log p(\mathbf{Z}, \mathbf{X}^{(i)}|g, \boldsymbol{\Theta})$ w.r.t. $\boldsymbol{\Theta}$ using SCG;
```

$\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(I)}$ from the posterior distribution using hybrid Monte Carlo (Andrieu et al., 2003), and, hence, the data likelihood is estimated via Monte Carlo integration according to

$$\mathbb{E}_q[p(\mathbf{Z}, \mathbf{X}|g, \boldsymbol{\Theta})] \approx \frac{1}{I}\sum_{i=1}^{I} p(\mathbf{Z}, \mathbf{X}^{(i)}|g, \boldsymbol{\Theta}). \tag{17}$$

In the M step, we use SCG to update the hyperparameters. In practice, we choose the number of samples $I = 50$ and the number of EM iterations $L = 10$. Although this procedure, as described in Algorithm 1, is time-demanding, in practice, we can learn the hyperparameters off-line.

In practice, the maximum likelihood estimate of the hyperparameters may lead to over-fitting. For the IDDM, we found that the noise variance α_4 in Eq. (9) of the transition model is occasionally underestimated, e.g., $\alpha_4 < e^{-6}$, as Algorithm 1 estimates it based on only a few samples. The underestimated noise variance may prevent the learned model from generalizing to test data that have significant deviation from the training data. This phenomenon of over-confidence has been discussed in (Lawrence, 2005; Wang et al., 2008). To alleviate this problem, we add a small constant e^{-3} to the learned noise variance α_4.

The model also depends on the hyperparameter D_x, i.e., the dimensionality of the latent state space. Choosing an appropriate D_x is important. If the dimensionality is too small, the latent states cannot recover the observations, which leads to significant prediction errors. On the other hand, a high-dimensional state space results in redundancy and can cause a drop in performance and computational efficiency. Nevertheless, *model selection*, based on cross-validation for example, is conducted before learning and applying the model.

To summarize, the model $\mathcal{M} = \{\mathbf{X}, \boldsymbol{\Theta}\}$ can be learned from a data set $\mathcal{D}$. Subsequently, we use the model to infer the unobserved intention of a new ongoing movement, as described in the following section.

3 Approximate Intention Inference

After learning the model $\mathcal{M}$ from the training data set $\mathcal{D}$, 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 $\mathcal{M}$ and the data set $\mathcal{D}$. The measurement model defined in Eq. (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

$$\begin{aligned} p(g|\mathbf{z}_{1:T}) &= \frac{p(\mathbf{z}_{1:T}|g)p(g)}{p(\mathbf{z}_{1:T})} && (18) \\ &\propto p(g) \int p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}|g) d\mathbf{x}_{1:T}, && (19) \end{aligned}$$

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. (19) is not tractable due to the use of nonlinear GP transition model. Hence, we resort to approximate inference. In (Wang et al., 2012b), we introduced an EM algorithm for finding the maximum likelihood estimate of 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., 2012). 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 article, we extend our previous inference method introduced in (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. (19) is integrating out the latent states $\mathbf{x}_{1:T}$. A common approximation to the log marginal posterior is to compute a lower bound $\mathcal{B}(g) \leq \log p(g|\mathbf{z}_{1:T})$ based on Jensen's inequality (Bishop, 2006). The bound is given by

$$\begin{aligned} \mathcal{B}(g) &\triangleq \mathbb{E}_q \left[\log p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}, g)\right] + \mathcal{H}(q) && (20) \\ &= \log p(g|\mathbf{z}_{1:T}) - \mathrm{KL}\left(q||p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)\right) \\ &\leq \log p(g|\mathbf{z}_{1:T}), \end{aligned}$$

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 $\mathcal{B}(g)$ 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 $\mathcal{B}(g)$. When using the EM algorithm for the maximum likelihood estimate of the intention g, as in (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 $\mathcal{B}$ in Eq. (20) is given by

