The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Incremental Motion Primitive Learning by Physical Coaching Using Impedance Control

Dongheui Lee and Christian Ott

Abstract— **We present an approach for kinesthetic teaching of motion primitives for a humanoid robot. The proposed teaching method allows for iterative execution and motion refinement using a forgetting factor. During the iterative motion refinement, a confidence value specifies an area of allowed refinement around the nominal trajectory. A novel method for continuous generation of motions from a hidden Markov model (HMM) representation of motion primitives is proposed, which incorporates relative time information for each state. On the realtime control level, the kinesthetic teaching is handled by a customized impedance controller, which combines tracking performance with soft physical interaction and allows to implement soft boundaries for the motion refinement. The proposed methods were implemented and tested using DLR's humanoid upper-body robot Justin.**

I. INTRODUCTION

Programming of motions for a humanoid robot is challenging due to contact state dependent dynamic constraints and the large number of degrees of freedom (DOF) involved [1]–[3]. Several frameworks use the concept of motion primitives [4]–[7], since this allows a compact description of general motion patterns. Such motion primitives for humanoid robots can be acquired by mapping motions from a human demonstrator, or by an offline optimization process. In the case of human demonstration, kinematic differences between the demonstrator and the robot must be handled, and several frameworks for this motion *retargetting* problem have been proposed [4], [8]–[11]. Generation of motion primitives by mathematical optimization, on the other hand, can result in motions, which are efficient but may look unnatural from a human supervisor's subjective point of view depending on the choice of optimization criteria. Regardless of the method for creation of motion primitives, it thus may be desired that a human supervisor can modify the shape and timing of learned motion primitives in an intuitive way.

Physical human robot interaction in imitation learning can be roughly classified into interaction during execution and interaction during learning. Online adaptation of motion primitives during execution was treated in several works [4], [12]. In particular the adaption to a desired endpoint [4], [13] or an intermediate point [12] of the motion was treated in detail. In [12], physical interaction during motion generation was considered, while motion and interaction primitives were trained by an observational learning strategy.

D. Lee is with the Department of Electrical Engineering and Information Technology, Technical University of Munich, D-80290 Munich, Germany dhlee@tum.de

Ch. Ott is with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR e.V.), P.O. Box 1116, 82234 Wessling, Germany christian.ott@dlr.de

Physical interaction during learning, often called kinesthetic teaching, was treated in [14]–[16]. In [15], the physical coaching was realized by deactivating the controlled motion of individual selected joints. In [14], [16], very low servo gains were used for guiding the robot manually during teaching. However, these approaches of kinesthetic teaching often lead to unsynchronized motions because the teacher moves motors one by one rather than demonstrating natural coordinated movements.

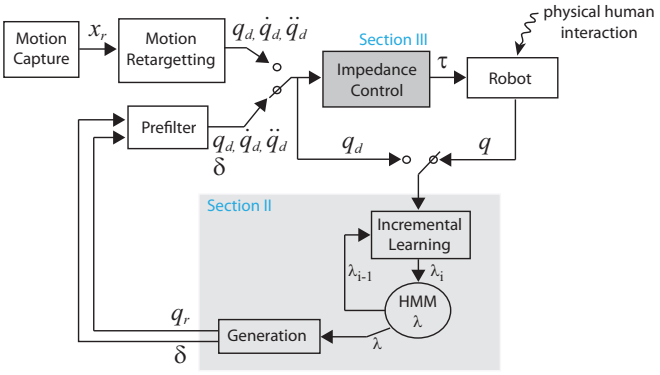

Fig. 1. System overview: Acquired motion primitives are iteratively refined by physical interaction with a human teacher.

In most previous works on imitation learning, (off-line) one-shot learning, where all training data is prepared before training, has been often considered. Recently, approaches for incremental learning have been proposed in [17], [18]. In these works, the knowledge of motion primitives is updated as more demonstrations are provided, without keeping all the training data in a database. In [18], they achieved incremental motion learning with a performance similar as batch learning. However, they still require additional information (e.g., posterior probabilities or the number of all previous training data) in memory, besides current model parameters and new incoming data. Moreover, a limitation with these methods is that they can become insensitive to new data when the training data set is large.

In order to achieve an intuitive teaching of natural motions, we propose the following concepts.

1) Combination of observational learning and kinesthetic eaching: In order to ensure synchronization (coordination) of complex whole body motions on a humanoid robot, our imitation learning procedure starts with observation learning (i.e. whole body motion retargetting from a human demonstrator to a robot) prior to kinesthetic demonstrations.

2) Incremental learning with a forgetting factor: Since retargetted movements might not fit exactly to the desired

movements due to kinematic differences and mapping errors, a refining process is required. In such a case, slowly forgetting old data is more efficient than having the same weighting factors for all training data in history, which was done in [17], [18].

3) A motion refinement tube concept as a guideline for kinesthetic correction: The tube represents an area of allowed deviation from the nominal trajectory of the motion primitive. The refinement tube helps the human teacher to correct only the desired part of the motion without accidentially disturbing other joints.

4) A customized impedance control, which allows to realize the refinement tube: The proposed impedance controller allows, in contrast to [15], [16], to combine tracking of motion primitives in free-space with a kinesthetic modification by a human supervisor. As a consequence, multiple motion elements (e.g., hands, torso motions) can be corrected simultaneously during the physical coaching.

5) Improving generation of HMM-based motion primitives: Existing HMM-based motion primitives cannot capture continuity of trajectories due to the discrete nature of states. In order to overcome this limitation, we extend the HMM-based motion primitive representation by learning the correlation between temporal and spatial data within a discrete state. By using this continuous HMM output, the feedforward terms for the impedance controller can be computed by a simple fast prefilter and there is no need for a slow filter for smoothening the trajectory.

