Generalization of Discrete Compliant Movement Primitives

Miha Deniša, Andrej Gams, Aleš Ude and Tadej Petrič
Humanoid and Cognitive Robotics Lab
Department of Automatics, Biocybernetics and Robotics
Jožef Stefan Institute
1000 Ljubljana, Slovenia
Email: {miha.denisa, andrej.gams, ales.ude, tadej.petric}@ijs.si

***Abstract*—This paper addresses the problem of achieving high robot compliance while maintaining low tracking error without the use of dynamical models. The proposed approach uses programing by demonstration to learn new task related compliant movement. The presented Compliant Movement Primitives are a combination of 1) position trajectories, gained through human demonstration and encoded as Dynamical Movement Primitives and 2) corresponding torque trajectories encoded as a linear combination of radial basis functions. A set of example Compliant Movement Primitives is used with statistical generalization in order to execute previously unexplored tasks inside the training space. The proposed control approach and generalization was evaluated with a discrete pick-and-place task on a Kuka LWR robot. The evaluation showed a major decrease in tracking error compared to a classic feedback approach and no significant rise in tracking error while using generalized Compliant Movement Primitives.**

I. Introduction

Programming by demonstration (PbD) [1]–[4] is a popular way of gaining new sensorimotor knowledge through human demonstration. Learning of trajectories may rely on a single example demonstration, e.g., training of dynamic movement primitives (DMPs) [5], [6]. It can also be done with the use of multiple demonstrations. Statistical generalization can employ multiple demonstrations and synthesize an appropriate trajectory for a new task within the training space from a set of recorded movements [7]–[9].

Another aspect addressed in this paper is the robot's reaction to unforeseen contacts with the environment. This is especially important when the robot shares its workspace with humans. One way of minimizing impact forces is by detecting and interpreting contacts. This can be done by using sensitive skin, represented by a soft tactile sensing array [10], [11] or by a bumper based hard shell [12]. A robot can also be made safer without the use of additional sensors by making it compliant. They can be made passively compliant by design [13] by reducing the weight and hardness of the robot structure [14], [15] or through actuator design by implementing elastic elements [16], [17].

As an alternative to passive approaches, compliance can also be achieved either through advanced control approaches [18], [19] or through torque control strategies [20], [21]. But besides having access to actual torques through sensors, a correct dynamic model for each task variation must be available. The task-specific dynamic can be obtained with mathematical modeling of the task [22], [23]. However, mathematical modeling is a difficult and time consuming task that can only be performed by an expert. As an alternative, different biologically inspired methods were proposed for dynamic robot control [24].

This paper proposes achieving active compliance through torque control. But instead of using a dynamical model for each task variant, we introduce Compliant Movement Primitives (CMPs), which are inspired by the human sensorimotor ability to learn arbitrary dynamic tasks. The first step in the process of gaining CMPs is the demonstration of the desired motion trajectory and its encoding as DMP. This trajectory is then executed on a robot while using a high-gain feedback control. The actually executed torque signals are recorded and then encoded as a linear combination of radial basis functions, defined as a Torque Primitive (TP). The task-related position trajectory supplemented with the corresponding torques is denoted as a CMP. The proposed control approach uses the torque component as a feedforward term while using the position trajectory in a low-gain feedback loop. This ensures that low gains can be used for control and consequently the desired task can be performed safely.

While the proposed approach eliminates the need for dynamical modeling, a CMP needs to be learned for each task variation. This can be mitigated through generalization. An example set of demonstrated CMPs is generalized with statistical methods in order to compute a new CMP suitable for the desired task variant inside the training space. With this addition to the proposed approach the robot is able to learn and perform variations of the (semantically) same task in a compliant manner without the need for experts to program movements and dynamical models.

II. Compliant Movement Primitives

A compliant movement, which accomplishes a task in a compliant manner, is defined in this paper as a combination of desired position trajectories and corresponding torques signals

$$\boldsymbol{h}(t) = [\boldsymbol{q}_d(t), \boldsymbol{\tau}_f(t)]. \quad (1)$$

978-1-4673-7509-2/15/$31.00 2015 IEEE

The kinematic part $\boldsymbol{q}_d(t)$ represents desired joint trajectories of the compliant movement,

$$\boldsymbol{q}_d(t) = [q_{d1}(t),\ q_{d2}(t), \ldots,\ q_{dn}(t)], \tag{2}$$

where n denotes number of degrees of freedom (DOF). The second part presents the dynamics of the movement, i.e., corresponding joint torque signals

$$\boldsymbol{\tau}_f(t) = [\tau_{d1}(t),\ \tau_{d2}(t), \ldots,\ \tau_{dn}(t)]. \tag{3}$$

This section of the paper presents a proposed approach for learning compliant movements through human demonstration, encoding them as a combination of linear basis functions, and a proposed control approach for their execution.

