Iterative Motion Primitive Learning and Refinement by Compliant Motion Control

Dongheui Lee and Christian Ott

***Abstract*— We present an approach for motion primitive learning and refinement for a humanoid robot. The proposed teaching method starts with observational learning and applies iterative kinesthetic motion refinement. Observational learning is realized by a marker control approach. Kinesthetic teaching is supported by introducing the motion refinement tube in order to avoid to disturb other joints accidentally. The proposed teaching method allows for iterative execution and motion refinement using a forgetting factor. On the realtime control level, the kinesthetic teaching is handled by a customized Cartesian impedance controller, which combines tracking performance with compliant physical interaction and allows to implement soft boundaries for the motion refinement. A new method for continuous generation of motions from a hidden Markov model (HMM) representation of motion primitives is proposed. The proposed methods were implemented and tested using DLR's humanoid upper-body robot Justin.**

I. INTRODUCTION

Imitation learning nowadays is a well established concept in the robotics community. The paradigm of imitation learning aims at providing a nonexpert user with an intuitive interface for teaching the robot various tasks ranging from simple movements to manipulation skills. Often imitation learning framework employs the concept of motion primitives [1]–[3], since this allows a compact description of general motion patterns.

Observational demonstrations and kinesthetic demonstrations have been widely used in imitation learning. In observational demonstrations, the teacher demonstrates a motion by self performance (Fig. 1 left). In this case, differences in the teacher's and robot's kinematics, as well as dynamical constraints, must be taken into account. In our work, this motion retargetting problem is solved by a marker control approach [4], which lead to natural synchronized whole body motions. On the other hand, in kinesthetic teaching, the teacher physically guides the robot through the motion (Fig. 1 right). In most works [5], [6], kinesthetic teaching was realized either by deactivating the controlled motion of individual joints or by using very low servo gains. As a consequence, these approaches often lead to unsynchronized motions because the teacher moves motors one by one rather than demonstrating natural coordinated movements. These limitations can be overcome by combining imitation of human's whole body motion with a compliant behavior for physical interaction. By utilizing physical human robot interaction control approaches, we achieve kinesthetic teaching while keeping synchronization of whole body motion.

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 [7], [8], where the knowledge of motion primitives is updated as more demonstrations are provided, without keeping all the training data in a database. 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 our method, this problem is handled by integrating a forgetting factor into the iterative learning algorithm which leads to exponential forgetting of old data and thus results in an effective correction of kinematic mapping errors.

Fig. 1. Teaching strategies: observational demonstration (left) and kinesthetic demonstration (right).

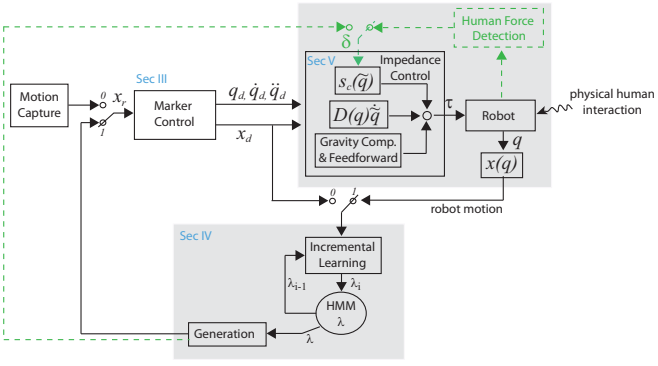

Fig. 2. System overview of the proposed approach. Acquired motion primitives are iteratively refined by physical interaction with a human teacher.

II. PROPOSED FRAMEWORK

In order to achieve intuitive teaching of natural motions, we propose a method for incremental learning by using

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

Ch. Ott is with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR e.V.). christian.ott@dlr.de

physical interaction. A schematic overview is shown in Fig. 2. In order to ensure synchronization of complex whole body motions on a humanoid robot, our imitation learning procedure starts with observation learning prior to kinesthetic motion refinements. These trajectories are implemented via a customized impedance controller (Sec. V). In parallel, an HMM (hidden Markov model) based motion primitive learning algorithm is active. During the kinesthetic teaching, the reference trajectory and the parameters for a refinement tube δ are generated from the HMM (Section IV) by a novel continuous generation method. By a customized impedance control, the user can correct undesired aspects of the retargetted motion resulting from kinematic differences and mapping errors, without accidentally disturbing the robot motion in an undesired way.