A schematic overview of the proposed kinesthetic teaching approach is shown in Fig. 1. A motion retargetting algorithm converts motion capture data into joint angle trajectories for the robot. Herein, the marker control algorithm from [11] is adopted which allows to get synchronized natural whole body motion. These trajectories are implemented via a joint level impedance controller (Section III). In parallel, an HMM based motion primitive learning algorithm is active. During incremental physical teaching, the joint angle trajectory and the parameters for the refinement tube δ are generated from the HMM (Section II). An implementation and evaluation of the proposed concepts on the humanoid upper body robot Justin [19] is presented in Section IV.

II. INCREMENTAL MOTION LEARNING

A. Motion Primitive

HMM-based representation of motion primitives has been often used for generalization of movements demonstrated to a robot, because HMMs allow to represent spatiotemporal variabilities in a stochastic way [5], [20]. HMMs have their advantages in motion recognition due to their sequence-based nature. The temporal relation between states is encoded in terms of initial state probability and state transition probability. Time series of different speed can be handled without modification which allows to perform recognition of known motion primitives in an incoming data stream during runtime. A limitation of this approach is discontinuity of the generated trajectory due to the discrete states in the HMM. Therefore, a filtering technique or a Monte Carlo sampling and averaging technique are usually needed to smoothen the trajectory [5], [7], [12].

As an alternative method to achieve a continuous trajectory, a GMM (Gaussian mixture models) and GMR (Gaussian mixture regression) based approach was proposed in [6]. While conventional GMM cannot capture temporal information, because learning is performed based on data points rather than sequences, in [6] correlation between temporal and spatial data is encoded in Gaussian probability by adding timestamps explicitly in the training data points. Once GMM is trained, a smooth trajectory can be retrieved by GMR, therein timestamps should be given as an explicit input variable. A drawback of this approach is that data has to be segmented and scaled in time a priori. Thereafter, to the best of our knowledge, online recognition of an incoming data stream with a different speed than the training data is not possible.

Motion primitive representation in this work follows basic HMM properties, which allows good online recognition performance. In order to overcome the discrete nature of states in the HMM (which results in generation of stepwise sequences), correlation between temporal and spatial data is learned explicitly. Herein, a normalized time variable is introduced in each state. This variable is 0 when entering a state, 1 when leaving the state, and linearly interpolated during the stay at the state. This relative temporal representation allows to handle data sequences of different speed, even whose speed is time-varying within a motion primitive. In contrast to [6], a preprocessing of observation (e.g., scaling in time) for learning and recognition is not needed.

The training algorithm is as follows. First, from a motion sequence ${}^s\mathcal{O} = \{{}^s\boldsymbol{o}(t)\}$ (At this stage, training data consists of a time series of spatial data)[1], HMM parameters are trained via an EM algorithm [21]. Then, the corresponding state sequence for the spatial data sequence ${}^s\mathcal{O}$ is calculated via the Viterbi algorithm [21]. From this state sequence, the relative temporal sequence ${}^t\mathcal{O} = \{{}^to(t)\}$ within each state is calculated. Therefore, the mean of temporal data of each state is always the same, which is 0.5. Then, from ${}^t\mathcal{O}$ and ${}^s\mathcal{O}$, the covariance ${}^{ts}\boldsymbol{\Sigma}$ for each state is calculated.

B. Motion Generation - Trajectory and Tube

The generation procedure of the new proposed motion primitive (Sec. II-A) is as follows. Motion patterns are decoded using the expectation operator in the stochastic model. The motion generation is a stochastic process, consisting of state transition generation and motion output generation from the state transition.

Step 1) A general state sequence $\mathcal{Q} = \{q(t)\}$ is generated deterministically. An initial state $q(1)$ is chosen according to the initial state probability distribution π_i, i.e.

$$q(1) = \arg\max_i \pi_i \quad .$$

[1]The spatial data ${}^s\boldsymbol{o}(t)$ can be in joint coordinates and/or Cartesian coordinates. The system overview in the case of joint space is shown in Fig. 1. Figure 4 illustrates the case of task space.

Let a_{ij} denote the probability to transit from state i to state j. The expected duration to stay at the state i is calculated as $1/(1-a_{ii})$. After the duration, the next state is calculated according to the state transition probability distribution, i.e.

$$q(t+1) = \arg\max_j a_{q(t)j} \quad .$$

Step 2) From the state sequence $\mathcal{Q}$, the relative temporal sequence ${}^t\mathcal{O}$ within each state is calculated. As mentioned in Sec. II-A, the value of the relative temporal data ${}^to(t)$ is ranging from 0 to 1.

Step 3) Given the state sequence $\mathcal{Q}$ and relative temporal sequence ${}^t\mathcal{O}$, a sequence of spatial data is generated using Gaussian regression [22]. For each time step t, a mixture of Gaussians in the corresponding state $q(t)$ is considered. For simplicity, we use i instead of $q(t)$ in the following equations (1)-(2). Given ${}^to(t)$, the conditional expectation ${}^s\boldsymbol{o}(t)$ is

$${}^s\boldsymbol{o}(t) = \sum_{k=1}^{K} \left\{ {}^s\boldsymbol{\mu}_{ik} + \frac{{}^{ts}\boldsymbol{\Sigma}_{ik}}{{}^{tt}\Sigma_{ik}} ({}^to(t) - {}^t\mu_{ik}) \right\} \qquad (1)$$

where ${}^s\boldsymbol{\mu}_{ik}$ is the mean vector of spatial data part in the k-th mixture component at state i. The symbol ${}^t\mu_{ik}$ is the mean vector of temporal data part in the k-th mixture component at state i. The k-th Gaussian at state i is given by

$$\boldsymbol{\Sigma}_{ik} = \begin{bmatrix} {}^{tt}\Sigma_{ik} & {}^{ts}\boldsymbol{\Sigma}_{ik} \\ {}^{ts}\boldsymbol{\Sigma}_{ik} & {}^{ss}\boldsymbol{\Sigma}_{ik} \end{bmatrix} \ ,$$