$$q(\mathbf{x}_{1:T}) = \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q) \approx p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g), \tag{21}$$

with the mean and block-tri-diagonal covariance matrix

$$\boldsymbol{\mu}_q = \begin{bmatrix} \boldsymbol{\mu}^x_{1|T} \\ \vdots \\ \boldsymbol{\mu}^x_{T|T} \end{bmatrix}, \quad \boldsymbol{\Sigma}_q = \begin{bmatrix} \boldsymbol{\Sigma}^x_{1|T} & \boldsymbol{\Sigma}^x_{1,2|T} & & \mathbf{0} \\ \boldsymbol{\Sigma}^x_{2,1|T} & \ddots & \ddots & \\ & \ddots & \ddots & \boldsymbol{\Sigma}^x_{T-1,T|T} \\ \mathbf{0} & & \boldsymbol{\Sigma}^x_{T,T-1|T} & \boldsymbol{\Sigma}^x_{T|T} \end{bmatrix}, \tag{22}$$

where we only need to consider the cross-covariance between consecutive states[1].

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 $\mathcal{B}(g)$ in Eq. (20).

In the following, we first detail step (a), i.e., the computation of q for our IDDM, in Section 3.1. Subsequently, we discuss step (b), i.e., efficient belief estimation, in Section 3.2.

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 $t = 2, \ldots, T$ are computed. We explicitly determine the marginals $p(\mathbf{x}_t|\mathbf{z}_{1:T}, g)$ for $t = 1, \ldots, T$, and the cross-covariance terms $\mathrm{cov}[\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:T}, g]$, $t = 2, \ldots, T$. These steps yield a Gaussian approximation with a block-tri-diagonal covariance matrix, see Eq. (22). These computations are based on forward-backward smoothing (GP-RTSS) as proposed in (Deisenroth et al., 2012).

As a first step, we compute the posterior distributions $p(\mathbf{x}_t|\mathbf{z}_{1:T}, g)$ with $t = 1, \ldots, T$. 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

$$\boldsymbol{\mu}^x_{t|t} = \boldsymbol{\mu}^x_{t|t-1} + \boldsymbol{\Sigma}^{xz}_{t|t-1}(\boldsymbol{\Sigma}^z_{t|t-1})^{-1}(\mathbf{z}_t - \boldsymbol{\mu}^z_{t|t-1})\,, \tag{23}$$

$$\boldsymbol{\Sigma}^x_{t|t} = \boldsymbol{\Sigma}^x_{t|t-1} - \boldsymbol{\Sigma}^{xz}_{t|t-1}(\boldsymbol{\Sigma}^z_{t|t-1})^{-1}\boldsymbol{\Sigma}^{zx}_{t|t-1}\,, \tag{24}$$

$$\boldsymbol{\mu}^x_{t-1|T} = \boldsymbol{\mu}^x_{t-1|t-1} + \mathbf{J}_{t-1}(\boldsymbol{\mu}^x_{t|T} - \boldsymbol{\mu}^x_{t|t-1})\,, \tag{25}$$

$$\boldsymbol{\Sigma}^x_{t|T} = \boldsymbol{\Sigma}^x_{t-1|t-1} + \mathbf{J}_{t-1}(\boldsymbol{\Sigma}^x_{t|T} - \boldsymbol{\Sigma}^x_{t|t-1})\mathbf{J}^\top_{t-1}\,, \tag{26}$$

[1] 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).

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

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

Without loss of generality, the marginal distribution $\mathcal{N}(\mathbf{x}_{t-1} \mid \boldsymbol{\mu}^x_{t-1|t-1}, \boldsymbol{\Sigma}^x_{t-1|t-1})$, 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 $\tilde{\boldsymbol{\mu}}_{t-1|t-1} = [(\boldsymbol{\mu}^x_{t-1|t-1})^\top, \boldsymbol{\mu}_g^\top]^\top$ 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 $\tilde{\mathbf{x}} = [\mathbf{x}^\top, g]^\top$.

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

$$\begin{aligned}(\mu^x_{t|t-1})_a &= \mathbb{E}_{\mathbf{x}_{t-1}}\left[\mathbb{E}_{f_a}[f_a(\tilde{\mathbf{x}}_{t-1})|\tilde{\mathbf{x}}_{t-1}]|\mathbf{z}_{1:t-1}\right] \qquad (29)\\ &= \int m^a_x(\tilde{\mathbf{x}}_{t-1})p(\tilde{\mathbf{x}}_{t-1}|\mathbf{z}_{1:t-1})d\tilde{\mathbf{x}}_{t-1} \,,\end{aligned}$$

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

$$(\mu^x_{t|t-1})_a = \mathbf{q}^\top \boldsymbol{\gamma}_a \,, \tag{30}$$

where we define

$$\mathbf{q}^\top = \int k_x([\mathbf{x}_{t-1}, g], \mathbf{X}_x)p(\tilde{\mathbf{x}}_{t-1}|\mathbf{z}_{1:t-1})d\tilde{\mathbf{x}}_{t-1} \,. \tag{31}$$

Here, $\mathbf{X}_x$ denotes the set of the M GP training inputs $\tilde{\mathbf{x}}_i = [\mathbf{x}_i^\top, g_i^\top]^\top$ of the transition GP. Since k_x is a Gaussian kernel, we can solve the integral in Eq. (31) analytically and obtain the vector $\mathbf{q}$ with entries q_i with $i = 1, \ldots, M$ as

