Dynamic Imitation in a Humanoid Robot through Nonparametric Probabilistic Inference

David B. Grimes **Rawichote Chalodhorn** **Rajesh P. N. Rao**

Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195
http://neural.cs.washington.edu
{grimes,choppy,rao}@cs.washington.edu

***Abstract*— We tackle the problem of learning imitative whole-body motions in a humanoid robot using probabilistic inference in Bayesian networks. Our inference-based approach affords a straightforward method to exploit rich yet uncertain prior information obtained from human motion capture data. Dynamic imitation implies that the robot must interact with its environment and account for forces such as gravity and inertia during imitation. Rather than explicitly modeling these forces and the body of the humanoid as in traditional approaches, we show that stable imitative motion can be achieved by learning a sensor-based representation of dynamic balance. Bayesian networks provide a sound theoretical framework for combining prior kinematic information (from observing a human demonstrator) with prior dynamic information (based on previous experience) to model and subsequently infer motions which, with high probability, will be dynamically stable. By posing the problem as one of inference in a Bayesian network, we show that methods developed for approximate inference can be leveraged to efficiently perform inference of actions. Additionally, by using nonparametric inference and a nonparametric (Gaussian process) forward model, our approach does not make any strong assumptions about the physical environment or the mass and inertial properties of the humanoid robot. We propose an iterative, probabilistically constrained algorithm for exploring the space of motor commands and show that the algorithm can quickly discover dynamically stable actions for whole-body imitation of human motion. Experimental results based on simulation and subsequent execution by a HOAP-2 humanoid robot demonstrate that our algorithm is able to imitate a human performing actions such as squatting and a one-legged balance.**

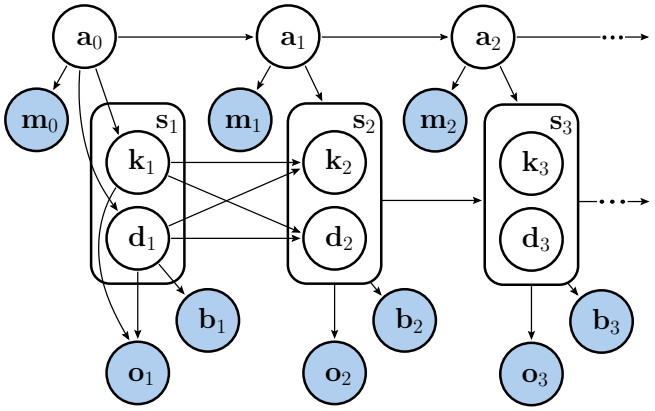

Fig. 1. **Graphical model for dynamically constrained imitation.** A dynamic Bayesian network (DBN) used to infer imitative motions depicts a set of variables with arrows representing conditional dependencies between variables. Variables which are observed (evidence variables) are shaded blue. Latent action variables $\mathbf{a}_t$ are modeled as generating both human kinematic postures $\mathbf{m}_t$ and the robot kinematic configuration $\mathbf{k}_t$. The modeled dynamic configuration of the robot $\mathbf{d}_t$, augments the kinematic information to form the full state of the robot $\mathbf{s}_t$. All conditional dependencies are shown between the first and second time slices. Subsequent times slices are shown with the arrows based on the state variable $\mathbf{s}_t$, revealing the simple first order Markovian structure of the DBN.

I. Introduction

Imitation learning presents a promising approach to the problem of enabling complex behavior learning in humanoid robots. Learning through imitation provides the robot with strong prior information by observing a skilled instructor (often assumed to be a human demonstrator). This paper presents a model for exploiting this prior information about whole-body motions gathered from observing a human performance of the motion. Although the observation of the teacher is informative, there is a high degree of uncertainty in how the robot can and should imitate. Our model accounts for some of these sources of uncertainty including: noisy and missing kinematic estimates of the teacher, mapping ambiguities between the human and robot kinematic spaces, and lastly, the large uncertainty due to the effect of physical forces imparted on the robot during imitation.

Bayesian networks provide a sound theoretical approach to incorporating prior, yet uncertain information. Thus we pose the problem of finding dynamically balanced imitative motions as one of learning and inference in a Bayesian network. This also allows us to utilize additional prior information crucial for achieving stability: a probabilistic sensor-based model of dynamic balance. Despite compelling advances in solving complex continuous partially observable Markov decision problems (POMDPs) [1], [2] we pose the problem as one of inference also for a pragmatic reason: to leverage and evaluate recent approximate inference approaches to efficiently solve problems that have previously been regarded as intractable.

II. Related Work

Efficiently generating dynamically balanced biped and humanoid motion has long been considered a difficult and important research problem. Our overall approach is similar in spirit to Yamane and Nakamura's *dynamics filter* [3]. However, unlike their approach which requires a physics-based model of the robot, our approach is *model-free* in the sense of not requiring any knowledge of dynamic properties such as mass or moment of inertia. Other approaches based on the zero-moment point (ZMP) [4], [5] or inverted pendulum models [6] also require accurate knowledge of physical parameters to achieve stable motion. On the other hand, sensor-based or adaptive approaches are typically aimed at stabilization within a particular gait model [7], [8] and do not easily generalize to other classes of whole-body motion. Finally, none of these models specify a probabilistic method for the incorporation of uncertain prior information from human kinematic estimates.

