

Probabilistic Modeling of Human Movements for Intention Inference

Zhikun Wang^{1,2}, Marc Peter Deisenroth², Heni Ben Amor², David Vogt³, Bernhard Schölkopf¹, Jan Peters^{1,2}

¹MPI for Intelligent Systems, Spemannstr. 38, 72076 Tübingen, Germany.

²TU Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany.

³TU Freiberg, Bernhard-von-Cotta Str. 2, 09596 Freiberg, Germany.

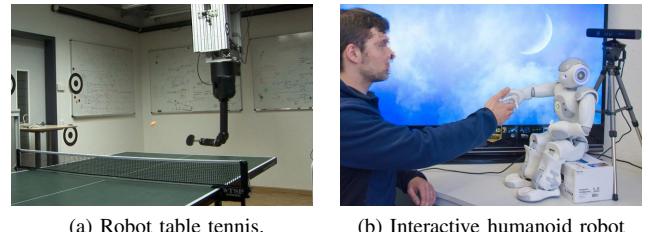
{zhikun, bs}@tuebingen.mpg.de {marc, amor, peters}@ias.tu-darmstadt.de

Abstract—Inference of human intention may be an essential step towards understanding human actions and is hence important for realizing efficient human-robot interaction. In this paper, we propose the Intention-Driven Dynamics Model (IDDM), a latent variable model for inferring unknown human intentions. We train the model based on observed human movements/actions. We introduce an efficient approximate inference algorithm to infer the human’s intention from an ongoing movement. We verify the feasibility of the IDDM in two scenarios, i.e., target inference in robot table tennis and action recognition for interactive humanoid robots. In both tasks, the IDDM achieves substantial improvements over state-of-the-art regression and classification.

I. 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 [22], which can enhance their ability to interact with humans. However, effective perception alone may not be sufficient for Human-Robot Interaction (HRI), since the robot’s reactions should depend on understanding the human’s action. An important understanding problem is inferring the others’ *intention* (also referred to as *goal*, *target*, *desire*, *plan*) [23], which humans heavily rely on, for example, in sports, games and social activities. Humans can learn and improve the ability of prediction by training. For example, skilled tennis players are usually trained to possess better anticipation than amateurs. This observation raises the question of how a robot can also learn to infer the underlying intention from human movements.

In order to infer the intention from an ongoing movement, we first address its inverse problem, i.e., modeling how the movement is governed by the intention. This idea is built upon the hypothesis that a human movement usually follows a goal-directed policy [4, 10]. The human movement considered here is represented by a time series of observations. This makes discrete-time dynamics models a straightforward choice for movement modeling and intention inference. In a robotic scenario, we often rely on high-dimensional and noisy sensor data. However, the intrinsic dimensionality is typically much smaller. Therefore, we seek a low-dimensional latent representation of the relevant information in the data, and then model how the intention governs the dynamics in this low-dimensional state space. Jointly considering both the low-dimensional representation and the latent dynamics leads to



(a) Robot table tennis.

(b) Interactive humanoid robot

Fig. 1: Two representative HRI scenarios where intention inference plays an important role: (a) target inference in robot table tennis games. (b) action recognition for human-robot interaction.

smooth trajectories in latent space with respect to the intention.

In this paper, we exploit this smoothness property and propose the Intention-Driven Dynamics Model (IDDM), in which the dynamic in the latent states is driven by the intention of the human action/behavior. The IDDM can simultaneously find a good low-dimensional representation of high-dimensional and noisy observations and describe the dynamics in the low-dimensional latent space. Using the learned model, human intention can be inferred from an ongoing movement. As exact inference is not tractable in our model, we propose an approximate inference algorithm and show that it can efficiently infer the intention of a human partner.

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, as shown in Fig. 1a. The robot’s hardware constraints impose strong limitations on its flexibility. It requires sufficient time to execute a ball-hitting plan: movement initiation to an appropriate preparation pose is needed *before* the opponent returns the ball, to achieve the required velocity for returning the ball [27]. The robot player uses different preparation poses for forehand and backhand hitting plans. Hence, it is necessary to choose between them based on inference of the opponent’s target location.

(2) *Action recognition for interactive humanoid robot.* In this setting, we use our technique to improve the interaction capabilities of a NAO humanoid robot, as shown in Fig. 1b. In

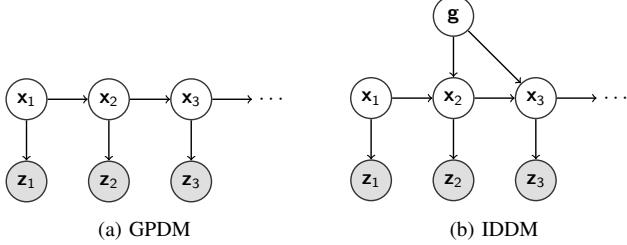


Fig. 2: Graphical models of the Gaussian process dynamical model (GPDM) and the proposed intention-driven dynamics model (IDDM). The proposed model explicitly incorporates the intention as an input to the transition function.

order to realize natural and compelling interactions, the robot needs to correctly recognize the actions of its human partner. This ability, in turn, allows it to act in a proactive manner. We show that the IDDM has the potential to identify the intention of action from movements in a simplified scenario.

The paper is organized as follows. We present the IDDM and address the problem of training in Section II. In Section III, we study efficient approximate algorithms for intention inference. We verify the feasibility of the proposed methods in the two scenarios in Section IV and V. Finally, we provide a brief review of related work in Section VI, and conclude in Section VII.¹

II. INTENTION-DRIVEN DYNAMICS MODEL

We propose the Intention-Driven Dynamics Model (IDDM), which is an extension of the Gaussian Process Dynamical Models (GPDM) [25]. The GPDM provides 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. 2a, the transition function in GPDM is only determined by the latent state. However, in the cases considered in this paper, the underlying intention, as an important drive of human movements, can hardly be discovered directly from the observations or the estimated latent states. Considering that the dynamics can be substantially different when the actions are based on different intentions, we propose the IDDM, the graphical model of which is shown in Fig. 2b. The IDDM explicitly incorporates the intention into the transition function in the latent state space. The dynamics model is inspired by the hypothesis that the human action is directed by the goal [4, 10]. For example, in table tennis, the player swings the racket in order to return the ball to the intended target.