$$q_i = \alpha_1|\boldsymbol{\Omega}|^{-\frac{1}{2}} \exp\big(-\tfrac{1}{2}\boldsymbol{\zeta}_i^\top(\boldsymbol{\Lambda}\boldsymbol{\Omega})^{-1}\boldsymbol{\zeta}_i\big) \,, \tag{32}$$

$$\boldsymbol{\zeta}_i = \tilde{\mathbf{x}}_i - \tilde{\boldsymbol{\mu}}_{t-1|t-1} \,, \quad \boldsymbol{\Omega} = \boldsymbol{\Sigma}_{t-1|t-1}\boldsymbol{\Lambda}^{-1} + \mathbf{I} \,, \tag{33}$$

where $\mathbf{\Lambda}$ 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_{ab}^x of the *marginal predictive covariance matrix* $\mathbf{\Sigma}_{t|t-1}^x$ in Eq. (28) are given by

$$\sigma_{ab}^x = \begin{cases} \gamma_a^\top(\mathbf{Q}^x - \mathbf{q}\mathbf{q}^\top)\boldsymbol{\gamma}_b & \text{if } a \neq b\,, \\ \gamma_a^\top(\mathbf{Q}^x - \mathbf{q}\mathbf{q}^\top)\boldsymbol{\gamma}_b + \alpha_1 - \mathrm{tr}\big((\mathbf{K}_x + \alpha_4\mathbf{I})^{-1}\mathbf{Q}^x\big) + \alpha_4 & \text{if } a = b\,. \end{cases} \tag{34}$$

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

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

with

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

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 $\mathbf{\Sigma}_{t-1,t|t-1}^x = \mathrm{cov}[\mathbf{x}_{t-1}, \mathbf{x}_t|\mathbf{z}_{1:t-1}, g]$ is given as the upper part of the cross-covariance

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

when we set $\boldsymbol{\mu}_g = g$ and $\mathbf{\Sigma}_g = \mathbf{0}$. Note that $\mathbf{q}$ and $\mathbf{\Omega}$ are defined in Eq. (32) and (33), 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. (23)–(26). To determine a Gaussian approximation

$$\mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{t|t-1}^x \\ \boldsymbol{\mu}_{t|t-1}^z\end{bmatrix}, \begin{bmatrix}\mathbf{\Sigma}_{t|t-1}^x & \mathbf{\Sigma}_{t|t-1}^{xz} \\ \mathbf{\Sigma}_{t|t-1}^{zx} & \mathbf{\Sigma}_{t|t-1}^z\end{bmatrix}\right) \tag{35}$$

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. (11), we compute the *marginal mean* $\boldsymbol{\mu}_{t|t-1}^z$ in Eq. (35) for observation dimension $a = 1, \ldots, D_z$ according to

$$\begin{aligned} \mathbb{E}_{h,\mathbf{x}_{t-1}}[h_a(\mathbf{x}_t)|\mathbf{z}_{1:t-1}, g] &= \int m(\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, g)\,\mathrm{d}\mathbf{x}_t \\ &= \int \mathbf{x}_t^\top\mathbf{X}_z^\top p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, g)\,\mathrm{d}\mathbf{x}_t\boldsymbol{\xi}_a = \mathbf{q}^\top\boldsymbol{\xi}_a\,, \qquad (36) \\ \mathbf{q} &= \mathbf{X}_z\boldsymbol{\mu}_{t|t-1}^x\,. \end{aligned}$$

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, $a = 1, \dots, D_z$. The elements σ_{ab}^z of the *marginal covariance matrix* $\boldsymbol{\Sigma}_{t|t-1}^z$ in Eq. (35) are given as

$$\sigma_{ab}^z = \begin{cases} \boldsymbol{\xi}_a^\top(\mathbf{Q}^z - \mathbf{q}\mathbf{q}^\top)\boldsymbol{\xi}_a & \text{if } a \neq b\,, \\ \boldsymbol{\Sigma}_{t|t-1}^x + \boldsymbol{\mu}_{t|t-1}^x(\boldsymbol{\mu}_{t|t-1}^x)^\top - \operatorname{tr}\big(\mathbf{K}_z^{-1}\mathbf{Q}^z\big) + \boldsymbol{\xi}_a^\top(\mathbf{Q}^z - \mathbf{q}\mathbf{q}^\top)\boldsymbol{\xi}_a & \text{if } a = b\,, \end{cases} \tag{37}$$

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

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

The *cross-covariance* $\boldsymbol{\Sigma}_{t|t-1}^{xz} = \operatorname{cov}[\mathbf{x}_t, \mathbf{z}_t|\mathbf{z}_{1:t-1}, g]$ in Eq. (35) is given as

$$\boldsymbol{\Sigma}_{t|t-1}^{xz} = \boldsymbol{\Sigma}_{t|t-1}^x\mathbf{X}_z^\top\boldsymbol{\xi}_a \tag{38}$$

for all observed dimensions $a = 1, \dots, D_z$. The mean $\boldsymbol{\mu}_{t|t-1}^z$ in Eq. (36), the covariance matrix $\boldsymbol{\Sigma}_{t|t-1}^z$ in Eq. (37), and the cross-covariance in Eq. (38) fully determine the Gaussian distribution in Eq. (35). Hence, following (Deisenroth and Ohlsson, 2011), we can now compute the latent state posteriors (filter and smoothing distributions) according to Eq. (23)–(26).