A. Learning compliant movements

The kinematic part of the compliant movement is gained from human demonstration. Although various techniques for human motion capture can be used [25]–[27], kinesthetic guidance [28] is used in this paper. With it, a set of example motion trajectories is captured in joint space,

$$\boldsymbol{Q}_x = \{\{\boldsymbol{q}_{x1}(t), \boldsymbol{c}_{q_1}\},\ \{\boldsymbol{q}_{x2}(t), \boldsymbol{c}_{q_2}\}, \ldots,\ \{\boldsymbol{q}_{xN}(t), \boldsymbol{c}_{q_N}\}\}, \tag{4}$$

where x stands for *examples* and N denotes the total number of captured examples. Each example trajectory successfully accomplishes a version of the task described through query $\boldsymbol{c}_q$. These N kinematic task descriptors influence the position part of the CMP and can denote a variety of things, e.g., initial position of a movement in task space that varies in a single dimension, $c_q = x_{\text{start}}$, final position of the discrete movement in joints space, $\boldsymbol{c}_q = [q_1, q_2, \ldots, q_n]$, the position of a known obstacle, $\boldsymbol{c}_q = [p_x, p_y]$, etc.

The dynamic part of the compliant movement is gained through execution of captured kinematic trajectories on a robot. Demonstrated movements $\boldsymbol{q}_x$ are executed in a controlled environment or under human supervision using a high gain feedback controller

$$\boldsymbol{\tau}_u = \mathbf{K}(\boldsymbol{q}_x - \boldsymbol{q}) + \mathbf{D}(\dot{\boldsymbol{q}}_x - \dot{\boldsymbol{q}}), \tag{5}$$

where high values of $\mathbf{K}$ ensure the required tracking accuracy (for further details on control framework see the Appendix). The $\boldsymbol{\tau}_x$ are measured on each joint during execution,

$$\boldsymbol{\tau}_x = \boldsymbol{\tau}_m. \tag{6}$$

With the above equations we assume that the robot's dynamical model is not known beforehand. In this case the corresponding torque signals would be used to compensate for task-specific dynamics and robot's own dynamics. In the case of known robot's dynamic model, (5) changes to

$$\boldsymbol{\tau}_u = \mathbf{K}(\boldsymbol{q}_x - \boldsymbol{q}) + \mathbf{D}(\dot{\boldsymbol{q}}_x - \dot{\boldsymbol{q}}) + \boldsymbol{f}_{dynamic}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}), \tag{7}$$

with $\boldsymbol{f}_{dynamic}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ representing the model. In this case the corresponding torques (6) are gained by subtracting the known robot's dynamics from actual measured joint torques,

$$\boldsymbol{\tau}_x = \boldsymbol{\tau}_m - \boldsymbol{f}_{dynamic}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}). \tag{8}$$

The dynamic part of the compliant movement now compensates for the task dynamics and mitigates any model's errors.

The same demonstrated kinematic trajectory $\boldsymbol{q}_x$ can be executed multiple times under different conditions which vary the dynamic part of the task. Each of these dynamic task descriptors $\boldsymbol{c}_\tau$ will produce a different compliant movement with same kinematic parts but varying dynamic parts. For example, the same kinematic trajectory can be executed at varying velocities and produce compliant movements with different corresponding torques. Similar, moving objects of varying mass $c_\tau = m$ over the same position trajectory will produce corresponding torques that differ for each mass. Thus we can obtain a set of example torque signals,

$$\boldsymbol{T}_x = \{\{\boldsymbol{\tau}_{x1}, \boldsymbol{c}_{\tau 1}\},\ \{\boldsymbol{\tau}_{x2}, \boldsymbol{c}_{\tau 2}\}, \ldots,\ \{\boldsymbol{\tau}_{x(NM)}, \boldsymbol{c}_{\tau(NM)}\}\}, \tag{9}$$

where M is the number of times each example motion trajectory $\boldsymbol{q}_{xi}$, was executed with varying conditions $\boldsymbol{c}_{\tau(ij)}$, $i = 1, \ldots, N$, $j = 1, \ldots, M$.

Through human demonstration and execution under varying condition we can learn a set of total NM example compliant movements, i.e., pairs of motion trajectories and corresponding torques,

$$\begin{aligned}\boldsymbol{H}_x = \{&\{\boldsymbol{q}_{x,1}, \boldsymbol{\tau}_{x,1}\}, \ldots,\ \{\boldsymbol{q}_{x,1}, \boldsymbol{\tau}_{x,M}\},\\ &\{\boldsymbol{q}_{x,2}, \boldsymbol{\tau}_{x,M+1}\}, \ldots,\ \{\boldsymbol{q}_{x,2}, \boldsymbol{\tau}_{x,2M}\}, \ldots,\\ &\{\boldsymbol{q}_{x,N}, \boldsymbol{\tau}_{x,(N-1)M+1}\}, \ldots,\ \{\boldsymbol{q}_{x,N}, \boldsymbol{\tau}_{x,(NM)}\}\},\end{aligned} \tag{10}$$

which can be used to compliantly execute tasks under varying conditions defined by query points,

$$\begin{aligned}\boldsymbol{C}_x = \{&\{\boldsymbol{c}_{q_1},\ \boldsymbol{c}_{\tau,1}\}, \ldots,\ \{\boldsymbol{c}_{q_1},\ \boldsymbol{c}_{\tau,M}\},\\ &\{\boldsymbol{c}_{q_2},\ \boldsymbol{c}_{\tau,M+1}\}, \ldots,\ \{\boldsymbol{c}_{q_2},\ \boldsymbol{c}_{\tau,2M}\}, \ldots,\\ &\{\boldsymbol{c}_{q_N},\ \boldsymbol{c}_{\tau,(N+1)M+1}\}, \ldots,\ \{\boldsymbol{c}_{q,N},\ \boldsymbol{c}_{\tau,(NM)}\}\}.\end{aligned} \tag{11}$$

B. Encoding CMPs

Each learned movement is encoded as a Compliant Movement Primitive (CMP). Motion trajectories are encoded as Dynamic Movement Primitives (DMPs), while corresponding torque signals are encoded as a linear combination of Gaussian functions, defined in this paper as a Torque Primitive (TP). A CMP is defined as a combination of a motion trajectory, encoded as a DMP, and corresponding torques, encoded as a TP.