where ${}^{tt}\Sigma_{ik}$ and ${}^{ts}\boldsymbol{\Sigma}_{ik}$ are the variance of temporal data and the covariance between temporal and spatial data, respectively. The conditional variance of ${}^s\boldsymbol{o}(t)$ is estimated as

$${}^{s|t}\boldsymbol{\Sigma} = \sum_{k=1}^{K} \left\{ \mathrm{diag}({}^{ss}\boldsymbol{\Sigma}_{ik}) - \frac{{}^{ts}\boldsymbol{\Sigma}_{ik}}{{}^{tt}\Sigma_{ik}} \right\} \qquad (2)$$

where ${}^{ss}\boldsymbol{\Sigma}_{ik}$ is the covariance matrix of spatial data of the k-th Gaussian at state i.

During kinesthetic teaching, the conditional variance ${}^{s|t}\boldsymbol{\Sigma}$ is used for designing the motion refinement tube. The tube is centered around ${}^s\boldsymbol{o}(t)$ and has a radius of $\boldsymbol{\delta} = 3^{s|t}\boldsymbol{\Sigma} + \epsilon$, where ϵ is the minimum allowance. This radius $\boldsymbol{\delta}$ is used as an input parameter of the impedance controller presented in Sec. III.

C. Incremental Learning with a forgetting factor

The incremental learning method herein is a variation of the EM algorithm for multiple observations [23]. It is also similar to the generative incremental learning method in [18] in the sense that the training data consists of new incoming observations and generated motion patterns from a motion primitive. Compared to the existing methods, our incremental learning method uses a forgetting factor which leads to exponential forgetting of previous data. This allows to correct the models efficiently and to avoid insensitivity to new incoming data for a large training set. No additional information (e.g., posterior probabilities or the number of all previous training data) rather than the current model parameters and new incoming data is required.

In our incremental learning algorithm, two motion sequences[2] (${}^s\mathcal{O}^e$, $e = 1 \cdots E$, $E = 2$) are given as training data. One motion sequence is the new incoming training data. The other one is a generated motion sequence from the current motion primitive (Section II-B). The weighting factor for each sequence (${}^s\mathcal{O}^e$) is given as w^e. For the new incoming motion sequence, w^e becomes the forgetting factor $w^e = \eta$. For the generated motion sequence, the weighting factor becomes $1-\eta$, so that $\sum_{e=1}^{E} w^e = 1$. In the M-step, new parameters $\bar{\lambda} = \{\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{a}}, \bar{\boldsymbol{c}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\}$ for the HMM are estimated by using the old HMM parameters $\lambda = \{\boldsymbol{\pi}, \boldsymbol{a}, \boldsymbol{c}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and the two training data. The initial state probability π_i is updated as the expected relative frequency spent in state i at time 1, i.e.

$$\bar{\pi}_i = \sum_{e=1}^{E} w^e \gamma_i^e(1) \ ,$$

where the variable $\gamma_i^e(t)$ denotes the probability of being at state i at time t for the observation sequence ${}^s\mathcal{O}^e$. The state transition probability a_{ij} is updated to the expected number of transitions from state i to state j relative to the expected total number of transitions from state i, i.e.

$$\bar{a}_{ij} = \frac{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e-1} \xi_{ij}^e(t)}{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e-1} \gamma_i^e(t)} \ ,$$

where T_e is the time duration of ${}^s\mathcal{O}^e$ and the variable $\xi_{ij}^e(t)$ is the probability of being in state i at time t and being in state j at time $t+1$ for the sequence ${}^s\mathcal{O}^e$. The update rules for the mixture of Gaussian distributions, which consist of the weight c_{ik}, mean vector ${}^s\boldsymbol{\mu}_{ik}$, and covariance matrix ${}^{ss}\boldsymbol{\Sigma}_{ik}$ for the k-th mixture component at state i, are as follows.

$$\begin{aligned} \bar{c}_{ik} &= \frac{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e} \gamma_{ik}^e(t)}{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e} \gamma_i^e(t)} \\ {}^s\bar{\boldsymbol{\mu}}_{ik} &= \frac{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e} \gamma_{ik}^e(t) {}^s\boldsymbol{o}^e(t)}{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e} \gamma_{ik}^e(t)} \end{aligned}$$

$${}^{ss}\bar{\boldsymbol{\Sigma}}_{ik} = \frac{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e} \gamma_{ik}^e(t) ({}^s\boldsymbol{o}^e(t) - {}^s\boldsymbol{\mu}_{ik})({}^s\boldsymbol{o}^e(t) - {}^s\boldsymbol{\mu}_{ik})^T}{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e} \gamma_{ik}^e(t)}$$

where the variable $\gamma_{ik}^e(t)$ denotes the probability of being at state i at time t with the k-th mixture component accounting for the observation ${}^s\boldsymbol{o}^e(t)$.

III. INTERACTION CONTROL FOR PHYSICAL MOTION REFINEMENT

Considering the physical motion refinement strategy discussed in the previous sections, we can identify the following demands on the real-time controller.

1) Trajectory tracking when there is no physical interaction.

2) Allow a compliant behavior with low stiffness if the robot is distracted from the nominal trajectory by a human teacher.

[2]Note that herein ${}^s\mathcal{O}^e$ is the e-th sequence of spatial data.

3) Allow to limit the range of allowed deviation from the nominal trajectory.

Two alternative strategies follow immediately from the above requirements: One possible solution would be to use two different controllers, a trajectory tracking controller and an interaction controller, and switch between these two controllers depending on the observed contact force. Alternatively, we can integrate the above requirements in an appropriate impedance behavior and use a customized impedance controller. This has the advantage that no switching between two controllers and no explicit detection/observation of the interaction force is required. The second alternative is thus followed in this paper.