The observations of a movement comprise a time series of observations $\mathbf{z}_{1:T} \triangleq [\mathbf{z}_1, \dots, \mathbf{z}_T]$. For notation simplicity and without loss of generality, we consider that all the observations of movements have the same length T . In the proposed generative model, we assume that the observation \mathbf{z}_i are

¹Supplementary technical details, demos and results can be found at <http://www.ias.tu-darmstadt.de/Research/ProbabilisticMovementModeling>

measured from the d -dimensional state space \mathcal{X} to the D -dimensional observation space \mathcal{Z} , given by

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

where $\mathbf{W} = \text{diag}(w_1, \dots, w_D)$, i.e., the i -th dimension of observations \mathbf{z}_t is the scaled i -th dimension of the outputs $\tilde{\mathbf{z}}_t$ with a parameter w_i . 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 Gaussian process (GP) prior distribution on every dimension of the unknown function \mathbf{h} [20], which can be marginalized out during learning and inference. A GP is fully specified by a mean function $m_z(\cdot)$ and a positive semidefinite covariance (kernel) function $k_z(\cdot, \cdot)$. The predictive probability of the observations \mathbf{z}_t is given by a Gaussian distribution $\mathbf{z}_t \sim \mathcal{N}(\mathbf{m}_z(\mathbf{x}_t), \Sigma_z(\mathbf{x}_t))$.

We consider first-order Markovian dynamics, see Fig. 2b, which is modeled by a latent transition function \mathbf{f} , given by

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

The state at time t is determined by the latent state at time $t-1$ as well as by the intention \mathbf{g} . We place a GP prior $\mathcal{GP}(m_x(\cdot), k_x(\cdot, \cdot))$ on every dimension of \mathbf{f} and marginalize it out. Then, the predictive distribution of the latent state \mathbf{x}_t conditioned on \mathbf{x}_{t-1} and the intention \mathbf{g} is a Gaussian distribution given by $\mathbf{x}_t \sim \mathcal{N}(\mathbf{m}_x(\mathbf{x}_{t-1}, \mathbf{g}), \Sigma_x(\mathbf{x}_{t-1}, \mathbf{g}))$.

To summarize, in the proposed IDDM, one set of GPs models the transition function in the latent space conditioned on the intention \mathbf{g} . A second set of GPs models the measurement mapping from the latent states \mathbf{x} and the observations \mathbf{z} .

A. Covariance Functions

For simplicity, we use GP prior mean functions that are zero everywhere, 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 with automatic relevance determination for the dynamics, i.e.,

$$k_x([\mathbf{x}, \mathbf{g}], [\mathbf{x}', \mathbf{g}']; \boldsymbol{\alpha}) = k_x(\mathbf{x}, \mathbf{x}'; \boldsymbol{\alpha})k_x(\mathbf{g}, \mathbf{g}'; \boldsymbol{\alpha}) \quad (3)$$

$$= \alpha_1 \exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|_{\Lambda_x}^2 - \frac{1}{2}\|\mathbf{g} - \mathbf{g}'\|_{\Lambda_g}^2\right) + \alpha_4^{-1}\delta_{\mathbf{x}\mathbf{x}'}\delta_{\mathbf{g}\mathbf{g}'},$$

where the shorthand notation $\|\mathbf{a}\|_{\Lambda}^2 \triangleq \mathbf{a}^T \Lambda \mathbf{a}$, the diagonal matrices $\Lambda_x = \text{diag}(\alpha_{21}, \dots, \alpha_{2d}) > \mathbf{0}$ and $\Lambda_g = \text{diag}(\alpha_{31}, \dots, \alpha_{3|g|}) > \mathbf{0}$ weight the corresponding input dimensions, $\boldsymbol{\alpha}$ 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 $\alpha_3 = \infty$ such that $k_x(g, g'; \boldsymbol{\alpha}) \equiv \delta_{g,g'}$.

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 [25] uses an isotropic Gaussian covariance function parameterized by the hyperparameters β

$$k_z(\mathbf{x}, \mathbf{x}'; \beta) = \exp\left(-\frac{\beta_1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) + \beta_2^{-1}\delta_{\mathbf{x}, \mathbf{x}'}, \quad (4)$$

as intuitively the latent states that generate human poses lie on a nonlinear manifold. Note that the hyperparameters β do not contain the signal variance, which is parameterized by the scaling factors \mathbf{W} . When applying it to target prediction in table tennis games, we use the linear kernel

$$k_z(\mathbf{x}, \mathbf{x}'; \beta) = \mathbf{x}^T \mathbf{x}' + \beta_1^{-1} \delta_{\mathbf{x}, \mathbf{x}'}, \quad (5)$$

as the observations are already low-dimensional, but subject to substantial noise. Empirical comparisons of different covariance functions for the measurement mapping, i.e., the isotropic Gaussian in Eq. (4) and the linear covariance function in Eq. (5), will be shown in Section IV.

B. Learning the IDDM

The proposed model can be learned from a training data set $\mathcal{D} = \{\mathbf{Z}, \mathbf{g}\}$ of J movements and corresponding intentions. Each movement \mathbf{Z}^j consists of a time series of observations given by $\mathbf{Z}^j = [\mathbf{z}_1^j, \dots, \mathbf{z}_T^j]^T$. We construct the overall observation matrix \mathbf{Z} by vertically concatenating observation matrices $\mathbf{Z}^1, \dots, \mathbf{Z}^J$, and the overall intention matrix \mathbf{g} from $\mathbf{g}^1, \dots, \mathbf{g}^J$. 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 \mathbf{g} can be obtained in the training data, 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. 3 for an illustration).