Inverse reinforcement learning [9] and apprenticeship learning [10] have been proposed to learn controllers for complex systems based on observing an expert and learning their reward function. However, the role of this type of expert and that of our human demonstrator must be distinguished. In the former case the teacher is directly controlling the artificial system. In the imitation learning paradigm, one can only observe the teacher controlling their own body. Further, despite kinematic similarities between the human and humanoid robot, the dynamic properties of the robot and human are very different and must be accounted for in the learning process.

There exists a large body of other work on imitation learning using a variety of approaches, ranging from using nonlinear dynamical systems for imitation [11] to imitating arm motions using biologically motivated methods [12]. We refer the reader to these and related literature [13]–[16] for more details and alternate approaches to the imitation problem.

III. Probabilistic Dynamic Balance Model

Our approach is based on the dynamic Bayesian network (DBN) shown in Figure 1. Imitative motions are modeled as a generative process: a single sequence of actions $\mathbf{a}_t$ generates both the human demonstrator's kinematic postures ($\mathbf{m}_t$) as well as the humanoid robot's kinematic postures ($\mathbf{k}_t$). We assume that a length T sequence of human kinematic estimates has been observed. In our experiments, we use a commercially available retroreflective marker-based optical motion capture system to obtain estimates of human joint angles through inverse kinematics (IK). The IK skeletal model of the human was restricted to have the same degrees of freedom as the Fujitsu HOAP-2 humanoid robot. This affords a trivial mapping (adjusting only for zero position and sign) between the two kinematic spaces.

Representing humanoid motion in the full kinematic configuration space is problematic due to the large number of degrees of freedom and the well known curse of dimensionality. Fortunately, with respect to a wide class of motions (such as walking, kicking, bowing), the full number of degrees of freedom (25 in the HOAP-2) is highly redundant. Dimensionality

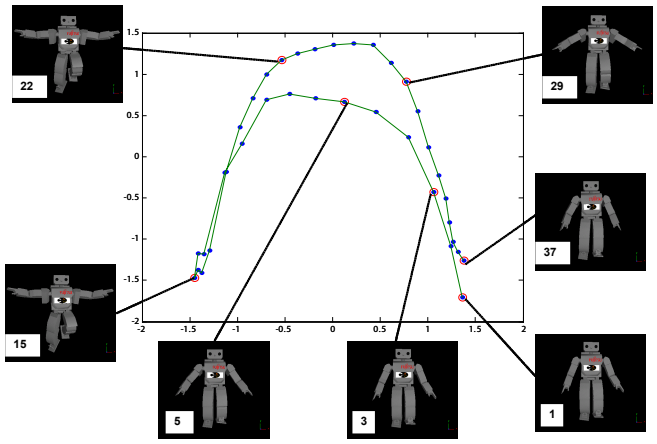

Fig. 2. **Latent posture space representation.** Using principal components analysis, a high degree of freedom motion (here, a one-legged balance) is embedded in a two-dimensional space. The blue line shows the sequence of postures represented in the low-dimensional space. For selected points along the trajectory, an image of the robot posture is shown using a purely kinematic simulation. For efficiency, we perform inference in the low-dimensional latent space to find a dynamically stable sequence of actions which imitate an observed behavior.

reduction techniques can be profitably used to represent high-dimensional data in compact low-dimensional latent spaces. For simplicity, we use linear principal components analysis (PCA) but other non-linear embedding techniques (such as the GPLVM [17]) may be worth exploring for representing wider classes of motion using fewer dimensions [18].

Using the estimated kinematic motion from several demonstrations of the motion or behavior, we form a matrix C from the d principal component vectors of the posture space. The matrix C represents a linear embedding of the original high-dimensional points in a d-dimensional space. An illustrative example showing a one legged balancing motion embedded in a two-dimensional space is shown in Figure 2.

We can thus describe the human's posture $\mathbf{m}_t$ as a linear function of the latent action $\mathbf{a}_t$ with assumed additive Gaussian noise ($v_\mathbf{m}$):

$$\mathbf{m}_t = C\mathbf{a}_t + v_\mathbf{m} \quad , \qquad v_\mathbf{m} \sim \mathcal{N}\left(\mu_m, \Sigma_m + \Sigma_w\right). \qquad (1)$$

The parameters of the Gaussian noise process denoted μ_m, Σ_m characterize the inherent noise in joint estimates from the motion capture system. In practice these can be estimated using maximum likelihood on calibration data obtained using a calibration rig. The Σ_w parameter determines how much the human's motion is likely to deviate from the shared latent action representation. This non-isotropic diagonal covariance allows for differentially weighting the deviation of joints in the human and the humanoid. For example it may be more important to reproduce arm movements more exactly than the ankle joint motions.