As an output of the motion retargetting algorithm we assume a joint angle trajectory $\boldsymbol{q}_d(t) \in \mathbb{R}^n$, for the n joints of a robot (see Fig. 1). Moreover, we assume that this trajectory is twice differentiable, and that the first and second time derivative are available. This is true for instance for the marker control algorithm described in [11]. If an inverse kinematics based retargetting algorithm like, e.g., the one from [10], is used instead, the availability of the desired acceleration can always be ensured by the use of an appropriate pre-filter.

We assume a rigid-body model of a robot with n degrees-of-freedom [24]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}} \tag{3}$$

with $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ as the symmetric and positive definite inertia matrix, $\boldsymbol{g}(\boldsymbol{q})$ as the gravity torques, and $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ as the matrix corresponding to the centrifugal and Coriolis forces, which fulfills the passivity property $\dot{\boldsymbol{M}} = \boldsymbol{C} + \boldsymbol{C}^T$ [24]. The control input is given by the vector of joint torques $\boldsymbol{\tau} \in \mathbb{R}^n$, and $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^n$ is a vector of external torques, which represents the interaction of the robot with its environment including the human teacher.

In impedance control the control goal is given by a dynamic relation between the robot motion and the external forces $\boldsymbol{\tau}_{\text{ext}}$ [25]. As usually done in robotics, we choose this dynamic relation in form of a dynamical system of second order in which we can identify generalized inertia, damping, and stiffness terms. It is well known that feedback of the external torques can be avoided in impedance control if the desired inertia is chosen equal to the natural inertia of the robot [26]. Bearing requirement 1) to 3) in mind, we therefore focus on the design of nonlinear stiffness and damping terms and choose a desired impedance behavior similar to the closed loop behavior of a PD+ like tracking controller [27]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\tilde{\boldsymbol{q}}} + (\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{D}(\boldsymbol{q}))\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{s}(\tilde{\boldsymbol{q}}) = \boldsymbol{\tau}_{\text{ext}} \ , \tag{4}$$

where $\tilde{\boldsymbol{q}} = \boldsymbol{q} - \boldsymbol{q}_d$ denotes the deviation of $\boldsymbol{q}$ from the virtual[3] equilibrium trajectory $\boldsymbol{q}_d$, and $\boldsymbol{D}(\boldsymbol{q})$ and $\boldsymbol{s}(\tilde{\boldsymbol{q}})$ are a nonlinear positive semi-definite damping matrix and a nonlinear stiffness term, respectively. The inclusion of $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ in (4) makes allowance for the position-dependence of the inertia term. If $\boldsymbol{s}(\tilde{\boldsymbol{q}})$ is chosen by a conservative force field, the dynamics (4) represents a passive mapping from the velocity error $\dot{\tilde{\boldsymbol{q}}}$ to the external torques $\boldsymbol{\tau}_{ext}$ ensuring the stability of the system in free motion and in feedback interconnection with a passive environment.

By comparing the desired impedance (4) with the original robot dynamics (3), we obtain the impedance control law

$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_d + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_d - \boldsymbol{D}(\boldsymbol{q})\dot{\tilde{\boldsymbol{q}}} - \boldsymbol{s}(\tilde{\boldsymbol{q}}) \ , \tag{5}$$

for which we still have to select appropriate stiffness and damping terms which account for requirement 1)-3).

The stiffness term is chosen for each joint such that it represents a high *local stiffness*[4] $k_{0,i}$ close to the virtual equilibrium (requirement 1, for accurate tracking), but has a low local stiffness for a larger deviation within the refinement tube $|\tilde{\boldsymbol{q}}| < \delta_i$, where $\delta_i \geq 0$ denotes one half of the diameter of the refinement tube for the i^{th} joint (requirement 2). This can be achieved by limiting the stiffness force to a rather small value $\tau_{\text{max,i}}$, by which the robot is pushed back to the equilibrium trajectory during physical teaching. For even larger deviations, the user should feel a soft constraint, which we implement by an increase in the local stiffness from 0 (within the refinement tube) to $k_{t,i}$ (requirement 3). An appropriate stiffness term for each joint i is given by

$$s_i(\tilde{q}_i) = \tau_{\max,i}\tanh\left(\frac{k_{0,i}}{\tau_{\max,i}}\tilde{q}_i\right) + s_{t,i}(\tilde{q}_i) \ , \tag{6}$$

$$s_{t,i}(\tilde{q}_i) = \begin{cases} 0 & |\tilde{q}_i| < \delta_i \\ \text{sign}(\tilde{q}_i)\ln(\cosh(k_{t,i}(\delta_i - |\tilde{q}_i|))) & |\tilde{q}_i| \geq \delta_i \end{cases} .$$

The shape of this stiffness function is shown in Fig. 2. The first term implements the high local stiffness which is saturated in the area of physical teaching, while the second term represents the soft constraint for implementing the refinement tube. The value δ_i is commanded based on the covariance information of the motion primitive representation, as shown in Sec. II-B. It can easily be verified that this stiffness term represents a conservative force field, for which the local stiffness matrix $\boldsymbol{K} := \partial\boldsymbol{s}(\tilde{\boldsymbol{q}})/\partial\tilde{\boldsymbol{q}}$ is a non-negative diagonal matrix. For the design of the damping matrix $\boldsymbol{D}(\boldsymbol{q})$, we utilize the *double diagonalization* method reported in [26] based on the symmetric matrices $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{K}$.

IV. EXPERIMENTS

A. Experimental Setup

The evaluation of the proposed incremental learning method is done using the humanoid upper-body robot Justin [19]. In the experiments, we used the 19 joints of the arms (2 times 7 DOF), torso (3 DOF), and head (2DOF), but did not use the mobile base and the fingers.