Similar to the work by [25], we find the *maximum a posteriori* (MAP) estimate of the latent states \mathbf{X} . Alternative learning methods and an empirical comparison can be found in [24, 5]. Given the model hyperparameters, the posterior distribution of latent states \mathbf{X} can be decomposed into the probability of observations given states and the probability of states given intention:

$$p(\mathbf{X}|\mathbf{Z}, \mathbf{g}, \alpha, \beta, \mathbf{W}) \propto p(\mathbf{Z}|\mathbf{X}, \beta, \mathbf{W})p(\mathbf{X}|\mathbf{g}, \alpha), \quad (6)$$

both obtained by the GP marginal likelihood [20].

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

$$\begin{aligned} p(\mathbf{Z}|\mathbf{X}, \beta, \mathbf{W}) \\ = \frac{|\mathbf{W}|^M}{\sqrt{(2\pi)^{MD} |\mathbf{K}_z|^D}} \exp\left(-\frac{1}{2} \text{tr}\left(\mathbf{K}_z^{-1} \mathbf{Z} \mathbf{W}^2 \mathbf{Z}^T\right)\right), \end{aligned} \quad (7)$$

where $M \triangleq JT$ is the length of observations \mathbf{Z} , and \mathbf{K}_z is the covariance matrix of \mathbf{X} computed by the kernel function $k_z(\cdot, \cdot)$.

Given the intention \mathbf{g} , the sequence of latent states \mathbf{X} has a Gaussian probability

$$\begin{aligned} p(\mathbf{X}|\mathbf{g}, \alpha) &= p(\mathbf{X}_1)p(\mathbf{X}_{2:T}|\mathbf{X}_{1:T-1}, \mathbf{g}, \alpha) \\ &= \frac{p(\mathbf{X}_1)}{\sqrt{(2\pi)^{md} |\mathbf{K}_x|^d}} \exp\left(-\frac{1}{2} \text{tr}\left(\mathbf{K}_x^{-1} \mathbf{X}_{2:T} \mathbf{X}_{2:T}^T\right)\right), \end{aligned} \quad (8)$$

where $\mathbf{X}_{\text{indices}}$ is constructed by vertically concatenated state matrices $\mathbf{x}_{\text{indices}}^1, \dots, \mathbf{x}_{\text{indices}}^J$, $m \triangleq J(T-1)$ is the length of $\mathbf{X}_{2:T}$, and \mathbf{K}_x is the covariance matrix of $\mathbf{X}_{1:T-1}$ computed

Algorithm 1: The algorithm learns the model hyperparameters α, β , and \mathbf{W} . They are obtained by maximizing the marginal likelihood using the Monte Carlo EM algorithm.

```

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

```

by the kernel function $k_x(\cdot, \cdot)$. We use a Gaussian prior distribution on the initial states \mathbf{X}_1 .

Based on Eqs. (7)-(8), the MAP estimate of the states is learned by maximizing the posterior in Eq. (6). We minimize the negative log-posterior

$$\begin{aligned} \mathcal{L}(\mathbf{X}) &= \frac{D}{2} \log |\mathbf{K}_z| + \frac{1}{2} \text{tr}(\mathbf{K}_z^{-1} \mathbf{Z} \mathbf{W}^2 \mathbf{Z}^T) - M \log |\mathbf{W}| \\ &+ \frac{d}{2} \log |\mathbf{K}_x| + \frac{1}{2} \text{tr}(\mathbf{K}_x^{-1} \mathbf{X}_{2:T} \mathbf{X}_{2:T}^T) + \frac{1}{2} \text{tr}(\mathbf{X}_1 \mathbf{X}_1^T) + \text{const} \end{aligned}$$

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

C. Learning Hyperparameters

A reliable approach to learning the hyperparameters $\Theta = \{\alpha, \beta, \mathbf{W}\}$ is to maximize the marginal likelihood $p(\mathbf{Z}|\mathbf{g}, \Theta) = \int p(\mathbf{Z}, \mathbf{X}|\mathbf{g}, \Theta) d\mathbf{X}$, which can be achieved approximately by using Expectation-Maximization (EM) algorithm. The EM algorithm computes the posterior distribution of states $q(\mathbf{X}) = p(\mathbf{X}|\mathbf{Z}, \mathbf{g}, \Theta)$ in the Expectation (E) step, and updates the hyperparameters by maximizing the expected data likelihood $\mathbb{E}_q[p(\mathbf{Z}, \mathbf{X}|\mathbf{g}, \Theta)]$ in the Maximization (M) step. However, the posterior distribution $q(\mathbf{X})$ is difficult to achieve in IDDM. Following [25], we draw samples of the states $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(I)}$ from the posterior distribution using Hybrid Monte Carlo (HMC) [2], and hence the data likelihood is estimated via Monte Carlo integration given by $\mathbb{E}_q[p(\mathbf{Z}, \mathbf{X}|\mathbf{g}, \Theta)] \approx \frac{1}{I} \sum_{i=1}^I p(\mathbf{Z}, \mathbf{X}^{(i)}|\mathbf{g}, \Theta)$. In the M step, we use SCG to update the hyperparameters. In practice, we choose the number of samples $I = 100$ and the number of EM iterations $L = 10$. Although this procedure, as described in Algorithm 1, is time demanding, this is not a big issue in practice as we learn the hyperparameters off-line.

The model also depends on the hyperparameter d : the dimensionality of the latent state space. Choosing an appropriate d is important. If the dimensionality is too small the latent states cannot recover the observations and, therefore, 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*, for example based on cross-validation, is helpful and should be conducted before learning and applying the model.

To summarize, the model $\mathcal{M} = \{\mathbf{X}, \Theta\}$ can be learned from a data set \mathcal{D} . It is used to infer the unobserved intention of a new ongoing movement.

III. APPROXIMATE INTENTION INFERENCE

After learning the model \mathcal{M} from the data set \mathcal{D} , the stationary intention \mathbf{g} can be inferred from a sequence of new observations $\mathbf{z}_{1:T}$ at test time. In this section, we omit the model \mathcal{M} and the data set \mathcal{D} for notation simplicity. Exact inference is not tractable as the intention \mathbf{g} is connected to all the unobserved latent states $\mathbf{x}_{1:T}$ in the graphical model. Therefore, we use EM algorithm to find the maximum-likelihood estimate (MLE) of the intention \mathbf{g}^* , given by

$$\mathbf{g}^* = \arg \max_{\mathbf{g}} p(\mathbf{z}_{1:T} | \mathbf{g}) = \int p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T} | \mathbf{g}) d\mathbf{x}_{1:T}, \quad (9)$$

which is obtained by maximizing the marginal likelihood with the latent states $\mathbf{x}_{1:T}$ integrated out. To find the MLE, we apply the EM algorithm.