The second major component of our model can be likened to a dynamics constraint. Rather than placing constraints on moments or center of mass, which require complex and

precisely tuned physical models, we leverage sensors which measure quantities closely related to dynamic stability. In this work, we utilize observations from a torso gyroscope ($\mathbf{g}_t$) and pressure sensors on the feet($\mathbf{f}_t$). Our framework easily generalizes to include other sensors and sources of information such as motion estimates based on visual information and/or proximity sensors. We represent the dynamics of the robot at time t using the following sensor-derived components:

$$\mathbf{d}_t = [g_t^x \; g_t^y \; p_t^1 \cdot \cdot \; p_t^k]' \;\; , \qquad p_t^i = \phi_i(\mathbf{f_t}) \tag{2}$$

where g_t^x and g_t^y represent the angular velocity of the humanoid torso with respect to the x and y axis respectively (rotation about the z (upward) axis is omitted as we are only concerned here with the force of gravity), and the variables p_t^i represent foot pressure differences extracted by a set of features ϕ_i from the eight pressure points (along four sides of each foot) of the vector $\mathbf{f}_t$. In our experiments, we used four linear features: 1) front - rear (both feet), 2) sum of left foot readings - sum of right foot readings, 3) left foot: right - left side, 4) right foot: right - left side. More robust, non-linear features of the foot pressure points could also be easily incorporated. The HOAP-2 robot's sensors provide measurements of the angular rotation $\mathbf{g}_t$ and foot pressure $\mathbf{f}_t$ every 1 millisecond. Because of inherent sensor noise, we utilize a Gaussian model $P(\mathbf{g}_t, \mathbf{f}_t|\mathbf{d}_t)$ and consider $\mathbf{d}_t$ partially observable.

We introduce the variable b_t to indicate if the robot will be dynamically balanced conditioned on the current dynamics configuration $\mathbf{d}_t$. In practice, because we always want to enforce this condition for all t, we observe that $b_t = 1$. In this sense, the variable b_t is simply a notational device and is akin to a dummy child of $\mathbf{d}_t$ in belief propagation [19] used for indicating evidence. Note that although we want to highly constrain $\mathbf{d}_t$, we also want to maintain a belief state over the dynamics configuration due to uncertainty in the forward model and the sensor observation model. For this reason, we do not want to simply "observe" $\mathbf{d}_t$ as a target stable dynamics configuration.

We constrain the dynamics configuration using highly peaked (small variance) central Gaussians about each of the components in $\mathbf{d}_t$:

$$P(b_t|\mathbf{d}_t) \propto b_t \, \mathcal{N}\left(g_t^x; 0, \sigma_x^2\right) \mathcal{N}\left(g_t^y; 0, \sigma_y^2\right) \prod_i^k \mathcal{N}\left(p_t^i; 0, \sigma_{p_i}^2\right). \tag{3}$$

where σ_x^2, σ_y^2, $\sigma_{p_i}^2$ are the variances of the x and y axis angular velocities, and the i-th foot pressure features.

We represent the kinematic state $\mathbf{k}_t$ using the same d-dimensional latent space as the action $\mathbf{a}_t$. Thus, the kinematic observation model (the $\mathbf{k}_t$ to $\mathbf{o}_t$ link in the Bayesian network), which gives the probability of the observation (joint encoder position) θ_t given a particular kinematic posture $\mathbf{k}_t$, is given by:

$$\theta_t = C\mathbf{k}_t + v_k \;\; , \qquad v_k \sim \mathcal{N}\left(0, \sigma_k^2\right) \tag{4}$$

The model also includes a prior over the initial action $P(\mathbf{a}_1)$, and temporal action "prior" $P(\mathbf{a}_{t+1}|\mathbf{a})$. The temporal action prior term is useful for indicating our preference that actions (for both the human and robot) be smoothly varying. We model this using a Gaussian relationship:

$$\mathbf{a}_{t+1} = \mathbf{a}_t + v_a \;\; , \qquad v_a \sim \mathcal{N}\left(0, \sigma_a^2\right). \tag{5}$$

Before discussing the temporal forward model, we introduce the notation $\mathbf{s}_t = [\mathbf{k}_t; \mathbf{d}_t]'$ which allows us to write the forward dynamics compactly as the conditional probability model $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$.

IV. Nonparametric Forward Model Learning

We now turn to the problem of forward prediction of the dynamics state component $\mathbf{d}_{t+1}$ given the previous state (consisting of both kinematics and dynamics components) $\mathbf{s}_t$ and an action command $\mathbf{a}_t$. Given our definitions of kinematic and dynamic state, we cannot assume a linear forward relationship nor do we have a physics-based model to linearize about. Thus our approach is based on learning the forward model $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ directly from empirical data collected from trials on the robot. Gaussian processes have been shown to be very powerful in learning stochastic non-linear relationships directly from empirical data [20]. As no finite set of parameters can describe a Gaussian process, this method is called *nonparametric*.

Empirical data gathered from exploration trials form a set of tuples $\mathcal{D} \subset \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ constructed via samples of the mapping $([\hat{\mathbf{s}}_t; \hat{\mathbf{a}}_t] \rightarrow \hat{\mathbf{s}}_{t+1})$. The inferred state $\hat{\mathbf{s}}_t$ is found using maximum likelihood estimation from the single observation $\mathbf{o}_t$ (alternatively, one could use the expectation maximization (EM) algorithm to incorporate estimation into model learning but single time-slice maximum likelihood state estimation was found to be sufficient here). Note that the tuples in $\mathcal{D}$ are time invariant; thus, the subscript t is dropped. From $\mathcal{D}$, we construct the process input data matrix by concatenating state and action vectors into D_{in} (also called the design matrix). The output data matrix D_{out} contains the subsequent state estimates.