As a motion capture system, we use the wearable motion capture suit from the company XSens [28]. This system provides the user with position and orientation data for 23 segments on the human body. The data is sent online

[3]In impedance control, the desired trajectory in free motion usually is called a *virtual* equilibrium trajectory.

[4]With the term *local stiffness*, we denote the differential $\partial s_i(\tilde{q})/\partial\tilde{q}$

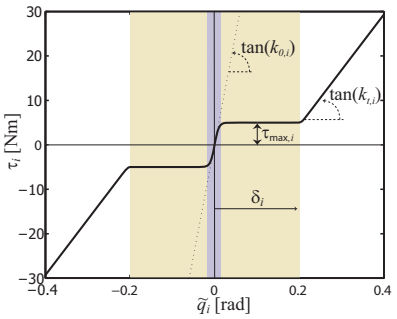

Fig. 2. Joint level stiffness term, exemplified for the case $\tau_{\mathrm{max,i}} = 10[\mathrm{Nm}]$, $k_{0,i} = 500[\mathrm{Nm/rad}]$, and $k_{t,i} = 0.25k_{0,i}$. The blue area shows the domain of high stiffness. The yellow area shows the coaching area in which a human user feels zero local stiffness, but a small constant force pushing towards the virtual equilibrium trajectory. The white area shows the border of the refinement tube implemented via a *soft constraint*.

to the realtime controller of Justin using a UDP packet streaming protocol. Out of the position and orientation data of all the segments we selected the following representative components of the upper body: pelvis *pose* (position and orientation), chest pose, head orientation, elbow position (right and left), and hand pose (right and left). The orientation data is represented by a unit quaternion description. The whole data set thus consists of 38 elements. Since the upper body of the demonstrator and the robot generally have similar anthropomorphic shape, but different kinematics and size, the motion retargetting problem arises (see Fig. 1). For the position data, we used a simple physiological re-scaling, which takes account of the different body size of the human and the robot. For the human motion retargetting, we use the marker control algorithm from [11]. Taking account of the segment data from the used motion capture suit, we implemented the algorithm using 6D virtual springs rather than only 3D translational springs as in [11]. Notice that the material presented in this paper does not substantially depend on the motion retargetting algorithm, and one could likewise use an inverse kinematics algorithm like the one presented in [10]. Figure 3 shows a human user wearing the motion capture suit, the corresponding set of segment positions, and the configuration of Justin computed from the marker control algorithm.

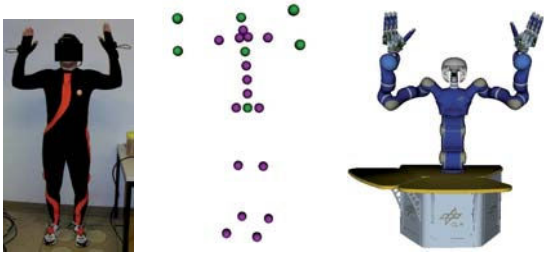

Fig. 3. Human user wearing the XSens motion capture suit [28], the set of segment positions, and the corresponding configuration of Justin. The green segment poses are used in the marker control algorithm.

B. Iterative Kinesthetic Refinement

For the kinesthetic teaching, the robot uses the impedance behavior from section III. The robot Justin is equipped with joint torque sensors in all the joints except for the neck and the mobile base. This allows to implement the impedance control law (5) in an outer control loop in combination with an inner torque control loop. This procedure can be justified from a control theoretical point of view by singular perturbation analysis [29] of a robot model in which the joint elasticity is considered [30], [31]. Since the neck joints are not equipped with joint torque sensors, we implement the impedance controller for the torso and the arms, while we use a standard position controller for the neck.

TABLE I

IMPEDANCE PARAMETERS FOR THE KINESTHETIC TEACHING

joint	$k_{0,i}[\mathrm{Nm/rad}]$	$k_{t,i}[\mathrm{Nm/rad}]$	$\tau_{\mathrm{max},i}[\mathrm{Nm}]$
torso R1	800	1600	20
torso P1	500	1000	20
torso P2	500	1000	20
shoulder 1	500	500	15
shoulder 2	500	500	12
upper arm	400	400	10
elbow	400	400	8
fore arm	200	200	3
wrist 1	200	200	2.5
wrist 2	200	200	2

Table I shows the used parameters of the stiffness term from the impedance control law (5). We utilize higher stiffness values for joints which have a higher load in order to ensure good tracking behavior. The maximum force values were tuned manually.

In the first experiment, we are evaluating the physical refinement of learned motion primitives without using the motion refinement tube, i.e. setting δ_i to very high values for all the joints. The motion primitives are represented directly by the 38 selected elements from the motion capture segment data (position and orientation data), as mentioned in Sec. IV-A. The used system structure is shown in Fig. 4.

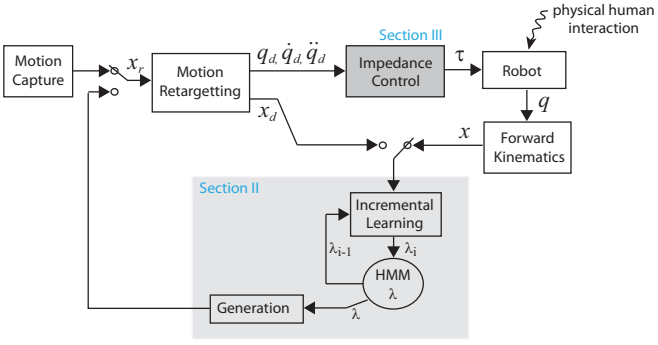

Fig. 4. System overview for the Cartesian control.

