

A Probabilistic Representation for Dynamic Movement Primitives

Franziska Meier^{1,2} and Stefan Schaal^{1,2}

¹CLMC Lab, University of Southern California, Los Angeles, USA

² Autonomous Motion Department, MPI for Intelligent Systems, Tübingen, Germany *

Abstract

Dynamic Movement Primitives have successfully been used to realize imitation learning, trial-and-error learning, reinforcement learning, movement recognition and segmentation and control. Because of this they have become a popular representation for motor primitives. In this work, we showcase how DMPs can be reformulated as a probabilistic linear dynamical system with control inputs. Through this probabilistic representation of DMPs, algorithms such as Kalman filtering and smoothing are directly applicable to perform inference on proprioceptive sensor measurements during execution. We show that inference in this probabilistic model automatically leads to a feedback term to online modulate the execution of a DMP. Furthermore, we show how inference allows us to measure the likelihood that we are successfully executing a given motion primitive. In this context, we show initial results of using the probabilistic model to detect execution failures on a simulated movement primitive dataset.

Introduction

One of the main challenges towards autonomous robots remains autonomous motion generation. A key observation has been that in certain environments, such as households, tasks that need to be executed tend to contain very repetitive behaviors (Tenorth, Bandouch, and Beetz 2009). Thus, the idea of identifying motion primitives to form building blocks for motion generation has become very popular (Ijspeert et al. 2013; Paraschos et al. 2013). A popular and effective way to learn these motion primitives is through imitation learning (Schaal 1999).

Part of the research in this area is concerned with motion representation, and a variety of options have been proposed (Ijspeert et al. 2013; Khansari-Zadeh and Billard 2010; Dragan et al. 2015; Wilson and Bobick 1999; Paraschos et al. 2013). However, recently, there has been interest in going beyond pure motion representation. In the context of closing action-perception loops, motor skill representations that allow to close that loop are attractive. The ability to feed back

sensory information during the execution of a motion primitive promises a less error prone motion execution framework (Pastor et al. 2011). Furthermore, the ability to associate a memory of how a motor skill should feel during execution can greatly enhance robustness in execution. This concept of enhancing a motor skill representation with a sensory memory has been termed associative skill memories (ASMs) (Pastor et al. 2012).

The idea of associative skill memories lies in leveraging the repeatability of motor skills to associate experienced sensor information with the motion primitive. This allows to adapt previously learned motor skills when this primitive is being executed again and deviates from previously experienced sensor information (Pastor et al. 2011). Thus far, existing approaches (Pastor et al. 2011; Gams et al. 2014) use the concept of coupling terms to incorporate sensor feedback into DMPs which typically involves setting task-dependent gains to modulate the effect of the feedback term.

In this work we endow Dynamic Movement Primitives with a probabilistic representation, that maintains their original functionality, but allows for uncertainty propagation. Inference in this probabilistic DMP model is realized through Kalman filtering and smoothing. An interesting component of probabilistic DMPs is the fact that – given a reference signal to track – sensor feedback is automatically considered when executing a desired behavior. As a result, the feedback term is an intrinsic part of this probabilistic formulation. Similar to Kalman filtering, the feedback term is now scaled based on a uncertainty based gain matrix. Besides this insight, we also highlight the benefits of combining a probabilistic motion primitive representation with a strong structural prior – which is given by the DMP framework in this work. We demonstrate the advantage of this by showing how to make use of the probabilistic model to perform failure detection when executing a motion primitive.

This paper is organized as follows: We start by reviewing the DMP framework and related probabilistic motion representations in the background section. Then in the main section we propose a graphical model representation for DMPs, detail the learning procedure and illustrate their usage. Finally, we discuss how of failure detection can be realized with probabilistic DMPs and evaluate it in that context.

*This research was supported in part by National Science Foundation grants IIS-1205249, IIS-1017134, EECS-0926052, the Office of Naval Research, the Okawa Foundation, and the Max-Planck-Society.