Although a linear model can fairly accurately model $P(\mathbf{k}_t|\mathbf{s}_t, \mathbf{a}_t)$ for much of the state and action spaces, we choose to model this nonparametrically for two reasons: (1) to obtain a unified method for learning $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$, and (2) around certain state configurations with large inertial or gravitation forces, the forward kinematics may be non-linear. We want our model to be able to capture this potentially non-linear relationship.

Gaussian process regression computes a predictive posterior of the i-th dimension of the output, $\mathbf{s}_{t+1}^{(i)}$ as normally distributed given a multivariate input vector $[\mathbf{s}_t; \mathbf{a}_t]$.

$$P(\mathbf{s}_{t+1}^{(i)}|\mathbf{s}_t, \mathbf{a}_t) = \mathcal{N}\left(\mu_*^{(i)}, \Lambda_*^{(i)}; \mathbf{s}_{t+1}^{(i)}\right) \tag{6}$$

The mean and variance of the predictive distribution are functions of the covariance matrix $K(D_{\text{in}}, D_{\text{in}})$ which is $n \times n$ where n denotes the number of values in the data set. The covariance matrix is defined by the covariance (or kernel) function $k(\cdot, \cdot)$. Different choices exist for the kernel function,

but for simplicity here we use the common squared exponential kernel. The squared exponential kernel is also referred to as the radial basis function (RBF) kernel:

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x}_p - \mathbf{x}_q)' \Sigma_l^{-1} (\mathbf{x}_p - \mathbf{x}_q)\right) + \sigma_n^2 \delta_{pq} \tag{7}$$

The squared exponential kernel, due to its modeling flexibility is considered an appropriate choice when one doesn't have domain specific knowledge about how the output of the process (co)varies as a function of its inputs. The exact form of the kernel is selected by setting the kernel hyperparameters $\Sigma_l^{-1}, \sigma_f^2, \sigma_n^2$ representing the length scales of each of the inputs (which determine how sensitive an output dimension is to changes in each input dimension), the noise free variance, and the additive process noise variance respectively. Hyperparameters were optimized using a standard scaled conjugate gradient approach, maximizing the log likelihood of the data $\mathcal{D}$ over these hyperparameters.

By evaluating the kernel function for each pair of data, we can compute the kernel matrix $K(D_{\text{in}}, D_{\text{in}})$ and its inverse necessary for the predicted posterior mean:

$$\mu_*^{(i)} = K([\mathbf{s}_t; \mathbf{a}_t], D_{\text{in}}) \left[K(D_{\text{in}}, D_{\text{in}}) + \sigma_n^2 I\right]^{-1} D_{\text{out}}^{(i)} \tag{8}$$

and variance:

$$\Lambda_*^{(i)} = K([\mathbf{s}_t; \mathbf{a}_t], [\mathbf{s}_t; \mathbf{a}_t]) - K([\mathbf{s}_t; \mathbf{a}_t], D_{\text{in}}) \left[K(D_{\text{in}}, D_{\text{in}}) + \sigma_n^2 I\right]^{-1} K(D_{\text{in}}, [\mathbf{s}_t; \mathbf{a}_t]) \tag{9}$$

.

Although the kernel and its inverse can be precomputed, the $O(n^3)$ time necessary for matrix inversion can become too costly in our case. Exploration of the world gains additional data after each trial which must be then incorporated into the model. A simple approach that appears effective given our experience is to simply represent the kernel matrix using a random subset of the data (when the number of data points grows over about 300). Recently, several approaches have tackled the problem of large kernel matrices, either by applying heuristics to select a subset of the data points [21] or by low-rank approximations of the kernel matrix [22].

V. Nonparametric Action Inference

We now present an algorithm for action selection based on belief propagation [19] within the graphical model shown in Figure 1. The result of performing belief propagation is a set of marginal beliefs $B(x) = P(x|\,\mathcal{E})$ where $\mathcal{E}$ is the set of all observed variables. Belief propagation was originally restricted to tree structured graphical models with discrete variables. Recent advances in machine learning have broadened the applicability to general graph structures [23] and to continuous variables in undirected graph structures [24], [25].

The inference approach we adopt is most similar to the NBP [25] method. While NBP is formulated for inference in a Markov random field (MRF) model, our approach uses Pearl's notation for belief propagation in directed Bayesian networks. We note that this difference is only semantic, and adopted out of convenience as any Bayesian network can be represented as a MRF, or more generally a factor graph [26]. Belief propagation formulated for a Bayesian network is more convenient in our setting given the natural conditional semantics of the forward and observation models.

Belief propagation (BP) computes marginals by passing messages along the edges of the graphical model. Messages are in the form of distributions over single variables. On BP iteration n, parent i of variable $\mathbf{x}$ would pass to $\mathbf{x}$ the distribution $\pi_{\mathbf{x}}^n(\mathbf{u}_i)$. Likewise, child j of variable $\mathbf{x}$ would pass to $\mathbf{x}$ the distribution $\lambda_{\mathbf{y}_j}^n(\mathbf{x})$. In a discrete (finite) space, messages are easily represented by multinomial distributions. For arbitrary continuous densities, accurately and efficiently representing messages is in itself a challenge. Seeking generality and the ability to handle complex multi-modal distributions, we use a nonparametric approach based on a collection of weighted kernel functions. Specifically, (as in the NBP approach [25]) we used Gaussian kernels whose parameters can be efficiently estimated. Although one might then view this message distribution as being parameterized, the result when many kernels functions are used is akin to a sample based representation (as in particle filters or the condensation algorithm [27]).