In the *E step*, we calculate the posterior distribution of the latent states $p(\mathbf{x}_{1:T} | \mathbf{z}_{1:T}, \mathbf{g}^0)$ based on the current estimate of the intention \mathbf{g}^0 . This problem corresponds to the filtering and smoothing problems in nonlinear dynamical systems. For the GPDM and the IDDM, approximate filtering and smoothing is necessary, using particle filtering or Gaussian approximations for example. In [13] Particle filters (GP-PF), extended Kalman filters (GP-EKF), and unscented Kalman filters (GP-UKF) for approximate filtering with GPs were proposed. Assumed Density Filter (GP-ADF) for efficient GP filtering and smoothing based on moment matching were proposed in [8, 9]. We follow the work in [8, 9] as the approximated posterior distribution $q(\mathbf{x}_{1:T}) \approx p(\mathbf{x}_{1:T} | \mathbf{z}_{1:T}, \mathbf{g}^0)$ provides credible error bars, i.e., it is robust to incoherent estimates. We describe our approach to approximate filtering and smoothing in the intention-driven dynamics model in Section III-A.

In the *M step*, we update the intention based on approximated posterior distribution $q(\mathbf{x}_{1:T}) \approx p(\mathbf{x}_{1:T} | \mathbf{z}_{1:T}, \mathbf{g}^0)$, by maximizing the expected data log-likelihood

$$\begin{aligned} \arg \max_{\mathbf{g}} Q(\mathbf{g} | \mathbf{g}^0) &= \mathbb{E}_q [\log p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T} | \mathbf{g})] \\ &= \underbrace{\mathbb{E}_q [\log p(\mathbf{z}_{1:T} | \mathbf{x}_{1:T})]}_{Q_z} + \underbrace{\mathbb{E}_q [\log p(\mathbf{x}_{1:T} | \mathbf{g})]}_{Q_x(\mathbf{g})}, \end{aligned} \quad (10)$$

where Q_z is independent of the intention \mathbf{g} and hence can be omitted, and decomposition of $Q_x(\mathbf{g})$ leads to

$$\arg \max_{\mathbf{g}} Q_x(\mathbf{g}) = \arg \max_{\mathbf{g}} \sum_{t=1}^{T-1} Q_t(\mathbf{g}), \quad (11)$$

where

$$Q_t(\mathbf{g}) = \iint q(\mathbf{x}_t, \mathbf{x}_{t+1}) \log p(\mathbf{x}_{t+1} | \mathbf{x}_t, \mathbf{g}) d\mathbf{x}_{t+1} d\mathbf{x}_t. \quad (12)$$

This optimization problem can be solved approximately. We present the M step in detail in Sec III-B.

A. Filtering and Smoothing in the IDDM

Given a prior distribution $p(\mathbf{x}_1)$ and the current estimate of \mathbf{g} , we are interested in computing the posterior distribution $p(\mathbf{x}_{1:T} | \mathbf{z}_{1:T}, \mathbf{g}^0)$.

A Gaussian approximation of the joint distribution $p(\mathbf{x}_{1:T} | \mathbf{z}_{1:T}, \mathbf{g}^0)$ is computed explicitly by computing the marginals $p(\mathbf{x}_t | \mathbf{z}_{1:T}, \mathbf{g}^0)$ and the cross-covariances $p(\mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{z}_{1:T}, \mathbf{g}^0)$. These steps yield a Gaussian approximation with a block-tri-diagonal covariance matrix. For the computations, we employ forward-backward smoothing (GP-RTSS) in GPDM, see [9].

The computation of the joint distributions $p(\mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{g}^0)$ and $p(\mathbf{x}_t, \mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{g}^0)$ suffices for forward-backward smoothing in the IDDM, as the Gaussian filter/smoothing updates can be expressed solely in terms of means and (cross-)covariances of these joint distributions [7].

We outline the computations required for the Gaussian approximation of $p(\mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{g}^0)$ using moment matching; the Gaussian approximation of $p(\mathbf{x}_t, \mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{g}^0)$ follows analogously.

We approximate the joint distribution $p(\mathbf{x}_{t-1}, \mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{g}^0)$ by the Gaussian

$$\mathcal{N} \left(\begin{bmatrix} \boldsymbol{\mu}_{t-1|t-1}^x \\ \boldsymbol{\mu}_{t|t-1}^x \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{t-1|t-1}^x & \boldsymbol{\Sigma}_{t-1,t|t-1}^x \\ \boldsymbol{\Sigma}_{t,t-1|t-1}^x & \boldsymbol{\Sigma}_{t|t-1}^x \end{bmatrix} \right). \quad (13)$$

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

Without loss of generality, in Eq. (13), we assume that the marginal distribution $\mathcal{N}(\mathbf{x}_{t-1} | \boldsymbol{\mu}_{t-1|t-1}^x, \boldsymbol{\Sigma}_{t-1|t-1}^x)$ is known (corresponds to the filter distribution at time $t-1$). We compute the remaining elements of Eq. (13) explicitly using moment-matching.