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Background

Dynamic Movement Primitives (DMPs) encode a desired movement trajectory in terms of the attractor dynamics of nonlinear differential equations (Ijspeert et al. 2013). For a 1 DOF system, the equations are given as:

$$\begin{aligned}\frac{1}{\tau}\dot{z} &= \alpha_z(\beta_z(g-p) - z) + sf(x) \\ \frac{1}{\tau}\dot{p} &= z\end{aligned}\quad (1)$$

such that $p, \dot{p}, \ddot{p} = \dot{z}$ are position, velocity, and acceleration of the movement trajectory, where

$$f(x) = \frac{\sum_{i=1}^N \psi_i w_i x}{\sum_{i=1}^N \psi_i}, \text{ with } \psi_i = \exp(-h_i(x - c_i)^2)$$

with

$$\frac{1}{\tau}\dot{x} = -\alpha_x x \text{ and } s = \frac{g - p_0}{g_{fit} - p_{0,fit}} = \frac{g - p_0}{\Delta g}$$

In general, it is assumed that the duration τ and goal position g are known. Thus, given τ and g the DMP is parametrized through weights $\mathbf{w} = (w_1, \dots, w_N)^T$ which are learned to represent the shape of any smooth movement. During this fitting process, the scaling variable s is set to one, and the value of Δg is stored as a constant for the DMP.

Probabilistic Motion Primitive Representations

The benefits of taking a probabilistic approach to motion representation has been discussed by a variety of authors (Toussaint 2009; Rückert et al. 2013; Meier et al. 2011; Calinon et al. 2012; Paraschos et al. 2013; Khansari-Zadeh and Billard 2011). The use of probabilistic models varies however. For instance, the approaches presented in (Calinon et al. 2012; Khansari-Zadeh and Billard 2011) take a dynamical systems view that utilizes statistical methods to encode variability of the motion. In contrast, (Toussaint 2009; Rückert et al. 2013) take a trajectory optimization approach using a probabilistic planning system.

Here, as outlined in the introduction, we take the dynamical systems view, as a first step towards an implementation of associate skill memories. In previous work (Meier et al. 2011) we have shown how we can reformulate the DMP equations into a linear dynamical system (Bishop 2006). Inference and learning in this linear dynamical system was formulated as a Kalman filtering/smoothing approach. This Kalman filter view of DMPs allowed us to perform online movement recognition (Meier et al. 2011) and segmentation of complex motor skills into underlying primitives (Meier, Theodorou, and Schaal 2012).

Finally, compared to previous work on probabilistic motion primitives, such as (Calinon et al. 2012; Paraschos et al. 2013), this representation explicitly represents the dynamical system structure as a dynamic graphical model and adds the possibility of considering sensor feedback as part of the inference process.

Probabilistic Dynamic Movement Primitives

In this section we introduce a new *Probabilistic Dynamic Movement Primitive* model. The goal of this work is to essentially provide a probabilistic model that can replace the standard non-probabilistic representation without loss in functionality. Thus, here we aim at deriving a graphical model that explicitly maintains positions, velocities and accelerations, such that a rollout of that model creates a full desired trajectory.

Deriving the Probabilistic DMP Model

We start out by deriving the new formulation and then show how learning of the new probabilistic DMPs is performed. The transformation system of a 1-DOF DMP can be discretized via Euler discretization, resulting in

$$\dot{z}_t = \tau(\alpha_z(\beta_z(g - p_{t-1}) - z_{t-1}) + f) \quad (2)$$

$$z_t = \dot{z}_t \Delta t + z_{t-1} \quad (3)$$

$$\dot{p}_t = \tau \dot{z}_t \quad (4)$$

$$\dot{p}_t = \tau \dot{z}_t \quad (5)$$

$$p_t = \dot{p}_t \Delta t + p_{t-1} \quad (6)$$

where Δt is the integration step size, and p_t, \dot{p}_t and \ddot{p}_t are position velocity and acceleration at time step t . by plugging in Equations 2, 3 into Equations 4, 5 and setting $\dot{z}_t = \frac{1}{\tau}\dot{p}_t$ and $z_t = \frac{1}{\tau}\dot{p}_t$ this can be reduced to

$$\ddot{p}_t = \tau^2(\alpha_z(\beta_z(g - p_{t-1}) - \frac{1}{\tau}\dot{p}_{t-1}) + f) \quad (7)$$