The proposed concepts are not limited to a specific motion representation (joint or task space). Due to the limited space, this paper explains task space implementation only. For joint space implementation, see our previous paper [9].

III. Human Motion Imitation

In order to imitate human's whole body motions, we proposed a marker control approach [4]. The main idea is to connect virtual springs between the marker positions on a human and corresponding points on the robot (see Fig. 3). When the cloud of marker points moves in space, the robot's motion will be guided by the forces generated from these springs. Instead of implementing the virtual springs directly, we let them act on a simplified simulation of the robot upper body dynamics including a free floating base link. This allows to implement the approach on a position controlled robot.

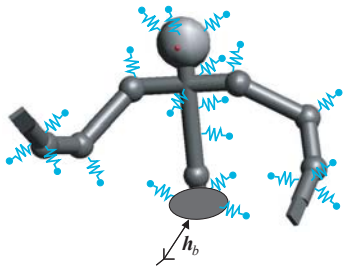

Fig. 3. Marker control: Virtual springs are connected between the reference marker points and corresponding points on the humanoid robot.

IV. Incremental Motion Learning

A. Motion Primitive

Our incremental motion learning approach is based on a motion primitive representation. A compact motion representation is desired for incremental learning from knowledgeable agents. In order to achieve real-time smooth interaction with a human, on-line recognition of human's motions without preprocessing continuous incoming motion data and a continuous motion reproduction without postprocessing for smoothing the trajectory are important. By considering these mentioned demands, we propose to combine HMM and a Gaussian regression technique. Motion primitive representation in this work follows basic HMM properties, which allows on-line recognition. 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 explicitly estimated. Our method allows to handle data sequences of different duration and speed, even whose speed is time-varying within a motion primitive. A preprocessing of observation (e.g., scaling in time or dynamic time warping) for learning and recognition is not needed.

TABLE I
Motion Primitive

Each motion primitive is represented as a left-right type HMM with N states. HMM parameters consist of π_i (the probability for the initial state to be i-th state), a_{ij} (the probability to transit from state i to state j), and $b_i(\boldsymbol{o})$ (the probability density function for the output of a vector $\boldsymbol{o}$ at state i). Herein, a Gaussian distribution is used: $b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the mean vector and the covariance matrix for the Gaussian in state i,

$$\boldsymbol{\mu}_i = \begin{bmatrix} {}^t\mu_i \\ {}^s\boldsymbol{\mu}_i \end{bmatrix}, \quad \boldsymbol{\Sigma}_i = \begin{bmatrix} {}^{tt}\Sigma_i & {}^{ts}\boldsymbol{\Sigma}_i \\ {}^{st}\boldsymbol{\Sigma}_i & {}^{ss}\boldsymbol{\Sigma}_i \end{bmatrix},$$

where the symbol ${}^{ts}\boldsymbol{\Sigma}_i$ is the covariance vector between temporal and spatial data of the Gaussian at state i.

The training algorithm is as follows. Note that A sequence of motion $\mathcal{O} = \{\boldsymbol{o}(t)\}$ consists of temporal data ${}^t\mathcal{O} = \{{}^to(t)\}$ and spatial data ${}^s\mathcal{O} = \{{}^s\boldsymbol{o}(t)\}$. In this paper, the temporal data is simply a timestamp: ${}^to(t) = t$. The denotation of the motion primitive is explained in Table I. First, from a motion sequence of spatial data ${}^s\mathcal{O} = \{{}^s\boldsymbol{o}(t)\}$ [1], HMM parameters are trained via an EM algorithm [10]. In case of incremental learning, see Sec. IV-C. Then, the optimal state sequence $\mathcal{Q} = \{q(t)\}$ for the spatial data sequence ${}^s\mathcal{O}$ is calculated via the Viterbi algorithm [10]. This correspondence of $q(t)$ to ${}^s\boldsymbol{o}(t)$ applies also to temporal data ${}^to(t)$. Thereafter, by using the affiliated temporal data to each state, the temporal mean ${}^t\mu$ and variance ${}^t\boldsymbol{\Sigma}$ for each state is estimated. 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

From an HMM, a motion sequence can be decoded using the expectation operator in the stochastic model. The generation procedure of the new proposed motion primitive (Sec. IV-A) is as follows. Motion patterns are decoded using the expectation operator in the stochastic model.