The motion trajectories are encoded as DMPs. Details regarding DMPs are omitted and the readers are referred to [5]–[7]. Corresponding torques $\boldsymbol{\tau}_x$, on the other hand, are rather encoded as a linear combination of basis functions. The torques for one DOF are given by

$$\tau_x(s) = \frac{\sum_{i=1}^{L_d} w_{\tau i}\psi_i(s)}{\sum_{i=1}^{L_d} \psi_i(s)}. \tag{12}$$

Note here that the phase variable s is common across the CMP, i.e., across all DOFs of the TP and the DMP. As with DMPs,

regression techniques are used in order to compute the torque primitive $\tau_x(s)$ by solving a simplified equation system:

$$f_{ij} = \tau_{xj}(t_i),\ i = 1, \ldots, T. \tag{13}$$

By encoding motion trajectories as DMPs and corresponding torque signals as TPs, we obtain a set of NM example CMPs

$$\boldsymbol{H}_x^{\mathrm{CMP}} = \{\boldsymbol{w}_{q\,k}, \boldsymbol{g}_{qk}, \boldsymbol{w}_{\tau k}, \upsilon_k, \boldsymbol{c}_k\},\ k = 1, \ldots, NM, \tag{14}$$

represented by DMP weights $\boldsymbol{w}_{q\,k}$ and goals $\boldsymbol{g}_{q,k}$, TP weights $\boldsymbol{w}_{\tau k}$, common durations of DMP and TP υ_k and query points $\boldsymbol{c}_k$.

C. Executing CMPs

In order to execute a movement in a compliant manner, we could use a standard feedback controller (5) with low $\mathbf{K}$ values. However, the tracking errors would raise drastically. That can be mitigated by adding a precise inverse dynamic model of the robot (7). Mathematical modeling can be a demanding and time consuming task, that needs to be performed by an expert. Furthermore, the dynamic task, for which a model can also be needed, can change often, especially in unstructured home environments.

As an alternative we propose using a controller given by

$$\boldsymbol{\tau}_u = \mathbf{K}(\boldsymbol{q}_d - \boldsymbol{q}) + \mathbf{D}(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_f, \tag{15}$$

or in case of a known dynamical model of a robot,

$$\boldsymbol{\tau}_u = \mathbf{K}(\boldsymbol{q}_d - \boldsymbol{q}) + \mathbf{D}(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_f + \boldsymbol{f}_{dynamic}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}). \tag{16}$$

In (15) and (16) $\boldsymbol{\tau}_f$ denotes the additional feedforward torque signal, which compensates for the task-specific dynamics and/or robot's flawed or non-existing dynamical model. To execute a CMP, we use the motion trajectory as desired trajectory and the learned corresponding torque as a feed forward torque.

III. Statistical Generalization of CMPs

Learning CMPs through human demonstration can simplify compliant execution of dynamically versatile tasks. But this cannot be done for each possible variation of the task. For each new task descriptor, a new CMP needs to be learned, i.e., motion trajectory needs to be learned through human demonstrations and executed on a robot for torque learning. If the movement deviates just on the torque level, e.g. because of different payload, then only supervised learning of the torque is needed. This new learning can however be avoided by using statistical generalization techniques. For that a set of learned CMPs which transition smoothly between each other as a function of query points (i.e. task descriptors) is needed. Generalization allows executing tasks at an arbitrary query point $\boldsymbol{c}$ within the learned query space.

For statistical generalization we use the Gaussian process regression (GPR), which can be used to learn a function

$$\boldsymbol{F}_{\boldsymbol{H}_x^{\mathrm{CMP}}} : \boldsymbol{c} \longmapsto [\boldsymbol{w}_q, \boldsymbol{g}_q, \boldsymbol{w}_\tau, \upsilon]. \tag{17}$$

A CMP, defined by $\boldsymbol{w}_q$, $\boldsymbol{g}_q$, $\boldsymbol{w}_\tau$, and υ, can be used to execute a task, defined by a query $\boldsymbol{c}$, in a compliant manner. By the above definition, $\boldsymbol{F}_{\boldsymbol{H}_x^{\mathrm{CMP}}}(\boldsymbol{c})$ computes an appropriate CMP parameters at the given query $\boldsymbol{c}$ i. e. task variation.

A Gaussian process describes a distribution over functions [29]. For each component function f of the vector valued function $\boldsymbol{F}_{\boldsymbol{H}_x^{\mathrm{CMP}}}(\boldsymbol{c})$, Gaussian process regression can be used to estimate f from data such as the one given in (14). In general, Gaussian process is defined as

$$f(\boldsymbol{c}) \sim \mathcal{GP}(m(\boldsymbol{c}), \mathrm{cov}(\boldsymbol{c}, \boldsymbol{c}')), \tag{18}$$

where $m(\boldsymbol{c}) = \mathbb{E}(f(\boldsymbol{c}))$ is its mean function and $\mathrm{cov}(\boldsymbol{c}, \boldsymbol{c}') = \mathbb{E}[(f(\boldsymbol{c}) - m(\boldsymbol{c}))(f(\boldsymbol{c}') - m(\boldsymbol{c}'))]$ its covariance function.