$$\dot{p}_t = \ddot{p}_t \Delta t + \dot{p}_{t-1} \quad (8)$$

$$p_t = \dot{p}_t \Delta t + p_{t-1} \quad (9)$$

By collecting p_t, \dot{p}_t and \ddot{p}_t into state $s_t = (\ddot{p}_t \quad \dot{p}_t \quad p_t)^T$ we can summarize this linear system of equations as

$$s_t = \mathbf{A}s_{t-1} + \mathbf{B}u_t \quad (10)$$

where the control input is given as $u_t = \alpha_z \beta_z g + s f_t$ and the transition matrix \mathbf{A} and control matrix \mathbf{B} are given as

$$\mathbf{A} = \begin{pmatrix} 0 & -\alpha_z \tau & -\alpha_z \beta_z \tau^2 \\ \Delta t & 1.0 & 0 \\ 0 & \Delta t & 1.0 \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

We like to account for two sources of uncertainty: transition noise, modeling any uncertainty of transitioning from one state to the next one; and observation noise, modeling noisy sensory measurements. Next we show how to incorporate both of these to arrive at the full probabilistic model.

A standard approach to modeling transition noise would be to assume additive, zero mean, Gaussian noise, eg

$$s_t = \mathbf{A}s_{t-1} + \mathbf{B}u_{t-1} + \varepsilon \quad (11)$$

with $\varepsilon = \mathcal{N}(\varepsilon | 0, \mathbf{Q})$ which would create a time-independent transition uncertainty. Here, however, we would like to model the transition uncertainty as a function of how certain we are about our non-linear forcing term f . Remember, f represents the shape of the motion primitive and is typically trained via imitation learning. Assuming that we several demonstrations

to learn from, we can estimate the mean forcing term f from the demonstrations, but also the hvariance, using Bayesian regression. Thus, for now, we assume that we have a predictive distribution over f_t , the non-linear forcing term at time step t

$$f_t \sim \mathcal{N}(f_t | \mu_{f_t}, \sigma_{f_t}^2) \quad (12)$$

with mean μ_{f_t} and variance $\sigma_{f_t}^2 t$. The details of deriving this distribution are given in the next Subsection. Assuming f_t to be drawn from a Gaussian distribution, automatically implies that the hidden state s_t is also Gaussian distributed

$$s_t \sim \mathcal{N}(s_t | \mathbf{A}s_{t-1} + \mathbf{B}(\alpha_z \beta_z g + s \mu_{f_t}), Q_t) \quad (13)$$

where $Q_t = s^2 \mathbf{B} \sigma_{f_t}^2 \mathbf{B}^T$. Thus, if the variance of the distribution over f_t is time-dependent, so is the state transition noise. Finally, we also want to be able to include noisy sensor measurements in our model. Thus, we assume that we receive observations o_t that are a function of the hidden state, corrupted by zero mean Gaussian noise

$$o_t \sim h(s_t) + \mathcal{N}(v | 0, R) \quad (14)$$

A simple example is for instance, the feedback on the actual position of the system. In this case, the observation function would be $h(s_t) = \mathbf{C}s_t$. with observation matrix $\mathbf{C} = (0 \ 0 \ 1)^T$.

Finally, putting it all together, our probabilistic formulation takes to form of a controlled linear dynamical system, with time dependent transition noise, and time-independent observation noise:

$$\begin{aligned} s_t &= \mathbf{A}s_{t-1} + \mathbf{B}(\alpha_z \beta_z g + s \mu_{f_t}) + \varepsilon_t \\ o_t &= h(s_t) + v \end{aligned}$$

with $\varepsilon_t \sim \mathcal{N}(\varepsilon_t | 0, s^2 \mathbf{B} \sigma_{f_t}^2 \mathbf{B}^T)$ and $v \sim \mathcal{N}(v | 0, R)$.

Note, for clarity, we have derived the probabilistic model for a specific DMP variant (Ijspeert et al. 2013). However, the same procedure can be followed to arrive at a dynamic graphical model representation for other variants of motion primitives, such as (Pastor et al. 2009).

Learning Probabilistic DMPs

Probabilistic Dynamic Movement Primitives can be learned through imitation learning similar to regular DMPs. Given K demonstrations P_{demo}^k of the same motion primitive, it is possible to estimate the distribution over the non-linear function term f via Bayesian regression (Bishop 2006).