Step 1) Since left-right type of HMM is used, the total duration T of the motion primitive is estimated by using temporal mean and variance of the last state as follows.

$$T = {}^t\boldsymbol{\mu}_N + \frac{-3 + \sqrt{1 + 48\ {}^{tt}\Sigma_N}}{4}$$

Herein, N is the number of states in an HMM. Then, the temporal sequence ${}^t\mathcal{O}$ becomes

$${}^to(t) = t, \quad \forall t = 1...T \quad .$$

[1]The spatial data ${}^s\boldsymbol{o}(t)$ can be in joint coordinates and/or Cartesian coordinates. In this paper, the case of task space is shown (Fig. 2).

Step 2) For the temporal data sequence ${}^t\mathcal{O}$, the responsibility $\gamma_i(t)$ of each state $i = 1...N$ at time $t = 1...T$ is calculated as

$$\zeta_i(t) = \frac{\alpha_i({}^t o(t))\beta_i({}^t o(t))}{\sum_{j=i}^{N} \alpha_j({}^t o(t))\beta_j({}^t o(t))} \quad , \tag{1}$$

which is the probability of being in state i at time t for the given observation sequence ${}^t\mathcal{O}$, rather than for a data point ${}^t o(t)$. Herein, $\alpha_i({}^t o(t))$ and $\beta_i({}^t o(t))$ are modified HMM forward and backward variables for the temporal sequence.

$$\begin{aligned} \alpha_i({}^t o(t)) &= P({}^t o(1)\ {}^t o(2) \cdots\ {}^t o(t), q_t = S_i|\lambda) \\ \beta_i({}^t o(t)) &= P({}^t o(t+1)\ {}^t o(t+2) \cdots\ {}^t o(T)|q_t = S_i, \lambda) \end{aligned}$$

Step 3) Given the temporal data ${}^t o(t)$, a sequence of spatial data is generated using Gaussian regression [11]. Considering the the responsibility of each state for ${}^t o(t)$, the conditional expectation ${}^s\boldsymbol{o}(t)$ is

$${}^s\boldsymbol{o}(t) = \sum_{i=1}^{N} \zeta_i(t)\ \{{}^s\boldsymbol{\mu}_i + \frac{{}^{ts}\boldsymbol{\Sigma}_i}{{}^{tt}\Sigma_i}({}^t o(t) - {}^t\mu_i)\} \quad . \tag{2}$$

The conditional variance of ${}^s\boldsymbol{o}(t)$ is estimated as

$${}^{s|t}\boldsymbol{\Sigma} = \sum_{i=1}^{N} \zeta_i(t)\ \{\mathrm{diag}({}^{ss}\boldsymbol{\Sigma}_i) - {}^{st}\boldsymbol{\Sigma}_i\ {}^{tt}\Sigma_i^{-1}\ {}^{ts}\boldsymbol{\Sigma}_i\} \quad . \tag{3}$$

C. Incremental Learning with a forgetting factor

The incremental learning method herein is a variation of the EM algorithm for multiple observations [10]. 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 by the algorithm in Sec. IV-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 $\bar{\lambda} = \{\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{a}}, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\}$ for the HMM are estimated by using the old HMM parameters $\lambda = \{\boldsymbol{\pi}, \boldsymbol{a}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ 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 Gaussian distributions are as follows.