Using iterated expectations, the a -th dimension of the mean of the marginal $p(\mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{g}^0)$ is

$$\begin{aligned} (\boldsymbol{\mu}_{t|t-1}^x)_a &= \mathbb{E}_{\mathbf{x}_{t-1}} [\mathbb{E}_{f_a} [f_a(\mathbf{x}_{t-1}, \mathbf{g}^0) | \mathbf{x}_{t-1}] | \mathbf{g}^0, \mathbf{z}_{1:t-1}] \\ &= \int m_x^a(\mathbf{x}_{t-1}, \mathbf{g}^0) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{g}^0) d\mathbf{x}_{t-1}, \end{aligned} \quad (14)$$

where we plugged in the posterior GP mean function, for the inner expectation. Writing out the posterior mean function and defining $\boldsymbol{\gamma}_a := \mathbf{K}_x^{-1} \mathbf{y}_a$, with $y_{a_i}, i = 1, \dots, M$, being the training targets of the GP with target dimension a , we obtain

$$(\boldsymbol{\mu}_{t|t-1}^x)_a = \mathbf{q}^\top \boldsymbol{\gamma}_a, \quad \text{where} \quad (15)$$

$$\mathbf{q}^T = \int k_x([\mathbf{x}_{t-1}, \mathbf{g}^0], \bar{\mathbf{X}}) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}, \mathbf{g}^0) d\mathbf{x}_{t-1}. \quad (16)$$

Here, $\bar{\mathbf{X}}$ denotes the set of the M GP training inputs $(\mathbf{x}_{ij}, \mathbf{g}_j)$, $i = 1, \dots, N, j = 1, \dots, J$. The intention \mathbf{g}_j is assumed stationary in a single trajectory $\mathbf{x}_{1j}, \dots, \mathbf{x}_{Nj}$. For notational convenience we assume that all training sequences are N time steps long.

If k_x is a Gaussian kernel, we can solve the integral in Eq. (16) analytically and obtain the vector \mathbf{q} with entries $q_i, i = 1, \dots, M$,

$$q_i = \sigma_f^2 |\Sigma_{t-1|t-1}^x \Lambda_x + \mathbf{I}|^{-\frac{1}{2}} \exp \left(-\frac{1}{2} \zeta_i^T \Omega^{-1} \zeta_i \right),$$

$$\zeta_i = \mathbf{x}_i - \mu_{t-1|t-1}^x, \quad \Omega = \Sigma_{t-1|t-1}^x + \Lambda_x^{-1}.$$

The computations of the (cross-)covariances $\Sigma_{t-1,t|t-1}^x$ and $\Sigma_{t|t-1}^x$ in Eq. (13) follow the same scheme—we have to solve integrals of a Gaussian prior $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, \mathbf{g}^0)$ times two Gaussian kernels (instead of a single one for the mean, see Eq. (16)). The computations are performed analytically but are omitted here. We refer to [8, 9] for details.

With the Gaussian kernel as in Eq. (4), a Gaussian approximation to the second joint $p(\mathbf{x}_t, \mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{g}^0)$ can be computed analogously. For the linear measurement kernel in Eq. (5), we can also solve the integration corresponding to Eq. (16) analytically and obtain

$$\mathbf{q} = \beta_1 \bar{\mathbf{X}} \mu_{t-1|t-1}^x. \quad (17)$$

Due to space restrictions, we omit further details, but the required computations are straightforward.

Following [7], we obtain the updates for the latent state posteriors (filter and smoothing distributions) as

$$\mu_{t|t}^x = \mu_{t|t-1}^x + \Sigma_{t|t-1}^{xz} (\Sigma_{t|t-1}^z)^{-1} (\mathbf{z}_t - \mu_{t|t-1}^z), \quad (18)$$

$$\Sigma_{t|t}^x = \Sigma_{t|t-1}^x - \Sigma_{t|t-1}^{xz} (\Sigma_{t|t-1}^z)^{-1} \Sigma_{t|t-1}^{zx}, \quad (19)$$

$$\mu_{t-1|T}^x = \mu_{t-1|t-1}^x + \mathbf{J}_{t-1} (\mu_{t|T}^x - \mu_{t|t-1}^x), \quad (20)$$

$$\Sigma_{t|T}^x = \Sigma_{t-1|t-1}^x + \mathbf{J}_{t-1} (\Sigma_{t|T}^x - \Sigma_{t|t-1}^x) \mathbf{J}_{t-1}^T, \quad (21)$$

where we have

$$\mathbf{J}_{t-1} = \Sigma_{t-1,t|t-1}^x (\Sigma_{t|t-1}^x)^{-1}. \quad (22)$$

These smoothing updates yield the marginals of $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, \mathbf{g}^0)$. The missing cross-covariances $\Sigma_{t-1,t|T}^x$ of $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, \mathbf{g}^0)$ that yield a block-tri-diagonal covariance matrix are

$$\Sigma_{t-1,t|T}^x = \mathbf{J}_{t-1} \Sigma_{t|T}^x, \quad (23)$$

where \mathbf{J}_{t-1} is given in Eq. (22). For more technical details, we refer to [6] and the supplementary material (see Footnote 1).

B. Maximization

With these computation, we obtain a Gaussian approximation to the posterior $q(\mathbf{x}_{1:T}) \approx p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, \mathbf{g}^0)$ with a block-tri-diagonal covariance matrix. The result

$$q(\mathbf{x}_t, \mathbf{x}_{t+1}) = \mathcal{N} \left(\begin{bmatrix} \mu_{t|T}^x \\ \mu_{t+1|T}^x \end{bmatrix}, \begin{bmatrix} \Sigma_{t|T}^x & \Sigma_{t,t+1|T}^x \\ \Sigma_{t+1,t|T}^x & \Sigma_{t+1|T}^x \end{bmatrix} \right) \quad (24)$$

is used in the M step. Here, we first consider continuous intention variables \mathbf{g} .

The integration in Eq. (12) can be rewritten as

$$\begin{aligned} \mathcal{Q}_t(\mathbf{g}) &= \iint q(\mathbf{x}_t, \mathbf{x}_{t+1}) \log(p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{g})q(\mathbf{x}_t)) d\mathbf{x}_{t+1} d\mathbf{x}_t \\ &\quad - \int q(\mathbf{x}_t) \log q(\mathbf{x}_t) d\mathbf{x}_t. \end{aligned} \quad (25)$$

Algorithm 2: The EM algorithm finds the maximum-likelihood estimate of continuous intention \mathbf{g} .