The standard approach to estimating the noise covariance \mathbf{R} of a linear dynamical system is based on the *expectation-maximization*(EM) procedure. The EM algorithm iterates between estimating the posterior of the hidden state of the motion primitive, and maximizing the expected complete log likelihood with respect to the parameter of interest. The complete data log likelihood for K demonstrations is given by:

$$\begin{aligned} \ln p(P, \mathbf{S} | \tau, g) &= \sum_{k=1}^K \sum_{t=1}^T \ln \mathcal{N}(o_t^k | \mathbf{C}s_t^k, \mathbf{R}) + \ln \mathcal{N}(s_1^k | 0, \mathbf{Q}_0) \\ &+ \sum_{k=1}^K \sum_{t=2}^T \ln \mathcal{N}(s_t^k | \mathbf{A}s_{t-1}^k + \mathbf{B}u_{t-1}^k, \mathbf{Q}_t) \end{aligned} \quad (15)$$

```

1: procedure ROLLOUT( $w, \mathbf{Q}, \mathbf{R}$ )
2:   for  $t = 1 : T$  do
3:      $\mu_{p,t} = A\mu_{u,t-1} + Bu_{t-1}$ 
4:      $V_{p,t} = AV_{u,t-1}A^T + Q$ 
5:   end for
6: end procedure

```

Figure 1: Probabilistic DMP: rollout

```

1: procedure EXECUTEANDMONITOR( $w, \mathbf{Q}, \mathbf{R}$ )
2:   for  $t = 1 : T$  do
3:      $\mu_{p,t} = A\mu_{u,t-1} + Bu_{t-1}$ 
4:      $V_{p,t} = AV_{u,t-1}A^T + Q$ 
5:      $S = CV_{p,t}C^T + R$ 
6:      $K = V_{p,t}C^T S^{-1}$ 
7:      $\mu_{u,t} = \mu_{p,t} + K(o_t - C^T \mu_{p,t})$ 
8:      $V_{u,t} = V_{p,t} - KCV_{p,t}$ 
9:   end for
10:  return loglik
11: end procedure

```

Figure 2: Probabilistic DMP: executing and tracking a reference signal

Taking the expectation of the complete-data log likelihood with respect to the posterior $p(\mathbf{S} | P, \theta^{old})$ defines the function

$$Q(\theta, \theta^{old}) = \mathbb{E}_{\mathbf{S} | \theta^{old}} [\ln p(P, \mathbf{S} | \theta)],$$

which we maximize with respect to parameters θ . The updates for \mathbf{R} can now be derived by setting the derivative of this function to 0, and then solve for \mathbf{R} which can be done analytically. This completes the learning of a probabilistic dynamic movement primitive, where the parameters required to fully describe the probabilistic representation are $\theta_{\text{primitive}} = \{\mu_w, \Sigma_w, \alpha, \beta, \mathbf{R}\}$.

Executing a probabilistic DMP

Dynamic Movement Primitives are typically used to generate desired trajectories that a controller is expected to track. Note, the hidden state of the probabilistic DMP is given by $s_t = (\ddot{p}_t \ \dot{p}_t \ p_t)$ Pure feedforward trajectory generation is achieved by initializing the linear dynamical system with the task parameters (goal, start and duration of the motion) and then simply unrolling the probabilistic model (see Algorithm 1) - this will generate exactly the same desired trajectory as a standard DMP, including uncertainty estimates.

As discussed above, we can also consider noisy sensor feedback when executing a probabilistic DMP. Instead of simply forward predicting the hidden state, inference is performed to estimate the hidden state distribution. For linear dynamical systems this inference process is widely known as Kalman filtering. In order to do so, we need to formulate how the hidden state s_t generates the chosen sensor measurements o_t . This is done by defining the observation function $h(s_t)$ that creates the reference signal we want to track by transforming the hidden state. Assuming, we observe the actual position of the system, this leads to following inference steps: At time step t , the system feedforward predicts the new desired (hidden) state $\mu_{p,t}$