$$\begin{aligned} {}^s\bar{\boldsymbol{\mu}}_i &= \frac{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e} \gamma_i^e(t)\, {}^s\boldsymbol{o}^e(t)}{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e} \gamma_i^e(t)} \\ {}^{ss}\bar{\boldsymbol{\Sigma}}_i &= \frac{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e} \gamma_i^e(t)({}^s\boldsymbol{o}^e(t) - {}^s\boldsymbol{\mu}_i)({}^s\boldsymbol{o}^e(t) - {}^s\boldsymbol{\mu}_i)^T}{\sum_{e=1}^{E} w^e \sum_{t=1}^{T_e} \gamma_i^e(t)} \end{aligned}$$

D. Motion Refinement Tube

During kinesthetic teaching, the conditional variance ${}^{s|t}\boldsymbol{\Sigma}$ in (3) is used for designing the motion refinement tube. The tube is centered around ${}^s\boldsymbol{o}(t)$ and has a radius of

$$\boldsymbol{\delta} = 3\sqrt{{}^{s|t}\boldsymbol{\Sigma}} + \epsilon \quad , \tag{4}$$

where ϵ is the minimum allowance. This radius $\boldsymbol{\delta}$ is used as an input parameter of the impedance controller presented in Sec. V (See Fig. 4).

During the refinement process, only newly corrected motion elements are updated by incremental learning in Sec. IV-C in order to avoid decreasing the tube size for unnecessary parts. If a very large motion refinement is desired, the user can iteratively enlarge the size of the tube by moving the relevant joints within the range of the tube. In this way, the human can produce large modifications of selected joints/segments even when the relevant tube initially is small.

V. PHYSICAL MOTION REFINEMENT BY COMPLIANT MOTION CONTROL

A. Requirements

The physical motion refinement strategy described in this paper is based on a compliant motion control approach. The underlying real-time controller should allow to achieve good trajectory tracking performance when there is no physical interaction, but should at the same time implement a compliant (low stiffness) behavior when the robot is distracted from the nominal trajectory during kinesthetic teaching. Moreover, it is required to limit the allowed deviation from the nominal trajectory in order to implement the motion refinement tube.

Our approach to this compliant motion control problem is to integrate the above mentioned requirements into an appropriate customized impedance controller. Let us assume that the learned motion data is represented by a set of m reference positions $\boldsymbol{p}_{\mathrm{ref},i} \in \mathbb{R}^3$ and orientations $\boldsymbol{R}_{\mathrm{ref},i} \in SO(3)$, for which we have identified corresponding frames on the humanoid body. Additionally, for each of the m frames the desired size of the refinement tube is given by two scalar parameters $\delta_{t,i}$ and $\delta_{r,i}$ for translation and rotation, respectively. Moreover, we assume an overall control structure

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

as in Fig. 2. Here the desired trajectory for the underlying real-time controller is generated from the Cartesian reference trajectories by a motion retargetting algorithm, e.g. the marker control algorithm from [4]. Therefore, we assume that the desired trajectory is given by a twice differentiable desired joint angle trajectory $\boldsymbol{q}_d(t)$. Clearly, from $\boldsymbol{q}_d(t)$ also the desired position $\boldsymbol{p}_{d,i} \in \mathbb{R}^3$ and orientation $\boldsymbol{R}_{d,i}$ for the m reference frames on the robot are known from the forward kinematics mappings.

B. Impedance design

For the design of the impedance controller, we assume a rigid-body model of the robot with n DOFs [12]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}} \tag{5}$$

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 $\dot{\boldsymbol{M}} = \boldsymbol{C} + \boldsymbol{C}^T$ [12]. 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 $\boldsymbol{\tau}_{\text{ext}}$ [13]. Our strategy for the impedance design is as follows. We aim at an impedance relation, which represents a passive mapping from the velocity error $\dot{\tilde{\boldsymbol{q}}} = \boldsymbol{q} - \boldsymbol{q}_d$ to the external torques $\boldsymbol{\tau}_{ext}$ ensuring the stability of the system in free motion and in feedback interconnection with a passive environment. 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 if the desired inertia is chosen equal to the natural inertia of the robot [14]. As an impedance behavior we choose a behavior similar to the closed loop behavior of a PD+ like tracking controller [15] with customized stiffness and damping terms. The chosen behavior is given by

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\tilde{\boldsymbol{q}}} + (\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{D}(\boldsymbol{q}))\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{s}(\boldsymbol{q},\boldsymbol{q}_d) = \boldsymbol{\tau}_{\text{ext}} \ , \tag{6}$$

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 $\boldsymbol{s}(\boldsymbol{q},\boldsymbol{q}_d)$ 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 (6) makes allowance for the position-dependence of the inertia term.