Input : Observations $\mathbf{x}_{1:T}$
Output: MLE of the continuous intention \mathbf{g}

1 Initialize \mathbf{g} ;
2 **repeat**
3 $\mathbf{g}^0 \leftarrow \mathbf{g}$;
4 E step: approximate $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, \mathbf{g}^0)$ with $q(\mathbf{x}_{1:T})$ by forward-backward smoothing (GP-RTSS) ;
5 M step: update \mathbf{g} by optimizing (11) using SCG with J steps, based on Eq. (26) ;
6 **until** $\|\mathbf{g} - \mathbf{g}^0\| < \epsilon$;

We can approximate $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{g})q(\mathbf{x}_t)$ with a joint Gaussian distribution using the same technique as in the E step, and obtain $\tilde{q}(\mathbf{x}_t, \mathbf{x}_{t+1}|\mathbf{g}) = \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})$, as well as gradients $\partial \tilde{\mu} / \partial \mathbf{g}$ and $\partial \tilde{\Sigma} / \partial \mathbf{g}$. As a result, the Eq. (12) can be approximated as

$$\mathcal{Q}_t(\mathbf{g}) \approx D_{\text{KL}}(q_{t,t+1} || \tilde{q}_{t,t+1}) + H(q_{t,t+1}) + H(q(\mathbf{x}_t)), \quad (26)$$

with shorthand notation $q_{t,t+1} \triangleq q(\mathbf{x}_t, \mathbf{x}_{t+1})$ and $\tilde{q}_{t,t+1} \triangleq \tilde{q}(\mathbf{x}_t, \mathbf{x}_{t+1}|\mathbf{g})$. In Eq. (26), the entropy of a Gaussian distribution and the KL divergence between two Gaussians can be computed analytically. Therefore, we can analytically compute the approximate value of $\mathcal{Q}_t(\mathbf{g})$ in Eq. (26) and the corresponding gradients with respect to the intention \mathbf{g} . The M step, the optimization problem in (10) and (11), is then solved using SCG. The Algorithm 2 describes the inference algorithm for continuous intentions \mathbf{g} . We set the number of SCG steps to $J = 20$ in experiments.

When the intention \mathbf{g} is a discrete variable, for example the type of action, a more efficient algorithm is to enumerate all the possible value of \mathbf{g} and choose the one with the maximum approximated value (Jensen's lower bound) of the log marginal likelihood in Eq. (9), given by

$$\log p(\mathbf{z}_{1:T}|\mathbf{g}) \approx H(q(\mathbf{x}_{1:T})) + \mathbb{E}_q [\log p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}|\mathbf{g})], \quad (27)$$

where $q(\mathbf{x}_{1:T}) \approx p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, \mathbf{g})$ is the approximated posterior distribution given the current value of intention \mathbf{g} .

To summarize, we proposed an efficient approximate approach to intention inference from a new movement. The method can deal with both continuous (Algorithm 2) and discrete (Eq. (27)) intention variables.

IV. ROBOT TABLE TENNIS

Playing table tennis is a challenging task for robots, as it requires accurate prediction of the ball's movement and very fast response. Hence, robot table tennis has been used by many groups as a benchmark task in robotics [16, 17]. Thus far, none of the groups which have worked on robot table tennis ever got to the levels of a young child despite having robots that could see and move faster and more accurate than humans [17]. 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 [17]. In

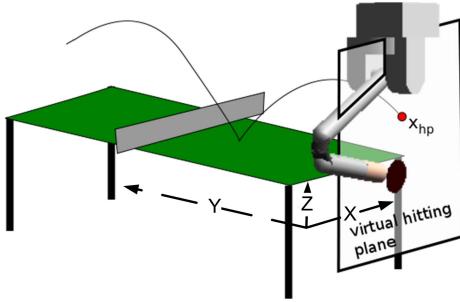


Fig. 3: The robot’s hitting point is the intersection of the coming ball’s trajectory and the virtual hitting plane 80 cm behind the table.

this paper, we use a Barrett WAM robot arm to play table tennis against human players. The robot’s hardware constraints impose strong limitations on its flexibility.

The robot requires sufficient time to execute a ball-hitting plan: to achieve the required velocity for returning the ball, movement initiation to an appropriate preparation pose is needed before the opponent hits the ball. The robot player uses different preparation poses for forehand and backhand hitting plans. Hence, it is necessary to choose between them based on the modeling the opponent’s preference [26] and inference of the opponent’s target location for the ball [27].

The robot perceives the ball and the opponent’s racket in real-time, using seven Prosilica GE640C cameras [27]. These cameras are 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. The racket tracking system provides the information of the opponent’s racket, i.e., position and orientation. As a result, the observation \mathbf{z}_t includes the ball’s position and velocity, and the opponent’s racket position, velocity, and orientation *before* the human plays the ball. We downsample the observations at 12Hz. Here, the position and velocity of the ball are processed online with an extended Kalman filter. However, the same smoothing method cannot be applied to the racket’s trajectory, as its dynamics are unknown. Therefore, the obtained states of the racket are subject to substantial noise and the model has to be robust to this noise.

The robot always chooses its hitting point on a virtual hitting plane (80 cm behind the table), as shown in Fig. 3. We define the human’s intended target g as the intersection of the returned ball’s trajectory with the virtual hitting plane. As the X -coordinate (see Fig. 3) is most important for choosing either forehand or backhand hitting plans, 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 wide).

To evaluate the performance of target prediction, we have 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 with 100 plays and a test set with 126 plays. The standard deviation of the target coordinate in the test set is 102.2 cm. A

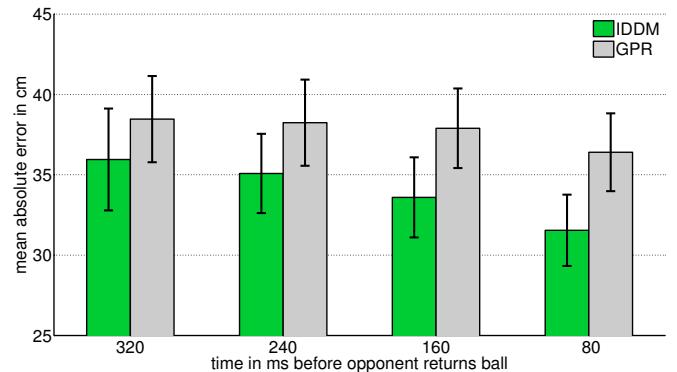


Fig. 4: Mean absolute error of the ball’s target with standard error of the mean. These errors are *before* the opponent has hit the ball and only based on his movements.

straightforward approach to prediction is to learn a mapping from the features \mathbf{z}_t to the target g . Hence, we compare our model to Gaussian Process Regression (GPR) using a Gaussian kernel with automatic relevance determination.