$$\mu_{p,t} = \mathbf{A}\mu_{t-1} + \mathbf{B}u_{t-1} \quad (16)$$

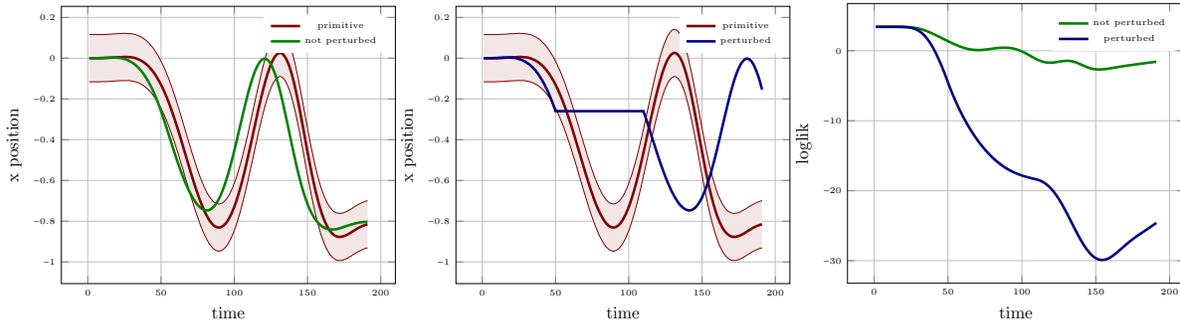


Figure 3: Failure detection illustration: In the (left) and (middle) plot the mean trajectory of a learned primitive with the variance at each time step. Additionally: (left) the green trajectory illustrates a non-perturbed observed trajectory that slightly differs from the mean. (middle) The same trajectory but perturbed. (right) The log-likelihood of both observed trajectories calculated online while trajectories were unrolled.

then the Kalman innovation update, adds a term proportional to how much the reference signal deviates from the observed sensor feedback

$$\mu_t = \mu_{p,t} + K(o_t - h(s_t)). \quad (17)$$

Thus at time step $t + 1$ the desired behavior of the DMP is online modulated

$$\mu_{p,t+1} = \mathbf{A}(\mu_{p,t} + K(o_t - h(s_t))) + \mathbf{B}u_t \quad (18)$$

$$= \mathbf{A}\mu_{p,t} + \mathbf{B}u_t + \mathbf{A}K(o_t - h(s_t)) \quad (19)$$

Note, the similarity of this update to the online modulation performed in (Pastor et al. 2011). To summarize, the execution of a probabilistic DMP with noisy measurement observations is performed via Kalman filtering (see Algorithm 2) and automatically leads to online adaptation of desired behaviors to account for disturbances.

Summary

In summary, the main characteristics of our probabilistic DMP representation are:

- It is a probabilistic model that keep accelerations part of the hidden state - this allows to execute probabilistic DMPs just as regular DMPs - where a rollout of the hidden states produces desired accelerations, velocities and positions.
- The non-linear forcing term is modeled probabilistically with phase-dependent variability - creating a phase-dependent transition covariance in the linear dynamical system view.
- A reference signal can be tracked as part of the probabilistic model - which creates a principled way of modulating the desired behavior online, when the sensor feedback deviates from the reference signal.

Failure Detection with Probabilistic DMPs

Besides the insight presented above, other benefits of this probabilistic formulation exist. For instance, assuming we have learned a probabilistic representation for a motion primitive $\{\mu_w, \Sigma_w, \alpha, \beta, \mathbf{R}\}$ we can perform online failure detection: While executing the motion primitive using Algorithm 2, we can utilize the probabilistic formulation to continuously monitor how likely it is that this motion primitive is generating the measured actual state of the system.

We illustrate this application in Figure 3. This illustration shows a 1D primitive being executed, first perturbation free (right), and then we artificially hold the movement such that the primitive is not continued to executed but the probabilistic model expects it to (middle). On the left hand side we see how the likelihood values evolve during movement execution. Note, how once we artificially hold the movement the likelihood value significantly drops. Thus, we can use this likelihood measure to detect execution failures.

For initial quantitative evaluation purposes we recorded a dataset of 2D trajectories of letters with a digitizing tablet. All letters that are easily written with one stroke have been recorded, a total of 22. Each of this letter is meant to represent a movement primitive. To learn a probabilistic representation per primitive we collected 10 training demonstrations and an additional 10 demonstrations for testing purposes.