By comparing the desired impedance (6) with the original robot dynamics (5), we obtain the impedance control law

$$\boldsymbol{\tau} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}_d + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_d - \boldsymbol{D}(\boldsymbol{q})\dot{\tilde{\boldsymbol{q}}} - \boldsymbol{s}(\boldsymbol{q},\boldsymbol{q}_d) \ , \tag{7}$$

for which we still have to select appropriate stiffness and damping terms which allow for good tracking in free motion, compliant physical interaction during kinesthetic refinement, and an implementation of the refinement tube for translation $\delta_{t,i}$ and orientation $\delta_{r,i}$.

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

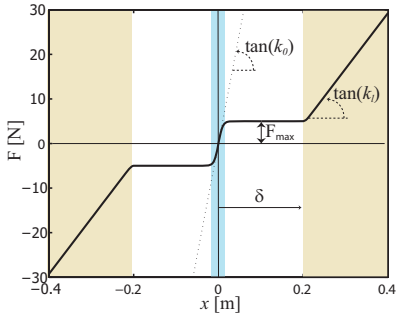

Fig. 4. Nonlinear stiffness term for the refinement tube implementation. The blue area shows the domain of high stiffness. The white area shows the coaching area in which a human user feels zero local stiffness, but a small constant force pushing towards the virtual equilibrium trajectory. The yellow area shows the border of the refinement tube implemented via a *soft constraint*.

The design of the stiffness term $\boldsymbol{s}(\boldsymbol{q},\boldsymbol{q}_d)$ is based on the scalar nonlinear function

$$f(x,\chi) := F_{\max}\tanh\left(\tfrac{k_0}{\tau_{\max}}x\right) + f_l(x) \ , \tag{8}$$

$$f_l(x) = \begin{cases} 0 & |x| < \delta \\ \operatorname{sign}(x)\ln(\cosh(k_t(\delta - |x|))) & |x| \geq \delta \end{cases} .$$

shown in Fig. 4, which depends on the design parameters $\chi = (k_0, k_l, F_{\max}, \delta)$. The potential function related to this stiffness term is given by $V_s(x,\chi) = \int_0^x f(\xi,\chi)d\xi$.

For each reference marker position and orientation, we define a Cartesian potential function of the form

$$V_{c,i}(\boldsymbol{q}) = V_s(e_{t,i}(\boldsymbol{q}),\chi_{t,i}) + V_s(e_{r,i}(\boldsymbol{q}),\chi_{r,i}) \tag{9}$$

where $e_{t,i}(\boldsymbol{q})$ and $e_{r,i}(\boldsymbol{q})$ are scalar position and orientation error functions between the reference data and the corresponding frames on the humanoid robot. The parameter sets $\chi_{t,i} = (k_{t0,i}, k_{tl,i}, F_{\max,i}, \delta_{t,i})$ and $\chi_{r,i} = (k_{r0,i}, k_{rl,i}, M_{\max,i}, \delta_{r,i})$ contain the stiffness parameters and refinement tube sizes as well as the maximum force and torque values for the nonlinear stiffness functions. The Cartesian stiffness term related to these potential functions can be implemented by

$$\boldsymbol{s}(\boldsymbol{q},\boldsymbol{q}_d) := \frac{\partial}{\partial \boldsymbol{q}}\left(\sum_{i=1}^{m} V_{c,i}(\boldsymbol{q})\right) \ . \tag{10}$$

Notice that in (7) we combine the Cartesian stiffness terms for the m reference marker positions and orientations with compatible feedforward terms from a joint level trajectory. The chosen solution does not require a prioritization of the Cartesian reference points. For the design of the damping matrix $\boldsymbol{D}(\boldsymbol{q})$, we utilize the *double diagonalization* method reported in [14] based on the symmetric matrices $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{K} := \partial \boldsymbol{s}(\boldsymbol{q},\boldsymbol{q}_d)/\partial \boldsymbol{q}$.