Figure 5 shows a qualitative presentation of the incremental learning including kinesthetic refinement process. In this experiment, five manually segmented demonstrations (one observational and four kinesthetic demonstrations) of a dancing motion were provided by a human. In the first demonstration, the human's movement was measured by the motion capture suit (Fig. 3) and retargetted to the robot by the marker control algorithm [11]. The retargetted motion

is shown in the first row in Fig. 5. Originally the human performed the dance, moving his right hand horizontally in front of his face. However, in the retargetted robot's dance, the robot's hand is above its head, as one can see in the first row of Fig. 5. In this sequence, the human teacher judged that the right hand motion is too high and pulls it down kinesthetically during the robot's execution (second row in Fig. 5). The human refined the robot's motion four times in total, by pulling the right hand down, rotating the torso, and positioning the left hand away from its mobile base. The third row shows the generalized motion primitive after four refinement steps with a forgetting factor of $\eta = 0.5$. Here, the robot's hand is moving approximately in the correct height. For comparison, the fourth row shows the corresponding result from batch learning, where all five training data are prepared in advance for the training process. Since the two proposed incremental methods in [18] have similar training results as batch learning according to [18], we compared only with the batch learning. One can see that the hand is still higher than during the kinesthetic refinement.

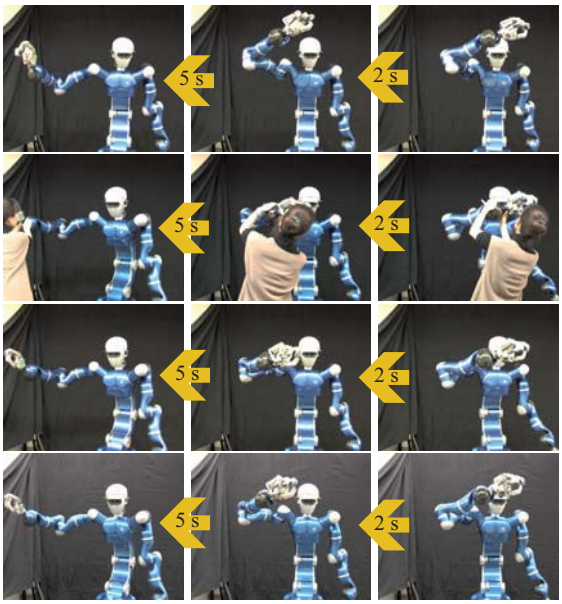

Fig. 5. Kinesthetic motion refinement: The upper row (from right to left) shows the snapshots from the original motion primitive. In the second and third row, the snapshots during and after the proposed learning are shown. The fourth row shows the corresponding result from batch learning.

Figure 6 shows the trained motion primitive at each step of the incremental learning. The trajectories of the right hand height are depicted because the hand height was the main focus of refinement. The incoming training data (black dashed lines) is encoded into HMM parameters. Herein, in an HMM, ten states and one Gaussian model for each state are used. The number of states is chosen from experimental experiences [7], [12]: HMMs with $10 \sim 20$ states work well with whole body motion patterns ($19 \sim 30$ DOF). The generalized motion primitive (red solid line) is generated from the HMM parameters by the algorithm in Sec. II-B. The new HMM parameters are updated using the old HMM parameters and the new incoming observation. For comparison, the

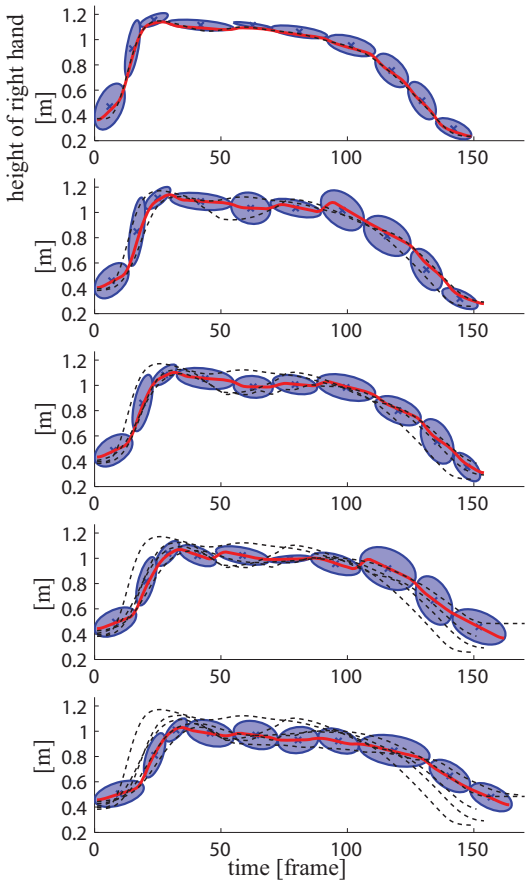

Fig. 6. Results of the proposed incremental learning and generation. Five sequences of training data were used to teach a dance motion. The right hand height trajectories are shown. The first row is the training result from the first training data. The fifth row shows the last result after five training steps. The black dashed curves are the training data provided so far. Incrementally incoming data is encoded into an HMM with 10 states. The red solid curve is the generalized trajectory of the current motion primitive by the proposed generation algorithm in Sec. II-B.

corresponding result of conventional methods (conventional HMM representation, batch learning [21], and deterministic generation [32]) is shown in Fig. 7. While the batch learning contains uncertainty of all the training data, our proposed incremental learning method refines the motion by forgetting old data exponentially. This results in smaller uncertainties after learning than batch learning. In Fig. 7, the Gaussians are aligned with the horizontal axis because the conventional HMM representation cannot encapture correlation between

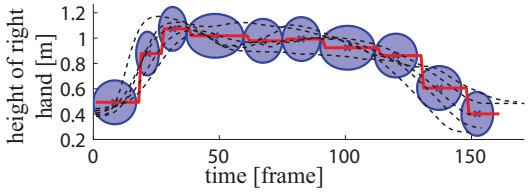

Fig. 7. Results of batch learning and deterministic generation. Five training data (black dashed) were embodied into an HMM with 10 states by one-shot offline learning. The right hand height trajectories are shown. From the HMM, the generalized trajectory (red solid line) is reproduced by the deterministic generation technique from [32].