Lets denote example outputs of one component of the CMP as $g_k = f(\boldsymbol{c}_k) + \epsilon$, where ϵ is normally distributed white noise, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, which denotes the measurement noise. Lets assume now that besides the original data we obtain a new set of inputs $\boldsymbol{c}_1^*, \ldots, \boldsymbol{c}_k^*$ and outputs $\boldsymbol{g}^* = [g_1^{*T}, \ldots, g_K^{*T}]^T$. If the mean of the process is assumed zero, the joint distribution of example outputs $\boldsymbol{g} = [g_1^T, \ldots, g_{NM}^T]^T$ and new outputs $\boldsymbol{g}^*$ is given by

$$\begin{bmatrix} \boldsymbol{g} \\ \boldsymbol{g}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \boldsymbol{\Sigma}(\mathbf{C}, \mathbf{C}) + \sigma_n^2 \mathbf{I} & \boldsymbol{\Sigma}(\mathbf{C}, \mathbf{C}^*) \\ \boldsymbol{\Sigma}(\mathbf{C}^*, \mathbf{C}) & \boldsymbol{\Sigma}(\mathbf{C}^*, \mathbf{C}^*) \end{bmatrix}\right), \tag{19}$$

where $\mathbf{C}$ denotes the matrix of example inputs, $\mathbf{C}^*$ matrix of the new inputs and $\boldsymbol{\Sigma}$ is obtained by pairwise evaluation of the covariance function cov. In our experiments we used the squared exponential covariance function

$$\mathrm{cov}(\boldsymbol{c}, \boldsymbol{c}') = \sigma_f^2 \exp\left(-\frac{\|\boldsymbol{c} - \boldsymbol{c}'\|^2}{2l^2}\right), \tag{20}$$

where parameter σ_f defines the variance of the signal $\{g_k\}$ and l is the characteristic length-scale, i.e. roughly the distance that one has to move in the input space before the value of the function changes significantly [29].

Thus given new queries $\boldsymbol{c}^*$, using (19) we can estimate the mean of output values $\boldsymbol{g}^*$ as follows [29]

$$\bar{\boldsymbol{g}}^* = \boldsymbol{\Sigma}(\mathbf{C}^*, \mathbf{C})[\boldsymbol{\Sigma}(\mathbf{C}, \mathbf{C}) + \sigma_n^2 \mathbf{I}]^{-1} \boldsymbol{g}. \tag{21}$$

As stated before, the Gaussian process is fully defined by mean and covariance functions. While assuming the mean to be zero, we still need to define the covariance function.

Equation (20) introduces new parameters that need to be determined. These free parameters, called hyperparameters,

$$\boldsymbol{\theta} = \{\sigma_f, l, \sigma_n\}. \tag{22}$$

affect the prediction of the Gaussian process. Ideally, hyperparameters are computed automatically. This can be accomplished by maximizing the log marginal likelihood

$$L = \log p(\boldsymbol{g}|\mathbf{C}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{g}^T[\boldsymbol{\Sigma}(\mathbf{C}, \mathbf{C}) + \sigma_n^2 \mathbf{I}]^{-1}\boldsymbol{g} - \frac{1}{2}\log\left(\det\left[\boldsymbol{\Sigma}(\mathbf{C}, \mathbf{C}) + \sigma_n^2 \mathbf{I}\right]\right) - \frac{n}{2}\log 2\pi, \tag{23}$$

which can be done using any of the standard nonlinear optimization routines.

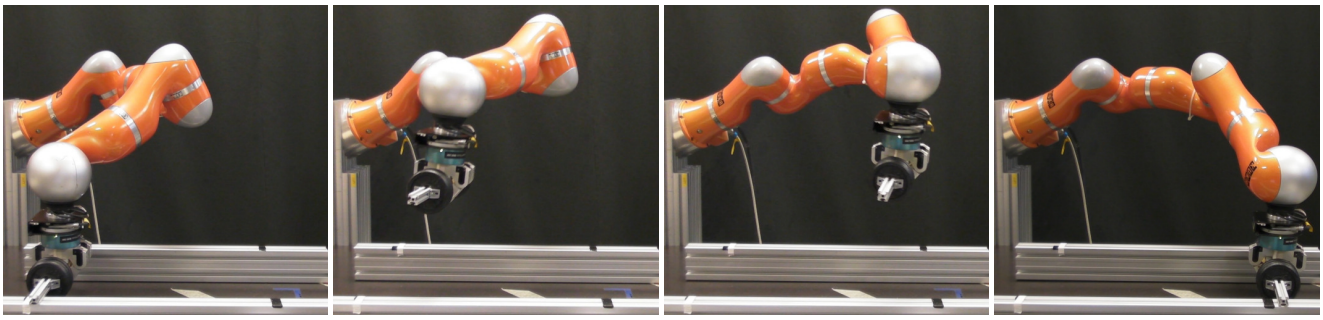

Fig. 1. Experimental setup and example task execution for stiffness evaluation. The robot picks up the hand-weight on the left, carries it to the right over a demonstrated position trajectory and releases it.