These smoothing updates in Eq. (23)–(26) yield the marginals of our Gaussian approximation to $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)$, see Eq. (22). The missing cross-covariances $\boldsymbol{\Sigma}_{t-1,t|T}^x$ of $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, g)$ that finally fully determine the block-tri-diagonal covariance matrix in Eq. (22) are given by

$$\boldsymbol{\Sigma}_{t-1,t|T}^x = \mathbf{J}_{t-1}\boldsymbol{\Sigma}_{t|T}^x\,, \tag{39}$$

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. (20), i.e., the computation of the approximate distribution q in Eq. (21). It remains to compute the bound $\mathcal{B}$ itself, which is described in the following.

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

$$q(\mathbf{x}_t, \mathbf{x}_{t+1}) = \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{t|T}^x \\ \boldsymbol{\mu}_{t+1|T}^x\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{t|T}^x & \boldsymbol{\Sigma}_{t,t+1|T}^x \\ \boldsymbol{\Sigma}_{t+1,t|T}^x & \boldsymbol{\Sigma}_{t+1|T}^x\end{bmatrix}\right) \tag{40}$$

for $t = 1, \dots, T-1$. The belief $p(g|\mathbf{z}_{1:T}) \approx \exp(\mathcal{B}(g))$ is estimated using Eq. (20), where the computation can be decomposed according to

$$\mathcal{B}(g) = \sum_{t=1}^{T-1} \underbrace{\mathbb{E}_q\left[\log p(\mathbf{x}_{t+1}|\mathbf{x}_t, g)\right]}_{\mathcal{Q}_t(g)} + p(g) + \mathcal{H}(q) + \text{const.} \tag{41}$$

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 $\mathcal{H}(q)$ of the Gaussian distribution q can be computed analytically, and is given by

$$\mathcal{H}(q) = \frac{1}{2}\left(TD_x + TD_x \log(2\pi) + \log|\mathbf{\Sigma}_q|\right). \tag{42}$$

We define

$$\begin{aligned}\mathcal{Q}_t(g) &\triangleq \mathbb{E}_q\left[\log p(\mathbf{x}_{t+1}|\mathbf{x}_t, g)\right] && (43)\\ &= \iint q(\mathbf{x}_t, \mathbf{x}_{t+1}) \log p(\mathbf{x}_{t+1}|\mathbf{x}_t, g) d\mathbf{x}_{t+1} d\mathbf{x}_t\\ &= \iint q(\mathbf{x}_t, \mathbf{x}_{t+1}) \log \underbrace{\left(p(\mathbf{x}_{t+1}|\mathbf{x}_t, g) q(\mathbf{x}_t)\right)}_{\approx \tilde{q}(\mathbf{x}_t, \mathbf{x}_{t+1})} d\mathbf{x}_{t+1} d\mathbf{x}_t - \int q(\mathbf{x}_t) \log q(\mathbf{x}_t) d\mathbf{x}_t,\end{aligned}$$

where $p(\mathbf{x}_{t+1}|\mathbf{x}_t, g)q(\mathbf{x}_t)$ can be approximated by a Gaussian distribution $\tilde{q}(\mathbf{x}_t, \mathbf{x}_{t+1}) = \mathcal{N}(\boldsymbol{\mu}_{\tilde{\mathbf{q}}}, \mathbf{\Sigma}_{\tilde{\mathbf{q}}})$ based on moment matching (Quiñonero-Candela et al., 2003). Here, we only compute the diagonal elements in the covariance matrix of $\mathbf{\Sigma}_{\tilde{\mathbf{q}}}$. As a result, Eq. (43) is approximated as

$$\mathcal{Q}_t(g) \approx \mathrm{KL}\big(q(\mathbf{x}_t, \mathbf{x}_{t+1}) || \tilde{q}(\mathbf{x}_t, \mathbf{x}_{t+1})\big) + \mathcal{H}\big(q(\mathbf{x}_t, \mathbf{x}_{t+1})\big) + \mathcal{H}\big(q(\mathbf{x}_t)\big), \tag{44}$$

where $\mathcal{H}(q)$ 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

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

As a result, we can compute the unnormalized belief $\mathcal{B}(g)$ for a given intention g approximately according to Eq. (41).

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 article, we advocate the discretization of intention. For example, in the table tennis task, the intention (opponent's target position) is a bounded scalar variable $g \in [g_{\text{min}}, g_{\text{max}}]$, where the bounds are given by physical constraints such as the table width and the length of robot arm. We uniformly choose $\{v_1, \ldots, v_K\}$ from $[g_{\text{min}}, g_{\text{max}}]$ and represent intention by the index, i.e., $g \in \{1, \ldots, K\}$.

Algorithm 2: 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 $g \in \{1, \dots, K\}$
foreach $g \in \{1, \dots, K\}$ do
    Compute smoothing distribution $q(\mathbf{x}_{1:T}) \approx p(\mathbf{x}_{1:T} | \mathbf{z}_{1:T}, g)$ ;
    Compute the value of $\boldsymbol{\mathcal{B}}(g) = \mathbb{E}_q \left[ \log p(\mathbf{x}_{1:T} | g) \right] + \log p(g)$ using the
    approximation in Eq. (44) ;
Estimate the posterior $p(g | \mathbf{x}_{1:T}) \approx \exp \boldsymbol{\mathcal{B}}(g) / (\sum_{g'=1}^{K} \exp \boldsymbol{\mathcal{B}}(g'))$.
```