For every recorded play, we compare the performance of the proposed IDDM intention inference and the GPR intention (forward) prediction at 80 ms, 160 ms, 240 ms and 320 ms *before* the opponent hits the ball. Note that this time-step is only used such that the algorithms can be compared, and that the algorithms are not aware of the hitting time of the opponent in advance. As demonstrated in Fig. 4, the proposed model outperforms the GPR. At 80 ms before the opponent hits the ball, the proposed model results in the mean absolute error of 31.55 cm, which achieves a 13.3% improvement over the GPR (p-value: 0.016), whose average error is 36.4 cm. One model-free naive intention prediction would be to always predict the center of the intentions in the training set. This naive prediction model causes an error of 81.9 cm. Hence, both the GPR model and the IDDM substantially outperform naive goal prediction.

Note that the prediction is made before the opponent hits the ball, and only used to choose between forehand and backhand hitting plans. More fine-tuning can follow later when the returned ball’s trajectory has been observed. Hence, a certain amount of error is tolerable since the robot can apply small changes to its hitting plan based on the ball’s trajectory. However, it cannot switch between forehand and backhand hitting plans because of torque/temporal constraints.

We perform *model selection* to determine the covariance function k_z , which can be either an isotropic Gaussian kernel, see Eq. (4), or a linear covariance function, see Eq. (5). Furthermore, we perform model selection to find the dimension d of the latent states. In the experiments, the model was selected by cross-validation on the training set. The best model under consideration is with a linear covariance function and four

TABLE I: The mean absolute errors (in cm) of the goal inference made 80 ms *before* the opponent hits the ball.

kernel	$d = 3$	$d = 4$	$d = 5$	$d = 6$
linear	41.52	31.55	35.36	37.04
Gaussian	38.49	34.16	34.44	37.28

dimensional latent state space. Experiments on the test set verified the model selection result, as shown in Table I.

Our results demonstrate that the IDDM can improve the target prediction in robot table tennis and choose the correct hitting plan. We have verified the model in a simulated environment, but using data from real human movements recorded from a human playing against another human. The simulation shows that the robot can successfully return the ball when given a prediction by the IDDM model. For a demo, see Footnote 1.

V. ACTION RECOGNITION FOR INTERACTIVE ROBOTS

To realize safe and meaningful HRI, it is important that robots can recognize the human’s action. The advent of robust, marker-less motion capture techniques [22] has provided us with the technology to record the full skeletal configuration of the human during HRI. Yet, 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 recognize the intention of action from movements in a simplified scenario. Using a Kinect camera, we recorded the 32-dimensional skeletal configuration of a human during the execution of a set of actions namely: crouching (C), jumping (J), kick-high (KH), kick-low (KL), defense (D), punch-high (PH), punch-low (PL), and turn-kick (TK). For each type of action we collect a training set consisting of ten repetitions and a test set of three repetitions. The system downsamples the output of Kinect and processes three skeletal configurations per second. The actions were performed slowly in the data set, for example one jumping action takes about 1.5 seconds, so that the NAO robot has sufficient time to respond.

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 compare the proposed algorithm to Support Vector Machines (SVM) [21] and multi-class Gaussian Process Classification (GPC) [20, 12].

The algorithms make a prediction after observing a new skeletal configuration. The proposed IDDM uses a sliding window of length $n = 5$, i.e., it recognizes actions based on the recent n observations. We choose the IDDM with a linear covariance function for k_z and a two-dimensional state space. The IDDM achieves the precision of 83.8%, which outperforms SVM (77.5%) and GPC (79.4%) using the same sliding windows. We observed that the SVM and GPC confuse between crouching and jumping, as they are similar in the early and late stages. In contrast, the IDDM can distinguish crouching (C) and jumping (J) from their different dynamics, as shown in Fig. 5. The distinction between C and J becomes significant while the human performs the actions: The longer the movement is observed the more evidence is used by the IDDM.

TABLE II: Trade-off between accuracy and time complexity, by varying the size of sliding windows n .

n	4	5	6
Time(s)	0.27	0.32	0.39
Accuracy	79.0%	83.8%	86.6%

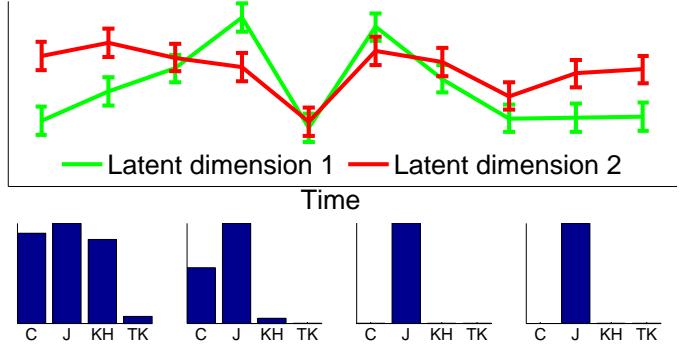


Fig. 5: The trajectories of the 2D latent states for two consecutive Jumping actions are obtained by smoothing. The error bars represent the corresponding standard deviations. The bar charts correspond to the likelihood of Crouching, Jumping, Kick-High and Turn-Kick at different stages of an action.

In order to apply the IDDM to real-time action recognition, we need to trade-off between accuracy and time complexity, by varying the size of sliding windows. As shown in Table II, the proposed model can yield real-time action recognition in 3Hz with a little sacrifice in accuracy. Note that this is only a preliminary experiment to show the feasibility of IDDM in action recognition. We will actively work on more efficient algorithms and implementations. For future experimental results and a demo, see Footnote 1.