Optimizing the hyperparameters $\boldsymbol{\theta}$ and calculating the inverse matrix $[\boldsymbol{\Sigma}(\mathbf{C},\mathbf{C})+\sigma_n^2\mathbf{I}]^{-1}$ (21) is the most computationally expensive part of Gaussian process regression. But all these calculations can be done offline as they depend only on the training data $\boldsymbol{H}_x^{\mathrm{CMP}}$.

Once the GP is *trained* using example training sets of CMPs, new appropriate CMPs for given queries $\boldsymbol{c}$ can be calculated by simple matrix multiplications, which can easily be accomplished in real time.

IV. Evaluation

To evaluate discrete CMPs and their generalization we used a Kuka LWR 4 arm with a mounted BarrettHand BH8-280. While our approach does not need the robot's dynamical model, Kuka's own dynamical model was used in all experiments presented in this paper, see (7, 8, 16). CMP compensated for any model's errors and for task dynamics. The experimental setup can be seen in Fig. 1 and Fig. 5. All 7 of robot's DOF were used throughout the experiments. The first part of this chapter focuses on CMP evaluation, more specifically, on tracking errors under various stiffness settings while using classical feedback control and the proposed CMPs. While the second part briefly looks at collision evaluation, the third part tackles generalized CMPs. It compares learned CMPs to CMPs generalized using a one dimensional query. It also evaluates trajectory tracking of CMPs generalized over a two dimensional query, effecting both parts of the CMP.

A. Learned CMP evaluation

The effects of the stiffness parameter on the tracking error was analyzed while executing a pick and place task using a classic feed-back controller and proposed CMPs. The experimental setup can be seen in Fig. 1, presenting an example pick and place movement in a series of snapshots. A trajectory $\boldsymbol{q}_s(t)$, moving a hand weight from the initial position to the final position, was demonstrated by kinesthetic guiding. It was then executed five times using a high-gain feedback controller (7), which ensures high tracking accuracy. For each repetition, the mass of the object was changed

$$\boldsymbol{c}_s = \{m_1, m_2, m_3, m_4, m_5\} = \{0.5, 1.5, 2.5, 3.5, 4.5\}\ \text{[kg]}, \tag{24}$$

and different corresponding torques were obtained. The motion trajectories and corresponding torques were encoded as CMPs

$$\boldsymbol{H}_s^{CMP} = \{\boldsymbol{w}_{q_i}, \boldsymbol{g}_{qi}, \boldsymbol{w}_{\tau i}, \upsilon_i, \boldsymbol{c}_i\},\ i = 1,2,3,4,5. \tag{25}$$

For evaluation, demonstrated movement was first executed several times while using classical feed-back control (7). For each object mass several executions were done under varying stiffness settings. The movement was then executed as a CMP, i.e., using feedforward torques (16), which were encoded in the CMPs. Again, several executions for each object mass were done under different stiffness settings. For comparison, the maximum error for each task execution is defined as

$$e_m = \max_t(||\boldsymbol{p}_a(t) - \boldsymbol{p}_d(t)||), \tag{26}$$

where $\boldsymbol{p}_a(t)$ is the measured robot position on the trajectory and $\boldsymbol{p}_d(t)$ is the desired position on the trajectory, both given in Cartesian space. Eight different stiffness settings were used

$$k_s = \{10, 25, 50, 125, 250, 500, 1000, 2000\}\ \text{[Nm/rad]}, \tag{27}$$

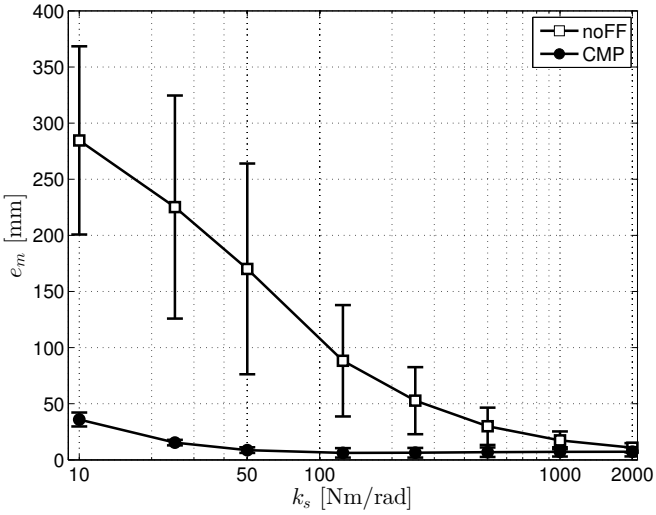

Fig. 2. Mean and standard deviation of task's maximum error e_m over the object's mass. The bottom line shows the error's mean and standard deviation for the proposed system with task-specific feedforward torques (CMP), while the top one shows the mean and standard deviation for the feedback control without feedforward torque signal (noFF).

TABLE I
MAXIMUM ERRORS FOR TASK EXECUTIONS USING TWO DIFFERENT CONTROL APPROACHES UNDER VARYING STIFFNESS SETTINGS. ALL ERROR VALUES ARE IN MILLIMETERS.