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 2. The smoothing distribution q defined in Eq. (40) 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 2 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, 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 robotics 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 come, 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 4.

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 2 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.

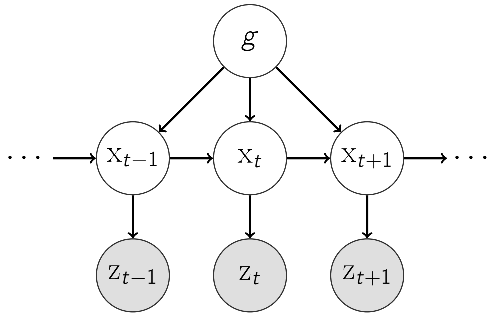


Figure 3: The graphical model of the IDDM in an online manner, which can handle a stream of observations.

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,

$$q(\mathbf{x}_{1:t}) \triangleq \sum_g p(g) q_g(\mathbf{x}_{1:t}). \tag{45}$$

The online inference algorithm then estimates the belief $\boldsymbol{\mathcal{B}}_t(g)$ 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.

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

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 Fig. 3). 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. (41), given by

$$\boldsymbol{\mathcal{B}}_{t-1}(g) \approx \mathbb{E}_q \left[\log p(g, \mathbf{x}_{1:t-1})\right] + \text{const}. \tag{46}$$

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

$$p(g, \mathbf{x}_{1:t}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}, g) p(g, \mathbf{x}_{1:t-1}). \tag{47}$$

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

$$\begin{aligned}
\boldsymbol{\mathcal{B}}_t(g) &= \mathbb{E}_q \left[\log p(g, \mathbf{x}_{1:t})\right] + \text{const} && (48)\\
&= \mathbb{E}_q \left[\log p(\mathbf{x}_t | \mathbf{x}_{t-1}, g)\right] + \mathbb{E}_q \left[\log p(g, \mathbf{x}_{1:t-1})\right] + \text{const} && (49)\\
&= \mathbb{E}_q \left[\log p(\mathbf{x}_t | \mathbf{x}_{t-1}, g)\right] + \boldsymbol{\mathcal{B}}_{t-1}(g) + \text{const}, && (50)
\end{aligned}$$

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. (43)-(44). The belief $\boldsymbol{\mathcal{B}}_t$ is then normalized, i.e., $\sum_g \exp(\boldsymbol{\mathcal{B}}_t(g)) = 1$.

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 $\boldsymbol{\mathcal{B}}_{t-1}$. The recursive formula of the belief is subsequently given by

$$\boldsymbol{\mathcal{B}}_t(g) = \mathbb{E}_q\left[\log p(\mathbf{x}_t|\mathbf{x}_{t-1}, g)\right] + (1-\epsilon)\boldsymbol{\mathcal{B}}_{t-1}(g), \tag{51}$$

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

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 $\boldsymbol{\mathcal{B}}_{t-1}$. In analogy to the EM algorithm, we iteratively update the belief on intention $\boldsymbol{\mathcal{B}}$ 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 not unnecessary since we do not update the previous belief $\boldsymbol{\mathcal{B}}_{1:t-1}$ 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 $\boldsymbol{\mathcal{B}}_{t-1}$.

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

$$q(\mathbf{x}_{t-1:t}) \propto \sum_g q_g(\mathbf{x}_{t-1:t}) p_{t-1}(g), \tag{52}$$

where the belief $p_{t-1}(g) \propto \exp(\boldsymbol{\mathcal{B}}_{t-1}(g))$. 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 $\boldsymbol{\mathcal{B}}_{t-1}$. As a result, the marginal smoothing distribution is given by

$$q(\mathbf{x}_{t-1:t}) \approx \int q_g(\mathbf{x}_{t-1:t}) \mathcal{N}(g|\mu_g, \sigma_g^2) dg, \tag{53}$$

which is computed using moment matching.

4.3 Discussion of the Online Inference Method

The online inference algorithm described in Algorithm 3 iteratively updates the belief of intention and latent states.