Belief propagation computes a belief distribution $B^n(\mathbf{x})$ based on the product of two sets of messages $\pi^n(\mathbf{x})$ and $\lambda^n(\mathbf{x})$, which represent the information coming from neighboring parent and children variable nodes respectively:

$$P(\mathbf{x}|\mathcal{E}) = B^n(\mathbf{x}) = \lambda^n(\mathbf{x})\pi^n(\mathbf{x}) \tag{10}$$

.

Although the product of two mixtures of Gaussians is also a mixture of Gaussians, the complexity of computing the product grows exponentially when performed repeatedly. Thus, we approximate products of messages based on the technique of multiscale sampling and multiplication of pairs of mixture components [28].

Following [19], we treat observed and hidden variables in the graph identically by allowing a node $\mathbf{x}$ to send itself the message $\lambda^\star(\mathbf{x})$. If the node is observed, we model this message as a Dirac delta distribution about the observed data point, and a uniform distribution otherwise. This "self message" is considered in the product of all messages from the m children (denoted $\mathbf{y}_j$) of X:

$$\lambda^n(\mathbf{x}) = \lambda^\star(\mathbf{x}) \prod_j^m \lambda_{\mathbf{y}_j}^n(\mathbf{x}). \tag{11}$$

Messages from parent variables are incorporated by integrating the conditional probability of $\mathbf{x}$ over all possible values of the k parents multiplied by the probability of that combination of values as evaluated in the corresponding messages from a parent node:

$$\pi^n(\mathbf{x}) = \int_{\mathbf{u}_1} \cdot\cdot \int_{\mathbf{u}_k} P(\mathbf{x}|\mathbf{u}_1, \cdot\cdot, \mathbf{u}_k) \prod_i^k \pi_{\mathbf{x}}^n(\mathbf{u}_i) d\mathbf{u}_{1\cdot\cdot k}. \tag{12}$$

Messages are updated according to the following two equations:

$$\lambda_{\mathbf{x}}^{n'}(\mathbf{u}_j) = \int_{\mathbf{x}} \lambda^n(\mathbf{x}) \int_{\mathbf{u}_1} \cdot\cdot \int_{\mathbf{u}_{j-1}} \int_{\mathbf{u}_{j+1}} \cdot\cdot \int_{\mathbf{u}_k} \cdot \\ P(\mathbf{x}|\mathbf{u}_1, \cdot, \mathbf{u}_k) \prod_{i \neq j} \pi_{\mathbf{x}}^n(\mathbf{u}_i) d\mathbf{u}_{1:k/j} \quad (13)$$

$$\pi_{\mathbf{y}_j}^{n'}(\mathbf{x}) = \pi^n(\mathbf{x}) \lambda_{\mathbf{x}}^n(\mathbf{x}) \prod_{i \neq j} \lambda_{\mathbf{y}_i}^n(\mathbf{x}) \quad (14)$$

Although the output of the Gaussian process is a normal distribution for a particular input value, the integrals in Equations 12 and 13 are not analytically solvable in closed form when conditionals such as $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ are Gaussian processes. Thus, in the case of messages involving the forward model, we apply a sample based (Monte-Carlo) technique to estimate the outgoing messages or beliefs. For example, in Equation 12, we draw samples from each parent distribution $\pi_{\mathbf{x}}^n(\mathbf{u}_i)$, and based on means and variances obtained from the Gaussian process model, we use an efficient kernel bandwidth estimation technique discussed in [25] to form the message $\pi^n(\mathbf{x})$. While in theory this approach requires a number of particles that is exponential in the dimensionality, our results show that using around 1000-1500 samples in a four dimensional latent space is sufficient. As future work, we intend to experiment with moment-matching methods which have been shown to be able to approximate the Gaussian process when the inputs are drawn from a normal distribution [29]. This has the potential to further reduce inference time.

The computation of $\lambda_{\mathbf{x}}^{n'}(\mathbf{u}_j)$ as shown in Equation 13 is approached similarly. However, in this case, we have to integrate over an output variable of the Gaussian process as well. Rather than "invert" the Gaussian process we simply learn backward Gaussian processes for mapping backward in time, and to the action variable $\mathbf{a}_t$ given states $\mathbf{s}_{t+1}$ and $\mathbf{s}_t$.

VI. Dynamic Imitation Results

In this section, we present results from our method for learning dynamically balanced imitative motions based on observing a human perform whole-body motions. Motion capture data was collected while performing various actions, each with three to five repetitions. Kinematic joint angles were then estimated using inverse kinematics. From this kinematic data, we construct the latent joint configuration space from the principal components basis matrix C. The number of principal components was found empirically and was guided by striking a balance of several factors. Firstly, the dimensionality chosen should afford accurate reconstruction of the prior kinematic motion. In our motion data, greater than 99% of the variance of the data was along the first four principal components. A second factor in selecting the dimensionality is to allow sufficient representational power for finding a stable motion within the latent space of actions. Finally, for reasons of efficiency, it is desirable to keep the number of latent dimensions to a minimum. We experimented with dimensionalities between three and six, and found four to be a good balance of representational freedom and efficiency.