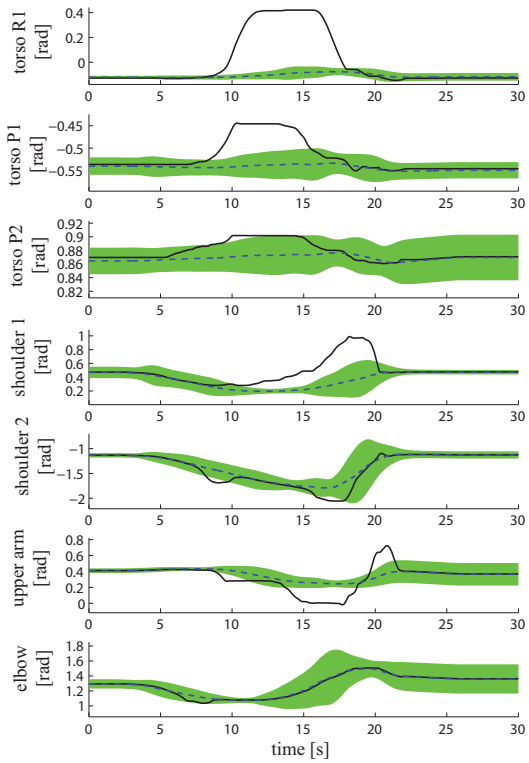

Fig. 8. Trajectories during kinesthetic teaching without the motion refinement tube. The dashed line shows the nominal trajectory and the solid line shows the measured joint angle trajectory. The green area around the nominal trajectory represents the tube, which was not used in the controller.

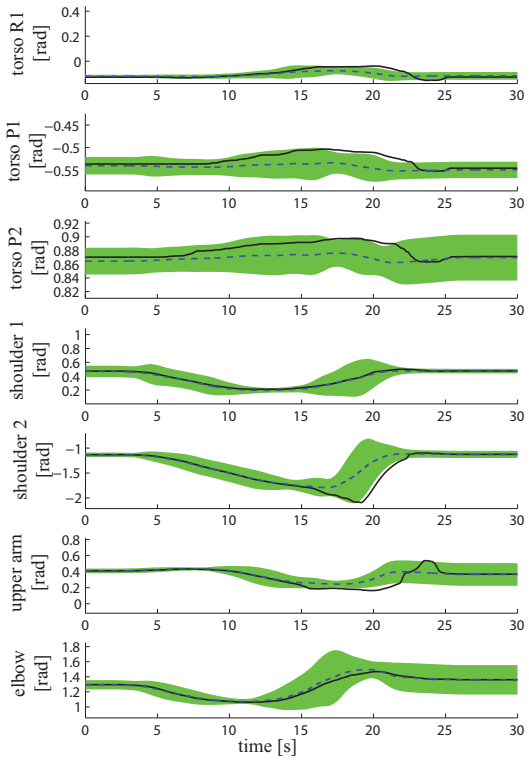

Fig. 9. Trajectories during kinesthetic teaching with the motion refinement tube. The dashed line shows the nominal trajectory and the solid line shows the measured joint angle trajectory. The green area around the nominal trajectory represents the motion refinement tube.

temporal and spatial data within a state. On the other hand, the proposed HMM representation can encode correlation between temporal and spatial data within a state and between states. This allows smooth trajectory generation (Sec. II-B), in contrast to the discontinuity of the generated trajectory at state transitions in conventional methods.

C. Kinesthetic Teaching using the Motion Refinement Tube

In the second experiment we focus on the refinement tube. For this, the robot initially learned a motion primitive corresponding to a "window wiping motion" performed mainly with the right hand. During the kinesthetic refinement, a human teacher tried to make the wiping motion larger. In contrast to the previous experiment, we used a motion primitive representation in joint space. Consequently, the motion retargetting was only necessary for the observational learning, and all subsequent refinement steps were performed using joint level data (see Fig. 1). In this way, we could directly use the covariance information from the probabilistic motion primitive description for the implementation of the motion refinement tube with the joint level impedance controller. In order to compute the feedforward terms in the impedance controller, the desired velocity and acceleration are generated by a second order prefilter with a cutoff frequency of 2 Hz.

We show a comparison between physical refinement with and without the refinement tube. In Fig. 8, a typical refinement result without the refinement tube is shown. In order to focus on the relevant parts of the motion, we show only the motion of the torso and the first four joints of the right arm (elbow, upper arm, and two for the shoulder) among 19 joints. The nominal trajectory during a second refinement step (after two observational demonstrations and one kinesthetic demonstration) is shown in the dashed line. The black line shows the measured motion during the physical interaction. In the case of Fig. 8, the tube was not implemented in the controller, such that when the human tried to refine the motion of the arm, he accidentally also modified the torso motion. Notice that the whole robot is controlled by the joint level impedance controller, which implements a soft behavior with respect to external disturbances when it is distracted from the nominal trajectory. As a consequence, the physical refinement becomes difficult for the human. In contrast to that, the implementation of the refinement tube allows to limit the motion of the joints during teaching in a controlled way. Figure 9 shows a similar result as in Fig. 8, but with the difference that now the refinement tube was activated. One can see that all the deviations from the nominal trajectory keep close to the borders of the refinement tube. In this way, the human can correct the desired part of the trajectory without accidentally disturbing the robot motion in an undesired way.

It should be mentioned that the practical use of the refinement tube requires also a meaningful design of the maximum torque parameters from Table I. In particular, one should choose higher values for joints located closer to the

base. This enables to refine motions of the extremities while not affecting the motion of the torso too much. Clearly, the refinement tube concept works better if the human acts "cooperatively" by trying to refine the motion only within the borders of the tube and does not try to "overcome" them. If a large modification is desired, the user can iteratively enlarge the size of the tube by moving the relevant joints.