Algorithm 3: 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 $\boldsymbol{\mathcal{B}}_1(g) = \log p(g)$ 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 $\boldsymbol{\mathcal{B}}_{t-1}$ ;
    Update marginal smoothing distribution $q(\mathbf{x}_{t-1})$ according to current belief $\boldsymbol{\mathcal{B}}_{t-1}$ ;
    foreach $g_t = \{1, \ldots, K\}$ do
        Compute $\boldsymbol{\mathcal{B}}^0(g) = \mathcal{Q}_{t-1}(g)$ using the approximation in Eq. (44) ;
    Update the belief $\boldsymbol{\mathcal{B}}_t = \boldsymbol{\mathcal{B}}^0 + (1 - \epsilon)\boldsymbol{\mathcal{B}}_{t-1}$ ;
    Normalize the belief $\boldsymbol{\mathcal{B}}_t \leftarrow \boldsymbol{\mathcal{B}}_t - \log\left(\sum_g \exp(\boldsymbol{\mathcal{B}}_t(g))\right)$ ;
```

The computational complexity of the smoothing step in Algorithm 3 is $\mathcal{O}(D_z^3 + D_x D_z^2 + N^2 D_x^3)$ when using the linear kernel function for the measurement mapping, and $\mathcal{O}(D_z^3 + N^2(D_x D_x^2 + D_x^3))$ 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 $\mathcal{O}(KN^2 D_x^2)$. Comparing 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 1.

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 levels of a young child, despite having robots with better perception, processing power, and accuracy

Table 1: Important approximations employed in the batch and online inference.

	batch	online
belief $p(g\|\mathbf{z}_{1:T})$	Jensen's lower bound $\mathcal{B}(g)$; cf. Eq. (20)	
approx. belief $\mathcal{B}(g)$	moment matching; cf. Eq. (44)	
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. (53)
stream of observations	sliding window	recursive update; cf. Eq. (51)

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). In this article, 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[2]. 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 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[3]. Therefore, prediction accuracy directly influences the performance of the robot player (Wang et al., 2011b).

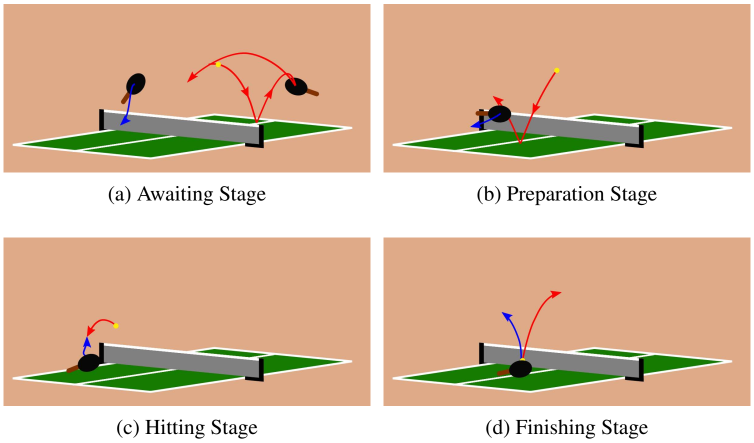
5.1 Experimental Setting

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 Fig. 4, 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.,

[2] 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.

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


(a) Awaiting Stage

(b) Preparation Stage

(c) Hitting Stage

(d) Finishing Stage

Figure 4: 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).

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). 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 $\mathcal{B}$.

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 Fig. 5. 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 Fig. 5) 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 to ±1.2 m from the robot's base (table is 1.52 m

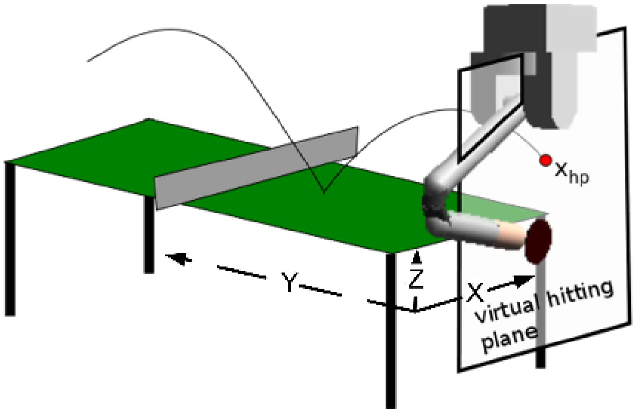


Figure 5: The robot's hitting point is the intersection of the coming ball's trajectory and the virtual hitting plane 80 cm behind the table. Figure is adapted from (Mülling et al., 2011).

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 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.

5.2 Results

As demonstrated in Fig. 6, the proposed IDDM model outperformed the GPR baseline. At 80 ms before the opponent hit the ball, the batch algorithm resulted in the 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,

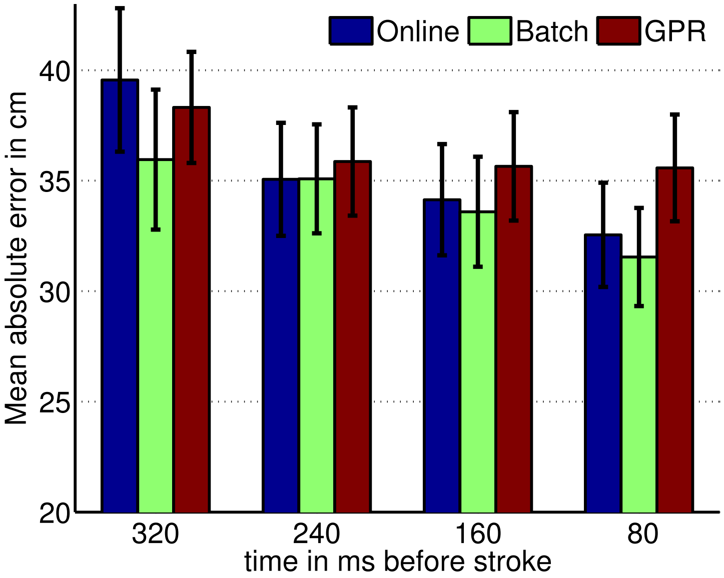


Figure 6: Mean absolute error of the ball's target with standard error of the mean. The algorithms use the observations obtained *before* the opponent has hit the ball.

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 $78.8\,\mathrm{cm}$. Hence, both the GPR and IDDM substantially outperformed naive goal prediction.

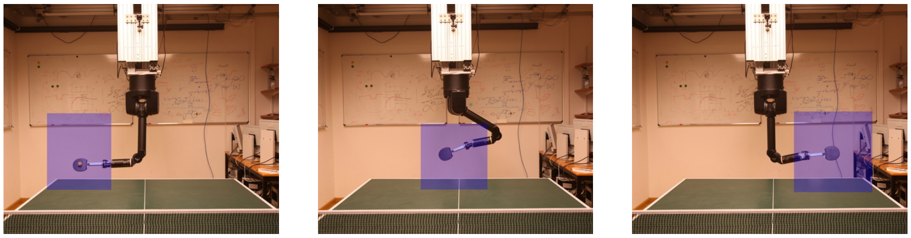
The online algorithm, with a shrinking factor $\epsilon = 0.2$ given in Eq. (51), took on average $70\,\mathrm{ms}$ to process every observation, which can potentially fulfill the real-time requirements of $80\,\mathrm{ms}$. The batch algorithm used a sliding window of size 4, and took on average $300\,\mathrm{ms}$ to process every observation. The online algorithm was significantly faster than the batch algorithm, with a small loss in accuracy[4]. 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. (10), or a linear kernel, see Eq. (11). 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 2.