In order to bootstrap the nonparametric forward model, we first perform a set of random exploration trials which are sampled from the initial set of beliefs $B(\mathbf{a}_t) \propto P(\mathbf{a}_t|\mathbf{m}_1 \dots \mathbf{m}_T)$. Parametric model variances (such as in the observation model, temporal action smoothness model, and human input kinematic model) were also set empirically to make sure that the relative values allowed for a compromise between kinematically similar imitations and dynamic stability of the resulting motion.

We tested an implementation of our method using the robotics simulator package Webots, which provides accurate dynamics simulation of the Fujitsu HOAP-2 robot. We used its sensor simulation capability to also model the necessary gyroscope and foot pressure sensor signals (to which we added realistic levels of Gaussian noise to help avoid overfitting the physics of the simulator).

After learning an initial forward model based on the data from the bootstrap trials, we add the dynamic balance variables (all set to 1) to the evidence set and again compute the beliefs $B(\mathbf{a}_t) \propto P(\mathbf{a}_t|\mathbf{m}_1 \dots \mathbf{m}_T, b_1 \dots b_T)$. Based on these beliefs, we compute and execute the maximum probability actions $\hat{\mathbf{a}}_t = argmax_{\mathbf{a}_t} B(\mathbf{a}_t)$. From this execution, we add the actions as well as the maximum likelihood state estimates to the data set $\mathcal{D}$ and update the kernel matrix. For efficiency, we found that updating the kernel hyperparameters in every trial is unnecessary. We re-optimized every five trials, with no discernible degradation in performance.

Note that currently we incorporate feedback after each trial. In theory one could utilize feedback immediately by performing a model update and inference step at each time step within a trial; however, for simplicity of implementation, we perform open-loop execution of selected actions.

We found that our model is able to quickly infer sequences of actions which do not cause the robot to lose balance and fall, even if all of the bootstrap iterations were unstable. Specifically we present results here based on dynamic imitation of two motions: a squat motion and a one-legged balance motion (in other experiments, we were able to also generate stable motions on examples such as bowing and leaning side-to-side). The resulting imitations of these two selected motions are shown in Figure 3.

Figure 4 shows how the duration that the robot remained balanced and did not fall quickly increases to the full motion length of 63 time steps, after 20 bootstrap and approximately 15 constrained trials. Figure 5 illustrates that the likelihood of the dynamics sensors increases dramatically once the probabilistic dynamic balance constraint propagated throughout the network.

Finally, in order to make sure that our inferred imitative motions (taken after the balance duration and stability log likelihoods converged) were not merely overfitting the physical simulation of Webots, we applied the final motion as open-loop commands to the actual HOAP-2 humanoid robot. We found that even in open-loop mode with no calibration to the real robot, our imitative motions were dynamically stable.