VI. EXPERIMENTS

The aforementioned methods are implemented and evaluated using DLR's humanoid upper-body robot Justin. In the experiments, we used the 19 joints of the arms (2 times 7

DOF), torso (3 DOF), and head (2 DOF) except the mobile base and the fingers. As a motion capture system, we use the wearable motion capture suit MOVEN from the company XSens. Among motion capture sensory information, 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). Figure 1 left subfigure shows a snapshot of realtime human motion imitation by the marker control algorithm.

For the kinesthetic teaching, the robot uses the force based impedance behavior from Sec. V. The robot Justin is equipped with joint torque sensors in all the joints except for the neck and the mobile base. Therefore, we implement the impedance controller for the torso and the arms, while we use a standard position controller for the neck.

In the first experiment, we are evaluating the incremental learning and refinement of motion primitives. Figure 5 shows a qualitative presentation of the incremental learning including the 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 and retargetted to the robot by the marker control algorithm. 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 (Fig. 5(a)). Therefore, the human teacher corrects the right hand motion by pulling it down during the robot's execution (Fig. 5(b)). The human refined the robot's whole body motion four times in total, by pulling the right hand down, rotating the torso, and positioning the left hand away from its mobile base. Figure 5(c) shows the generalized motion primitive after four refinement steps with a forgetting factor. Here, the robot's hand is moving approximately in the correct height.

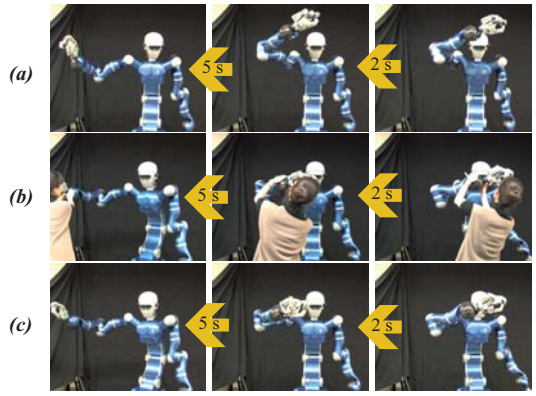

Fig. 5. Kinesthetic motion refinement: Subfigure (a) (from right to left) shows the snapshots from the original motion primitive. In (b) and (c), the snapshots during and after the physical coaching for refinement are shown.

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. For an HMM, ten states and one Gaussian model for each state are used. The generalized motion primitive (red solid line) is generated from the HMM parameters by the algorithm in Sec. IV-B. The new HMM parameters are updated using the old HMM parameters and the new incoming observation by the proposed incremental learning algorithm in Sec. IV-C.

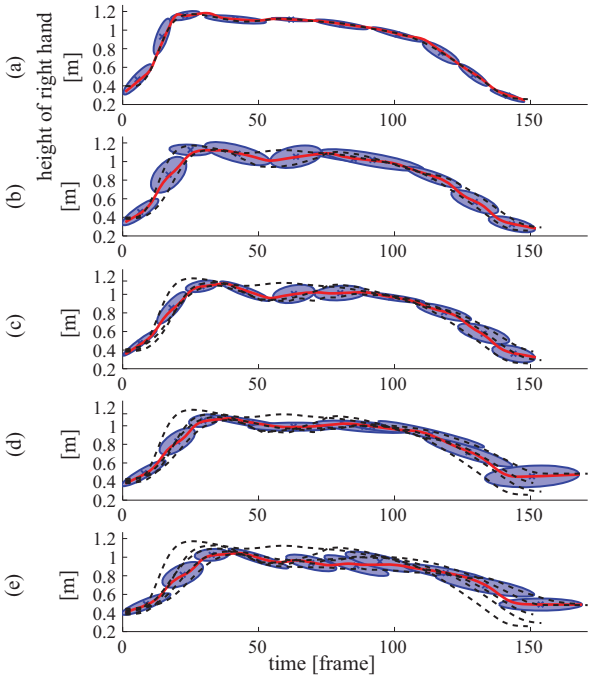

Fig. 6. Results of the proposed incremental learning and generation at each step: (a) from the first training data and (e) after five training steps. The right hand height trajectories are shown. The black dashed curves are the training data provided so far. The red solid curve is the generalized trajectory of the current motion primitive.