[4] 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 and 4.3

Table 2: The mean absolute errors (in cm) with standard error of the mean of the goal inference made $80\,\mathrm{ms}$ *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	$\mathbf{31.5 \pm 2.2}$	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



(a) Forehand pose.

(b) Middle pose.

(c) Backhand pose.

Figure 7: 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.

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 Fig. 7. As exhibited in Fig. 8, our method allows the robot to choose the responding hitting movement before 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).

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., 2011) has provided us with the technology to record the full skeletal configuration of the human during HRI. Nevertheless, recognition of the human's action from this high-dimensional data set poses serious challenges.

In this paper, 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

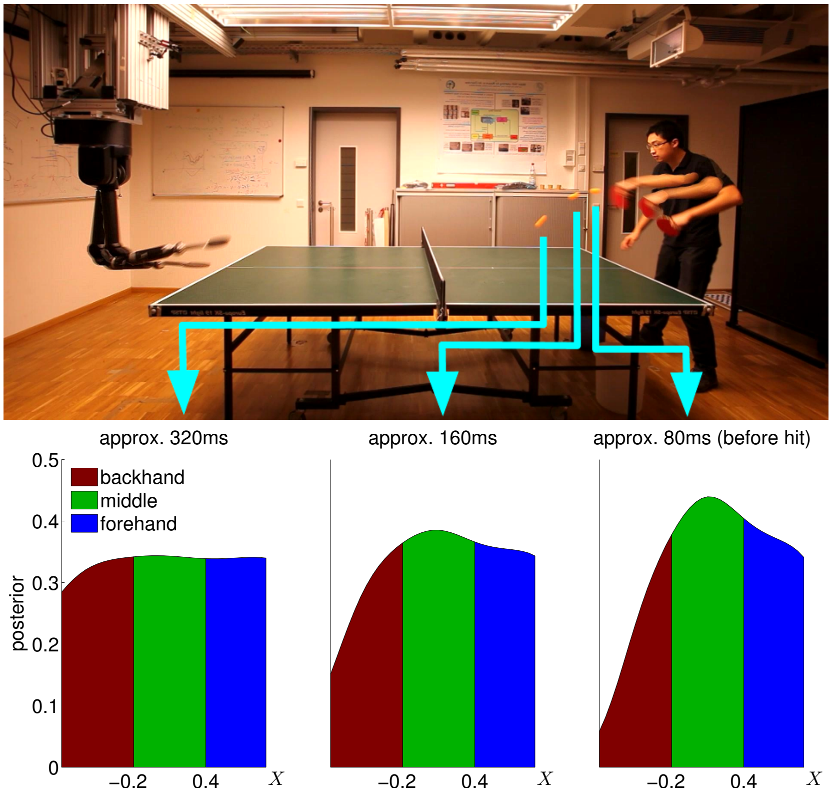


Figure 8: Bar plots show the distribution of the target (X coordinate) at approximately 320ms, 160ms, and 80ms before the player hits the ball. The prediction becomes more and more certain as the player finishes the stroke, and the robot later chose the middle hitting movement accordingly.

Table 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

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 processes 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[5], 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 the precision of 83.0% with significantly reduced computational 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, which became clearly separable while the human performed the actions.

The batch algorithm needs to choose the size of sliding windows, which influences both the accuracy and efficiency. As shown in Table 3, the batch algorithm could yield real-time action recognition in 3 Hz with a sliding window of size 5. The online algorithm, as shown in Table 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. (51), which describes how likely the type of actions is expected to change. We also found that the performance of online algorithm is not sensitive to this

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

parameter.

7 Discussion

In this article, we have proposed 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. Our contributions include: (1) suggesting the IDDM, which simultaneously finds a latent state representation of noisy and high-dimensional observations and models the dynamics that are driven by the intention; (2) introducing an online algorithm to efficiently infer the human's intention from an ongoing movement; (3) verifying the proposed model in two human-robot interaction scenarios. In particular, we have considered target inference in robot table tennis and action recognition for interactive robots. In these two scenarios, we show 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. 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.

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.

Acknowledgments

Part of the research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreements no. ICT-270327 (CompLACS) and no. ICT-248273 (GeRT). We thank Abdeslam Boularias for valuable discussions on this work.

References

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

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

Andrieu, C., De Freitas, N., Doucet, A., and Jordan, M. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 50(1):5–43.

Baker, C., Saxe, R., and Tenenbaum, J. (2009). Action understanding as inverse planning. *Cognition*, 113(3).

Baker, C., Tenenbaum, J., and Saxe, R. (2006). Bayesian models of human action understanding. In *Advances in Neural Information Processing Systems*.

Bandyopadhyay, T., Won, K. S., Frazzoli, E., Hsu, D., Lee, W. S., and Rus, D. (2012). Intention-aware motion planning. In *International Workshop on the Algorithmic Foundations of Robotics*.

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