		Stiffness setting k_s [Nm/rad]							
		10	25	50	125	250	500	1000	2000
no *FF*	m_1	138*	66.4	41.4	21.7	13.8	10.3	10.9	11.2
	m_2	295*	188*	123*	59	34.3	17.9	9.57	5.24
	m_3	313*	285*	171*	94.5*	52.6	29.2	15.6	8.62
	m_4	335*	297*	232*	117*	75.7*	41.7	22.8	13
	m_5	342*	290*	283*	149*	87.2*	50.8	28.2	16
	mean	285 (83.8)	225 (99.4)	170 (93.8)	88.3 (49.6)	52.7 (29.9)	30 (16.6)	17.4 (7.94)	10.8 (4.11)
CMP	m_1	41.9	16	9.1	9.14	10.5	11.1	11.5	11.6
	m_2	37.7	15.9	7.76	2.61	2.28	2.82	3.14	3.35
	m_3	31.9	13.2	7.22	3.53	2.85	3.01	3.47	3.48
	m_4	40.9	13	6.52	3.73	4.8	5.7	6.1	6.19
	m_5	27.4	19	12.9	12.2	11.3	11.5	11.4	11.5
	mean	36 (6.18)	15.4 (2.45)	8.71 (2.54)	6.24 (4.19)	6.35 (4.28)	6.83 (4.24)	7.12 (4.12)	7.22 (4.1)

*Due to a high tracking error the execution of the pick and place task was unsuccessful.

were k_s are the diagonal elements of the stiffness matrix $\mathbf{K} = k_s\mathbf{I}$, see (16) and (7). These values were selected in order to cover a wide spectrum of compliance exhibited by the Kuka LWR robot.

Table I shows the mean and standard deviation of e_m over all object weights $\boldsymbol{c}_s$ for each stiffness setting k_s. The top row presents values for the feedback controller without feedforward term, while the bottom row presents values gained by using CMPs, i.e. with the learned feedforward toques. Cases when the robot was not able to successfully accomplish the task, i.e., picking the object and placing it on the goal, are marked with a star (*). The same results are presented in Fig. 2 with mean values over the object's mass. We can see that the tracking error is much larger (with a statically significant difference) if feedforward torques are not used compared to the proposed approach. We can also detect the point where the errors start to increase notably, i.e. at stiffness values lower than 50, $k_s < 50$ Nm/rad. Based of these results, the stiffness value in the final experiment was set to $k_s = 50$ Nm/rad.

B. Collision evaluation

In the next experiment we evaluated CMPs while unexpectedly colliding with an object. A simple downward motion was demonstrated and executed in order to train the CMP. While the movement was re-executed an obstacle was put in its path. Three different approaches were used:1) standard control with high gains, 2) standard control with low gains, and 3) a previously learned CMP with low stiffness values. TCP tracking errors $(\mathbf{p}_a(t) - \mathbf{p}_d(t))$, measured tool center point forces and positions along the z axis are showed in Fig. 3. We can observe that while using a standard approach with high gains the tracking error remains small throughout the movement, but as the robot exhibits stiff behavior forces raise significantly after the contact with the obstacle. If the feedback gains are lowered, the robot moves in a compliant manner and the forces drop. But the tracking accuracy is diminished. The CMP combines positive aspects of both standard approaches. While the tracking accuracy remains high before the contact, the forces are low after the contact.

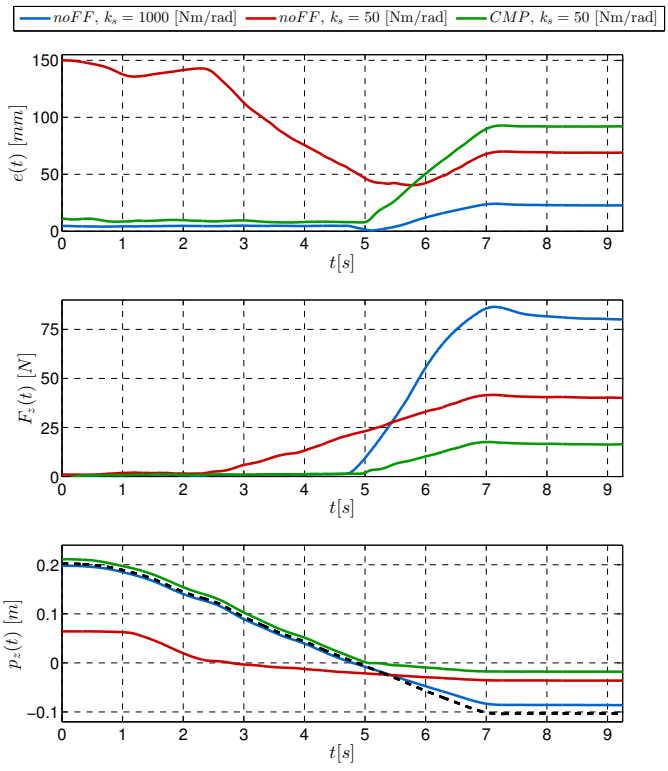

Fig. 3. Robot colliding with an object while using different stiffness settings and control approaches. The graphs present collision trajectories and forces under two different stiffness settings ($k_s = 1000$ Nm/rad and $k_s = 50$ Nm/rad) and with two different control approaches ($noFF$ and CMP). Blue trajectories show the performance of standard control with high gains, red trajectories the performance of standard low-gain control, and green trajectories the performance of CMPs. The top graph shows position errors, while the second one shows TCP forces in the z axis. The bottom one shows robot's actual task space position in the significant dimension, i .e., the vertical z axis. In the bottom graph the desired trajectory is denoted by a dashed line, while the obstacle starts at 0 m.