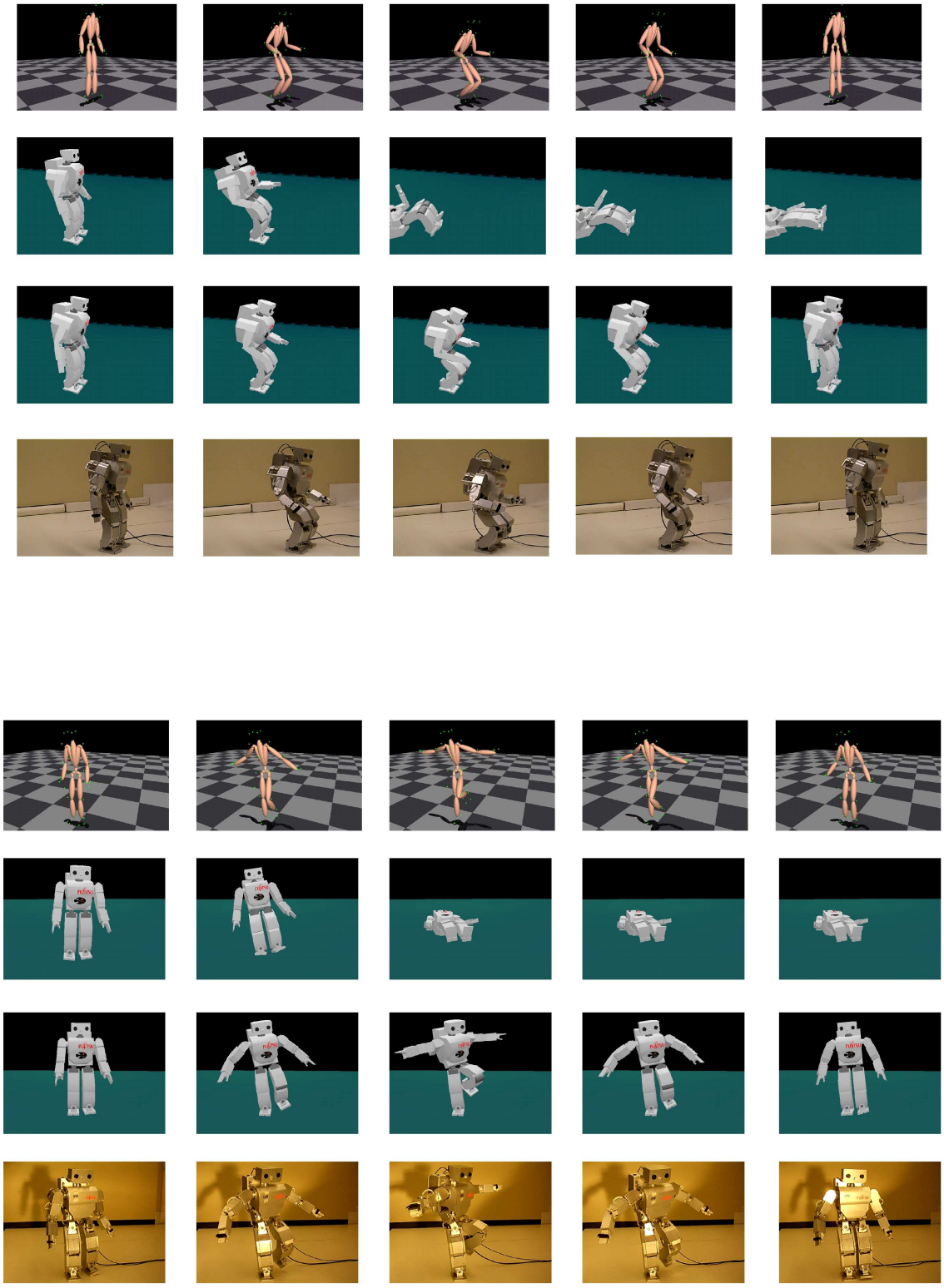

Fig. 3. **Dynamic imitation results**. Two sets of thumbnails demonstrate the imitation of two human motions: a squatting motion (first four rows) and a one-legged balance motion (next four rows). Within each set, the first row shows the motion of the human demonstrator (via a skeletal model fit to the marker data). The second row shows the result of a purely kinematic imitation performed in the simulator, which in both examples is dynamically unstable: the robot falls almost immediately. The third row shows the result of executing the final inferred actions found by our algorithm in the simulator. The final row consists of frames from a video of our HOAP-2 humanoid robot performing imitation using the same actions.

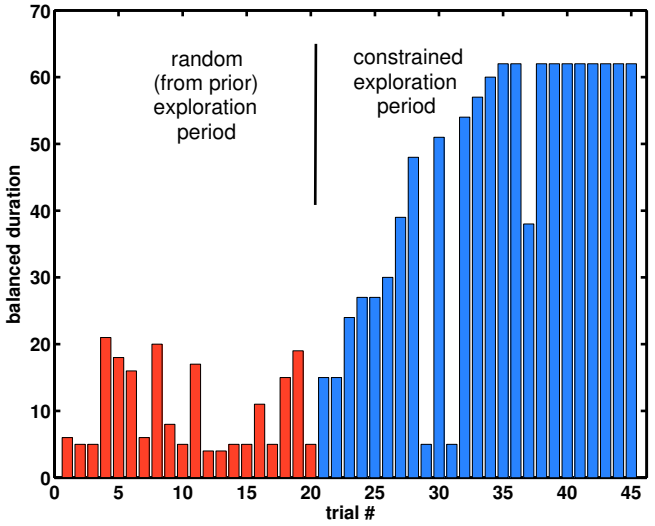

Fig. 4. **Dynamic balance duration over imitation trials.** The duration (number of time steps) that the executed imitation was balanced is shown as a function of the trial number. In this example, the robot was learning to imitate the one-legged balance motion discussed in the text and shown in Figure 3. The bootstrap (random exploration) trials are shown in red. The trials where actions inferred included contributions from the dynamic constraint distribution are shown in blue. Note that the full motion length is $T = 63$, which is achieved by our algorithm around the 15th inferred action sequence (after the 20 initial random exploration trials).

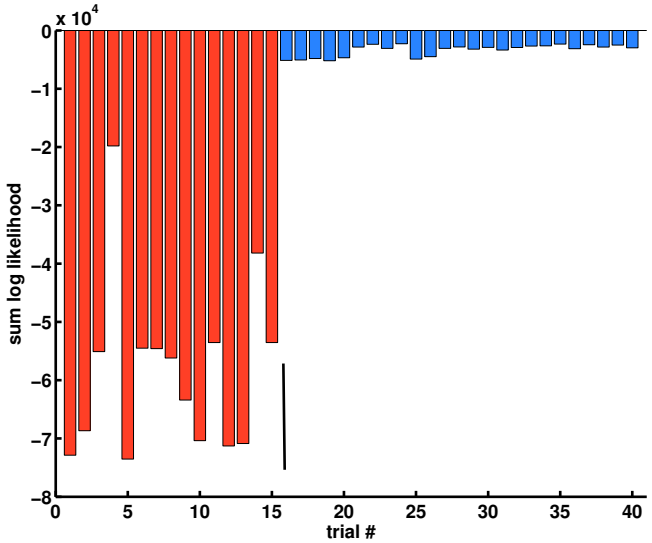

Fig. 5. **Log likelihood of dynamics configuration.** The sum over all time steps of the log likelihood of the probabilistic dynamics model $P(b_t|\mathbf{d}_t)$ is shown as a function of the trial number. Note that this likelihood increases dramatically once we learn a valid forward model and constrain the dynamics using the probabilistic dynamic balance model. In this example, the robot was learning to imitate the squatting motion shown in Figure 3.