Once a probabilistic representation of a motion primitive has been learned, we measure the minimum likelihood of each training demonstration given the learned model parameters, and store that value with the parameters. Throughout the execution of a motion primitive the loglikelihood value might increase or decrease depending on how much variation from the mean we observe. The challenge is thus to detect natural variation from a perturbation and/or failure. Here we simply classify an execution as failed, if at any point during execution the loglikelihood values drops below 2 times the minimum reported value for that primitive. On our data this works very well, such that of all $22 * 10 = 220$ test cases only 2 test cases were classified as failed when not perturbed. When we perturb the simulated execution of each test case by artificially blocking the execution (as illustrated in Figure 3), then all 220 test cases are classified as failed.

Conclusions

We have presented a probabilistic model for dynamic movement primitives. Inference in this graphical model is equivalent to Kalman filtering, and when performing inference a feedback term is automatically added to the DMP trajectory generation process. Besides this insight, we have shown the potential of probabilistic DMPs on the application of failure detection. Future work will explore and evaluate the potential impact of these probabilistic representation in more detail.

References

- Bishop, C. 2006. Pattern recognition and machine learning. Calinon, S.; Li, Z.; Alizadeh, T.; Tsagarakis, N. G.; and Caldwell, D. G. 2012. Statistical dynamical systems for skills acquisition in humanoids. In *International Conference on Humanoid Robots*.
- Dragan, A. D.; Muelling, K.; Bagnell, J. a.; and Srinivasa, S. S. 2015. Movement Primitives via Optimization. In *International Conference on Robotics and Automation*.
- Gams, A.; Nemeč, B.; Ijspeert, A. J.; and Ude, A. 2014. Coupling movement primitives: Interaction with the environment and bimanual tasks. *IEEE Transactions on Robotics*.
- Ijspeert, A. J.; Ijspeert, A. J.; Nakanishi, J.; Nakanishi, J.; Hoffmann, H.; Hoffmann, H.; Pastor, P.; Pastor, P.; Schaal, S.; and Schaal, S. 2013. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Computation*.
- Khansari-Zadeh, S. M., and Billard, A. 2010. Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2676–2683. IEEE.
- Khansari-Zadeh, S. M., and Billard, A. 2011. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics* 27(5):943–957.
- Meier, F.; Theodorou, E.; Stulp, F.; and Schaal, S. 2011. Movement segmentation using a primitive library. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2011)*.
- Meier, F.; Theodorou, E.; and Schaal, S. 2012. Movement segmentation and recognition for imitation learning. In *International Conference on Artificial Intelligence and Statistics*, 761–769.
- Paraschos, A.; Daniel, C.; Peters, J. R.; and Neumann, G. 2013. Probabilistic movement primitives. In *Advances in neural information processing systems*, 2616–2624.
- Pastor, P.; Hoffmann, H.; Asfour, T.; and Schaal, S. 2009. Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 763–768. IEEE.
- Pastor, P.; Schaal, S.; Pastor, P.; Righetti, L.; Kalakrishnan, M.; Righetti, L.; Kalakrishnan, M.; and Schaal, S. 2011. Online movement adaptation based on previous sensor experiences. In *International Conference on Intelligent Robots and Systems*.
- Pastor, P.; Kalakrishnan, M.; Meier, F.; Stulp, F.; Buchli, J.; Theodorou, E.; and Schaal, S. 2012. From dynamic movement primitives to associative skill memories. *Robotics and Autonomous Systems*.
- Rückert, E. A.; Neumann, G.; Toussaint, M.; and Maass, W. 2013. Learned graphical models for probabilistic planning provide a new class of movement primitives. *Frontiers in computational neuroscience* 6:97.
- Schaal, S. 1999. Is imitation learning the route to humanoid robots? 233–242.
- Tenorth, M.; Bandouch, J.; and Beetz, M. 2009. The tum kitchen data set of everyday manipulation activities for motion tracking and action recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 1089–1096. IEEE.
- Toussaint, M. 2009. Probabilistic inference as a model of planned behavior. *Künstliche Intelligenz* 3(9):23–29.
- Wilson, A. D., and Bobick, A. F. 1999. Parametric hidden Markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.