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer New York.

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

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Damianou, A., Titsias, M., and Lawrence, N. (2011). Variational Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems*.

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

Deisenroth, M., Huber, M., and Hanebeck, U. (2009). Analytic moment-based Gaussian process filtering. In *International Conference on Machine Learning*.

Deisenroth, M. and Ohlsson, H. (2011). A general perspective on Gaussian filtering and smoothing. In *American Control Conference*.

Deisenroth, M., Turner, R., Huber, M., Hanebeck, U., and Rasmussen, C. (2012). Robust filtering and smoothing with Gaussian processes. *Trans. on Automatic Control*.

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.

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.

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

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

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

Huang, Y., Xu, D., Tan, M., and Su, H. (2013). Adding active learning to LWR for ping-pong playing robot. *IEEE Trans. on Control Systems Technology*, accepted for publication.

Ijspeert, A., Nakanishi, J., and Schaal, S. (2002). Movement imitation with nonlinear dynamical systems in humanoid robots. In *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.

Khan, M., Mohamed, S., Marlin, B., and Murphy, K. (2012). A stick-breaking likelihood for categorical data analysis with latent Gaussian models. In *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 & Systems*.

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

Lampert, C. H. 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*.

Lawrence, N. (2004). Gaussian process latent variable models for visualization of high dimensional data. In *Advances in Neural Information Processing Systems*.

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

Liao, L., Patterson, D., Fox, D., and Kautz, H. (2007). Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6).

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

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*, accepted for publication.

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

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

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 *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Quiñonero-Candela, J. and Rasmussen, C. (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. *Imitation and Social Learning in Robots, Humans, and Animals*.

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

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

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

Simon, M. (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*.

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

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. *The International Journal of Robotics Research*, 28(11-12):1486–1506.

Wang, J., Fleet, D., and Hertzmann, A. (2008). Gaussian process dynamical models for human motion. *IEEE Trans. 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 *International Conference on Machine Learning*.

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

Wang, Z., Deisenroth, M., Amor, H., Vogt, D., Schölkopf, B., and Peters, J. (2012b). Probabilistic modeling of human movements for intention inference. In *Proceedings of Robotics: Science and Systems*.

Wang, Z., Lampert, C., Mülling, K., Schölkopf, B., and Peters, J. (2011b). Learning anticipation policies for robot table tennis. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

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.

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

Ziebart, B., Dey, A., and Bagnell, J. (2012). Probabilistic pointing target prediction via inverse optimal control. In *ACM International Conference on Intelligent User Interfaces*, pages 1–10.

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

Ziebart, B., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J., Hebert, M., Dey, A., and Srinivasa, S. (2009). Planning-based prediction for pedestrians. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3931–3936.