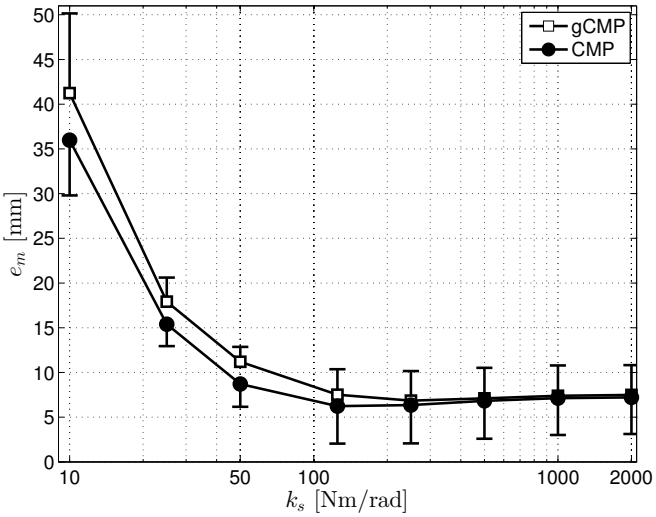

Fig. 4. Mean and standard deviation of task's maximum error e_m over the object's mass. The bottom line shows the error's mean and standard deviation while executing learned CMPs. The top line denotes executions of generalized CMPs (gCMP).

C. Generalized CMP evaluation

For the purpose of evaluating generalized CMPs, they were first compared to learned, non-generalized, CMPs. The experimental setup and task remained identical to the previous experiment (Fig.1). The same set of learned CMPs was also used (25). They were generalized (see Sec. III) over a one dimensional query, i.e., a varying object mass. Generalized CMPs could compliantly move an object with arbitrary mass within the training space. New, generalized, CMPs were executed for 9 different queries, covering demonstrated weights as well as points in between. For each query, the task was executed for 8 different stiffness settings. Fig. 4 compares the maximum tracking errors e_m of generalized CMPs to maximum tracking errors gained by executing learned CMPs, in the previous subsection. We can observe a slight, but not statistically significant, increase in tracking error at lower stiffness values, when the system is more susceptible to inaccuracies in generalized torque signals.

The final part of the evaluation focused on CMPs generalized over a two dimensional query and tracking errors w.r.t query points. The pick and place experimental setup can be seen in Fig. 5. Throughout this experiment a stiffness setting of $k_s = 50$ Nm/rad was used. In addition to varying object mass, the final configuration was also changing. The task was to move the hand weight to different final positions. The queries were defined as

$$\boldsymbol{c}_d = [c_g, c_m], \tag{28}$$

where c_g denotes the final, i.e., goal position varying in height and c_m denotes the varying mass of the object. First, the motion trajectories were obtained by kinesthetically guiding the robot for each example query c_g. Six example goal positions were used, varying for approximately 45 mm in height. Each of the six example motion trajectories,

$$\boldsymbol{Q}_x = \{\{\boldsymbol{q}_{x1}(t), c_{g1}\},\ \{\boldsymbol{q}_{x2}(t), c_{g2}\}, \ldots,\ \{\boldsymbol{q}_{x6}(t), c_{g6}\}\}, \tag{29}$$

was tracked 5 times with a robot using high-gain feedback controller at varying mass queries c_m. The object mass varied by 1 kg and covered the range from 0.5 kg to 4.5 kg. Altogether, 30 example pairs of motion trajectories and corresponding torques $\{\boldsymbol{q}(t)_x, \tau(t)_x\}$ were obtained, covering all combinations of query points $\{c_g, c_m\}$. By encoding them as CMPs, a set of 30 example CMPs was obtained

$$\boldsymbol{H}_x^{CMP} = \{\boldsymbol{w}_{q_i}, \boldsymbol{g}_{qi}, \boldsymbol{w}_{\tau i}, \upsilon_i, \boldsymbol{c}_i\},\ i = 1, \ldots, 30. \tag{30}$$

Example CMPs were employed as described in Section III to calculate new CMPs. The generalized CMPs can compliantly move the object of arbitrary mass to an arbitrary goal height within the training space defined by example query points.

In order to evaluate the generalization of CMPs over a two dimensional query, 99 compliant pick-and-place movements to different positions and with varying masses were executed. These new CMPs covered the whole training space, including 30 training and 66 new query points. The goal height varied by 22.5 mm, while the object mass was changed in steps of 0.5 kg.

For each task execution, maximum error e_m was calculated using (26). Table II shows the maximum errors for each generalized CMP. Results are also shown in Fig. 6. Note that the tracking error is slightly larger as the mass of the object increases. This is to be expected, as the system is more sensitive to the torque error contributed by inaccurate generalization when the hand weight is heavy (the actual task dynamics differ most from the Kuka's model own dynamical model).