In the second experiment, we show an evaluation of the refinement tube concept. For the purpose of a clear evaluation independently of the variability of human interaction, we utilize a deterministic force generation. We 'simulate' a deterministic teaching force by generating it directly in the controller. In addition to the impedance relation (7), we superpose the joint torque corresponding to a force of $55\ N$ acting at the right elbow. The force profile is generated by a step response of a second order filter with a cutoff frequency of 1 Hz. Here, we compare the Cartesian errors for the torso orientation, the right elbow orientation, and the right hand position and orientation. The relevant impedance parameters are given in Table II. Figure 7 shows the Cartesian errors compared to the chosen radii of the refinement tubes. The tube radii were selected manually for this comparison. The solid lines show the control errors when all tube radii were used as displayed in Fig. 7, while the dashed lines show the control error for the case without using the motion refinement tube, i.e. setting δ_i to very high values for the torso orientation. Here one can see that the size of the motion refinement tube can be used as a tool for enabling/disabling motion refinement in a certain range for selected motion

TABLE II

CARTESIAN IMPEDANCE PARAMETERS FOR THE KINESTHETIC TEACHING

	k_0	k_l	$F_{\max}$
torso orientation	800 [Nm/rad]	1600 [Nm/rad]	3 [Nm]
elbow position	1500 [N/m]	1500 [N/m]	30 [N]
hand position	1500 [N/m]	1500 [N/m]	30 [N]
hand orientation	100 [Nm/rad]	100 [Nm/rad]	3 [Nm]

elements. In this way, the user can intuitively select, which parts of the motion he wants to refine and which should be excluded in the refinement. In combination with an estimation of the interaction force, such a selection process could also be made automatically.

One can see that all the deviations from the nominal trajectory keep close to the borders of the refinement tube by 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. This experiment confirms the behavior of the refinement tube by a deterministic force generation.

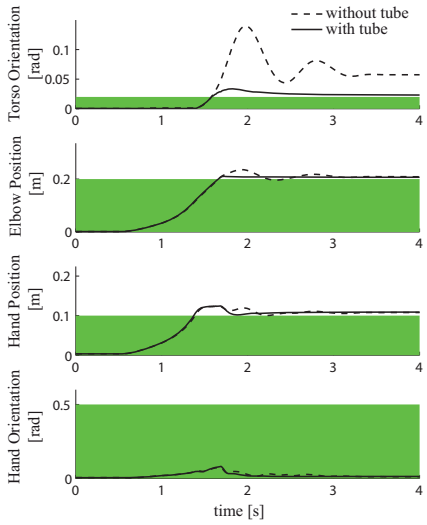

Fig. 7. Evaluation of the refinement tube in Cartesian space. The Cartesian errors for the torso orientation, the right arms elbow position, and the hand position and orientation are shown. The dashed line shows the result without the refinement tube for the torso orientation.

VII. CONCLUSIONS

This paper shows our recent work on incremental kinesthetic learning of motion primitives in the paradigm of programming by demonstration. As a baseline, an imitation method for commanding humanoid whole body motion was shown. Then, a refinement process of learned motion primitives using kinesthetic teaching was proposed. This ensures natural coordinated whole body motions. In order to avoid the insensitivity problem related to batch learning with large training data sets, we introduced a forgetting factor in the iterative learning procedure. 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 time information in each state, allowing for a continuous motion generation despite of the discrete nature of the states in an HMM. The concepts were evaluated using the humanoid upper body robot Justin.

ACKNOWLEDGMENTS

This research is partly supported by the DFG excellence initiative research cluster "Cognition for Technical Systems CoTeSys".

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] C. 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] D. Lee and C. Ott, "Incremental motion primitive learning by physical coaching using impedance control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2010.

[10] 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.

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

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

[13] 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.

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

[15] 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.