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we treated the problem of incremental kinesthetic learning of motion primitives using an HMM based representation. We initiated the teaching process with observational learning prior to kinesthetic teaching in order to ensure natural coordinated whole body motions. We introduced a forgetting factor, which allows a more efficent learning and avoids the insensitivity problem related to batch learning with large training data sets. The concept of a motion refinement tube was introduced and its realtime implementation in a customized impedance controller was presented. The HMM representation of motion primitives was extended by incorporating relative time information in each state, allowing for a continuous generation despite of the discrete nature of the states in an HMM.

In the implementation so far, the refinement tube concept was realized by a joint level impedance controller. The extension of this concept to an appropriate Cartesian controller is the topic of our current research. Moreover, we plan to extend our current setup to all the degrees of freedom of Justin, i.e. including the fingers and the mobile base, in order to handle more complicated two-handed manipulation tasks.

ACKNOWLEDGMENTS

This work is supported in part within the DFG excellence initiative research cluster "Cognition for Technical Systems CoTeSys".

REFERENCES

[1] F. Kanehiro, W. Suleiman, F. Lamiraux, E. Yoshida, and J.-P. Laumond, "Integrating dynamics into motion planning for humanoid robots," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2008, pp. 660–667.

[2] Y. Nakamura and K. Yamane, "Dynamics computation of structure-varying kinematic chains and its application to human figures," *IEEE Transactions on Robotics*, vol. 16, no. 2, pp. 124–134, 2000.

[3] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots under obstacle and dynamic balance constraints," in *IEEE International Conference on Robotics and Automation*, 2001, pp. 692– 698.

[4] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 1398–1403.

[5] T. Inamura, Y. Nakamura, and I. Toshima, "Embodied symbol emergence based on mimesis theory," *Int. Journal of Robotics Research*, vol. 23, no. 4, pp. 363–377, 2004.

[6] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, vol. 37, no. 2, pp. 286–298, 2007.

[7] D. Lee and Y. Nakamura, "Mimesis model from partial observations for a humanoid robot," *Int. Journal of Robotics Research*, vol. 29, no. 1, pp. 60–80, 2010.

[8] S. Nakaoka, A. Nakazawa, F. Kanahiro, K. Kaneko, M. Morisawa, and K. Ikeuchi, "Task model of lower body motion for a biped humanoid robot to imitate human dances," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 2769–2774.

[9] N. Pollard, J. Hodgins, M. Riley, and C. Atkeson, "Adapting human motion for the control of a humanoid robot," in *IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 1390–1397.

[10] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura, "Whole body humanoid control from human motion descriptors," in *IEEE International Conference on Robotics and Automation*, 2008, pp. 2677–2684.

[11] Ch. Ott, D. Lee, and Y. Nakamura, "Motion capture based human motion recognition and imitation by direct marker control," in *IEEE-RAS International Conference on Humanoid Robots*, 2008.

[12] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in human-humanoid interaction," *Int. Journal of Robotics Research*, 2010, (accepted).

[13] H. Hoffmann, P. Pastor, D. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *IEEE Int. Conf. on Robotics and Automation*, 2009.

[14] M. Ito, K. Noda, Y. Hoshino, and J. Tani, "Dynamic and interactive generation of object handling behaviors by a small humanoid robot using a dynamic neural network model," *Neural Networks*, vol. 19, no. 3, pp. 323–337, 2006.

[15] S. Calinon and A. Billard, "Active teaching in robot programming by demonstration," in *IEEE International Conference on Robot and Human Interactive Communication*, 2007, pp. 702–707.

[16] T. Inamura, N. Kojo, and M. Inaba, "Situation recognition and behavior induction based on geometric symbol representation of multimodal sensorimotor patterns," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, p. 51475152.

[17] D. Kulić, W. Takano, and Y. Nakamura, "Combining automated on-line segmentation and incremental clustering for whole body motions," in *IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 2591–2598.

[18] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *ACM/IEEE International Conference on Human-Robot Interaction*, 2007, pp. 255–262.

[19] Ch. Ott, O. Eiberger, W. Friedl, B. Bäuml, U. Hillenbrand, Ch. Borst, A. Albu-Schäffer, B. Brunner, H. Hirschmüller, S. Kielhöfer, R. Konietschke, M. Suppa, T. Wimböck, F. Zacharias, and G. Hirzinger, "A humanoid two-arm system for dexterous manipulation," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 276–283.

[20] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, pp. 109–116, 2004.

[21] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[22] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, p. 129145, 1996.

[23] J. A. Blimes, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," University of Berkeley, Tech. Rep. ICSI-TR-97-021, 1997.

[24] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer Verlag, 2009.

[25] N. Hogan, "Impedance control: An approach to manipulation, part I - theory," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1–7, 1985.

[26] A. Albu-Schäffer, Ch. Ott, U. Frese, and G. Hirzinger, "Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms," in *IEEE International Conference on Robotics and Automation*, 2003, pp. 3704–3709.

[27] B. Paden and R. Panja, "Globally asymptotically stable 'pd+' controller for robot manipulators," *International Journal of Control*, vol. 47, no. 6, pp. 1697–1712, 1988.

[28] Xsens moven. [Online]. Available: http://www.moven.com

[29] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[30] M. Spong, "Modeling and control of elastic joint robots," *Transactions of the ASME: Journal of Dynamic Systems, Measurement, and Control*, vol. 109, pp. 310–319, 1987.

[31] Ch. Ott, A. Albu-Schäffer, and G. Hirzinger, "Comparison of adaptive and nonadaptive tracking control laws for a flexible joint manipulator," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, 2002, pp. 2018–2024.

[32] D. Kulić, W. Takano, and Y. Nakamura, "Representability of human motions by factorial hidden markov models," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007.