Specifically, out of twenty trials, the robot never lost its balance during imitation of the squatting motion. In the case of the one legged balance, the robot was balanced throughout the motion in 16 out of the 20 trials (80%).

VII. Conclusion

We have proposed a new technique for learning dynamically stable whole-body motions in a humanoid robot from human demonstration. Our model is based on Gaussian processes and nonparametric Bayesian inference, and incorporates prior information about both desired kinematics as well as dynamics in a rigorous probabilistic framework. Empirical results using a HOAP-2 humanoid robot demonstrate that the proposed approach can be effective in learning stable whole-body motions from human motion capture data without requiring complicated physics-based dynamic models. Future work will focus on integrating closed-loop control into the probabilistic framework and expanding the repertoire of the robot's imitative abilities to more complex behaviors including locomotory actions.

Acknowledgments

This research was supported by NSF, the ONR Adaptive Neural Systems program, the Sloan Foundation, and the Packard Foundation.

References

[1] A. Y. Ng and M. I. Jordan, "PEGASUS: A policy search method for large MDPs and POMDPs," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 406–415.

[2] D. Bagnell and J. Schneider, "Autonomous helicopter control using reinforcement learning policy search methods," in *Proceedings of the International Conference on Robotics and Automation (ICRA 2001)*, 2001.

[3] K. Yamane and Y. Nakamura, "Dynamics filter - concept and implementation of on-line motion generator for human figures," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 421–432, 2003.

[4] M. Vukobratovic and B. Borovac, "Zero moment point-thirty-five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157–173, 2004.

[5] S. Kajita and K. Tani, "Adaptive gait control of a biped robot based on realtime sensing of the ground profile," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1996, pp. 570–577.

[6] J. Yamaguchi, N. Kinoshita, A. Takanishi, and I. Kato, "Development of a dynamic biped walking system for humanoid: development of a biped walking robot adapting to the humans' living floor," in *IEEE International Conference on Robotics and Automation (ICRA)*, 1996, pp. 232–239.

[7] J. Hu, J. Pratt, and G. Pratt, "Adaptive dynamic control of a biped walking robot with radial basis function neural networks," in *Internation Conference on Robotics and Automation (ICRA)*, 1998, pp. 400–405.

[8] M. Ogino, Y. Katoh, M. Aono, M. Asada, and K. Hosoda, "Reinforcement learning of humanoid rhythmic walking parameters based on visual information," *Advanced Robotics*, vol. 18, no. 7, pp. 677–697, 2004.

[9] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Proc. 17th International Conf. on Machine Learning*, 2000, pp. 663–670.

[10] P. Abbeel and A. Y. Ng, "Exploration and apprenticeship learning in reinforcement learning," in *Proc. 21st International Conference on Machine Learning*, 2005.

[11] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Trajectory formation for imitation with nonlinear dynamical systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 752–757.

[12] A. Billard and M. Mataric, "Learning human arm movements by imitation: Evaluation of a biologically-inspired connectionist architecture," *Robotics and Autonomous Systems*, no. 941, 2001.

[13] M. Y. Kuniyoshi and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Transaction on Robotics and Automation*, vol. 10, no. 6, pp. 799–822, Dec 1994.

[14] J. Demiris and G. Hayes, "A robot controller using learning by imitation," in *Proceedings of the 2nd International Symposium on Intelligent Robotic Systems*, 1994.

[15] C. G. Atkeson and S. Schaal, "Robot learning from demonsration," in *Proc. 14th International Conf. on Machine Learning*, 2000, pp. 12–20.

[16] B. Price and C. Boutilier, "Accelerating reinforcement learning through implicit imitation," *Journal of Artificial Intelligence Research*, vol. 19, pp. 569–629, 2003.

[17] N. D. Lawrence, "Gaussian process latent variable models for visualization of high dimensional data," in *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[18] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic, "Style-based inverse kinematics," in *ACM Transactions on Graphics (SIGGRAPH 2004)*.

[19] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[20] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models." in *Advances in Neural Information Processing Systems 18 [Proceedings of NIPS 2005]*. The MIT Press, 2006, pp. 1441–1448.

[21] N. Lawrence, M. Seeger, and R. Herbrich, "Fast sparse gaussian process methods: The informative vector machine," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 2003, pp. 625–632.

[22] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems (NIPS)*. Cambridge, MA: MIT Press, 2006, pp. 1259–1266.

[23] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural Computation*, vol. 12, no. 1, pp. 1–41, 2000.

[24] M. Isard, "Pampas: real-valued graphical models for computer vision," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2003.

[25] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation." in *CVPR (1)*, 2003, pp. 605–612.

[26] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm." *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[27] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[28] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky, "Efficient Multiscale Sampling from Products of Gaussian Mixtures," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2004.

[29] A. Girard, C. E. Rasmussen, J. Quionero-Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs - application to multiple-step ahead time series forecasting," in *Advances in Neural Information Processing Systems (NIPS)*, 2003.