VI. RELATED WORK

Inference of intention, or referred to in some contexts as goal, target, desire, plan, etc., has been investigated under different settings. For example, an early work [18] used Hidden Markov Models to model and predict human behavior where different dynamics models are adopted to the corresponding behaviors. Bayesian models were used for inferring goals from behavior in [19], where a policy conditional on the agent’s goal is learned to represent the behavior. Bayesian models can also interpret the agent’s behavior and predict its behavior in a similar environment with the learned model [3]. Inverse Reinforcement Learning (IRL) [1] is also worth mentioning. IRL assumes a rational agent that maximizes expected utility, and infers the underlying utility function from its behavior. In a recent work [10], a computational framework is proposed to model gaze following, where GPs are used to model the policies with actions directed by a goal.

Observation of human movement often consists of high-dimensional features. Determining the low-dimensional latent state space is an important issue for understanding observed actions. Gaussian Process Latent Variable Model (GPLVM) [15] finds the most likely latent variables by marginalizing out the mapping function from latent to observed space. Its extension, Gaussian Process Dynamical Models [25] can be used to model the dynamics of human motion while simultaneously finding low-dimensional latent states. For example, it can model and extrapolate the appearance of human walking.

Nonlinear dynamics models have successful applications in robotics. For example, Dynamic Motion Primitives [11]

were used in imitation learning, parametrizing the dynamic as differential equations to achieve robust dynamics and fast learning. Dynamics models are also helpful in tracking, for example, a small robotic blimp using two cameras [13], where GP-Bayes filters were proposed for efficient filtering. In a following work, the model is learned using GPLVM [14], so that the latent states need not be provided for learning.

VII. DISCUSSIONS

In this paper, we proposed the intention-driven dynamics model (IDDM). Our contributions include: (1) suggesting the IDDM, which simultaneously finds a good low-dimensional representation of high-dimensional and noisy observations, and models the dynamics that are driven by the intention; (2) introducing an approximate algorithm to efficiently infer the human's intention from an ongoing movement; (3) verifying the proposed model in two human-robot interaction scenarios, i.e., 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 prediction than algorithms without modeling dynamics.

There are many interesting directions that we are actively exploring. The model suffers somewhat from its computational complexity. The acceleration of the inference algorithm for its real-time application in robot table tennis can be achieved by a more efficient implementation, e.g. using parallel computing. Moreover, the discovery of action types without supervision can be interesting for HRI.

ACKNOWLEDGMENTS

Part of the research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement # 231495. We thank Abdeslam Boularias for valuable discussions on this work.

REFERENCES

- [1] P. Abbeel and A.Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *ICML*, 2004.
- [2] C. Andrieu, N. De Freitas, A. Doucet, and M.I. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1):5–43, 2003.
- [3] C. Baker, J. Tenenbaum, and R. Saxe. Bayesian models of human action understanding. In *NIPS*, 2006.
- [4] C.L. Baker, R. Saxe, and J.B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3), 2009.
- [5] A.C. Damianou, M.K. Titsias, and N.D. Lawrence. Variational gaussian process dynamical systems. In *NIPS*, 2011.
- [6] M.P. Deisenroth. *Efficient Reinforcement Learning using Gaussian Processes*. KIT Scientific Publ., 2010.
- [7] M.P. Deisenroth and H. Ohlsson. A general perspective on Gaussian filtering and smoothing. In *ACC*, 2011.
- [8] M.P. Deisenroth, M.F. Huber, and U.D. Hanebeck. Analytic moment-based Gaussian process filtering. In *ICML*, 2009.
- [9] M.P. Deisenroth, R. Turner, M. Huber, U.D. Hanebeck, and C.E. Rasmussen. Robust filtering and smoothing with Gaussian processes. *Trans. on Automatic Control*, 2012.
- [10] A.L. Friesen and R.P.N. Rao. Gaze following as goal inference: A Bayesian model. In *CogSci*, 2011.
- [11] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *ICRA*, 2002.
- [12] M.E. Khan, S. Mohamed, B.M. Marlin, and K.P. Murphy. A stick-breaking likelihood for categorical data analysis with latent Gaussian models. In *AISTATS*, 2012.
- [13] J. Ko and D. Fox. GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *Autonomous Robots*, 27(1), 2009.
- [14] J. Ko and D. Fox. Learning GP-BayesFilters via Gaussian process latent variable models. *Autonomous Robots*, 2009.
- [15] N.D. Lawrence. Gaussian process latent variable models for visualization of high dimensional data. In *NIPS*, 2004.
- [16] M. Matsushima, T. Hashimoto, M. Takeuchi, and F. Miyazaki. A learning approach to robotic table tennis. *IEEE Trans. on Robotics*, 21(4), 2005.
- [17] K. Mülling, J. Kober, and J. Peters. A biomimetic approach to robot table tennis. *Adaptive Behavior*, 19(5):359–376, 2011.
- [18] A. Pentland and A. Liu. Modeling and prediction of human behavior. *Neural Computation*, 11(1), 1999.
- [19] R.P.N. Rao, A.P. Shon, and A.N. Meltzoff. A Bayesian model of imitation in infants and robots. *Imitation and Social Learning in Robots, Humans, and Animals*, 2004.
- [20] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [21] B. Schölkopf and A.J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001. ISBN 0262194759.
- [22] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.
- [23] M.A. Simon. *Understanding Human Action: Social Explanation and the Vision of Social Science*. State University of New York Press, 1982.
- [24] R. Turner, M.P. Deisenroth, and C.E. Rasmussen. State-space inference and learning with Gaussian processes. In *AISTATS*, 2010.
- [25] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2008.
- [26] Z. Wang, A. Boularias, K. Mülling, and J. Peters. Balancing safety and exploitability in opponent modeling. In *AAAI*, 2011.
- [27] Z. Wang, C.H. Lampert, K. Mülling, B. Schölkopf, and J. Peters. Learning anticipation policies for robot table tennis. In *IROS*, 2011.