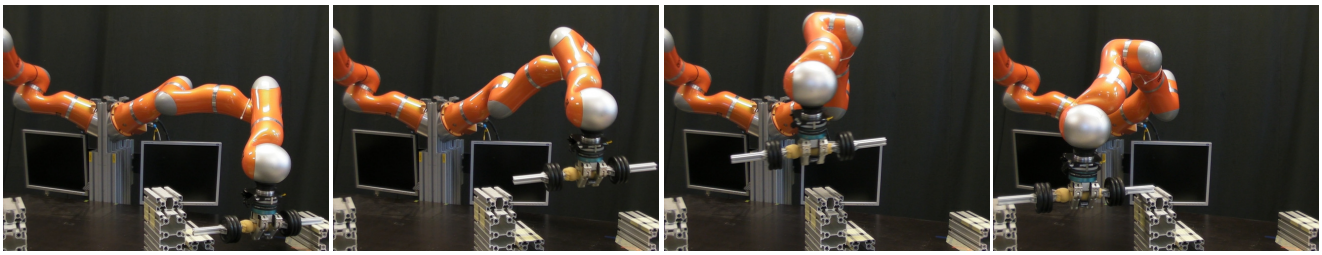

Fig. 5. Experimental setup for two dimensional query generalization evaluation. The robot picks up the hand-weight on the right, carries it to the left and releases it. The task varies in the object's mass and the height of the end position.

TABLE II
MAXIMUM ERRORS FOR GENERALIZED CMPS UNDER VARYING QUERIES.
ALL ERROR VALUES ARE IN MILLIMETERS.

		Final position height c_q [mm]										
		0	22.5	44.7	67.2	89.2	111.7	133.7	156.2	179.2	201.7	224.4
Object mass c_m [kg]	0.5	13.4	15.9	13.1	20.3	15.2	15.5	15.3	16.1	10.4	10.8	10.5
	1	14.1	16.7	14.2	18.7	13.4	13.4	15.2	17.3	15.6	16.9	14.7
	1.5	16.9	17.5	16.6	18.9	14.1	16.8	15.3	17	16.2	19.2	17.5
	2	15.5	16.3	14.8	17.6	13.6	13.5	12.6	13.4	12.6	18.7	12.1
	2.5	14.3	15.1	15.2	18.6	12.5	13.9	12.8	16.5	14.7	21.2	18
	3	16.9	16.2	16.7	23.9	14.9	17.4	13.9	23.2	22.2	28.6	19.6
	3.5	18.8	22.7	22.2	21.2	19.3	21.9	21.2	19.1	20.4	21.9	20.4
	4	22.3	27.2	24.4	20.6	22	23.1	20.9	19.7	20.1	20	24
	4.5	26.7	30.8	31.5	24.3	23.9	23.3	20.2	22.3	18.9	25.7	25.2

We calculated the mean and standard deviation of all these 99 maximum tracking errors e_m

$$m_e^x = 18.2(4.4) \text{ mm}. \tag{31}$$

If these values are compared to errors presented in previous evaluation scenarios, we can note that the errors resulting from generalized CMPs do not rise significantly compared to errors arising from CMPs directly learned from one of the example trajectories (Table I) or generalized CMPs using a one dimensional query (Fig. 4).

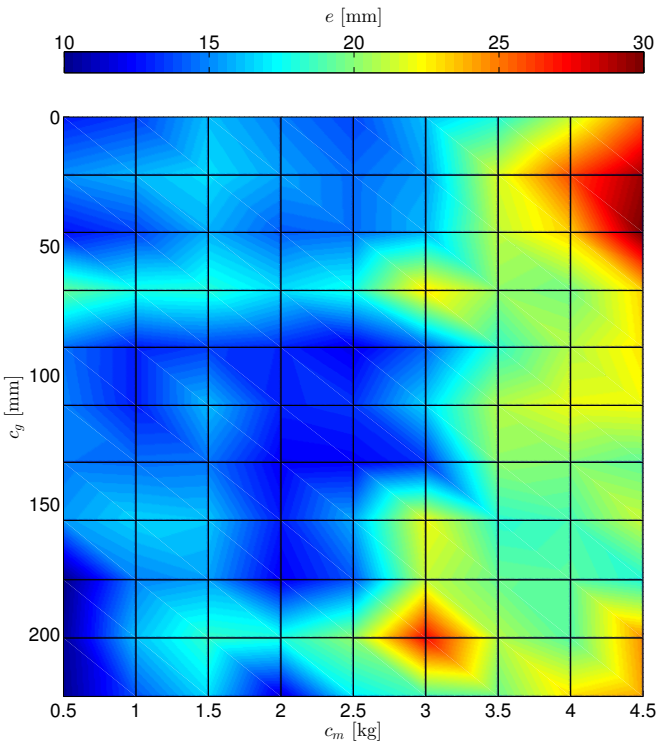

Fig. 6. Maximum errors e_m that resulted from executing generalized CMPs computed at different queries.

V. CONCLUSION

Proposed CMPs can be learned in a two step process: 1) position trajectory gained through programming by demostration and 2) corresponding torque signals gained by execution of position trajectories on a robot while using a high gain feedback control. The paper shows that CMPs can be executed by a proposed control approach in a compliant manner while maintaining low tracking errors. Evaluation on a discrete pick and place task without the use of dynamical models showed a significant decrease in tracking error over a wide range of compliance with respect to classic feedback control approaches. Collision evaluation showed that CMPs combine high tracking accuracy while executing a movement unperturbed and compliant behavior, i.e., low forces at unforeseen contacts. While no dynamical models are needed to gain a task specific CMP, it must be done for each task variation. This can be mitigated by using statistical generalization techniques in order to gain appropriate CMPs for previously unexplored task variations inside the training space. While evaluation of generalized position trajectories was already done [7], [8], this paper focuses on generalized CMPs and its effect on tracking errors. Evaluation showed no significant rise in error while executing generalized CMPs over one or two dimensional queries.

APPENDIX A
CONTROL FRAMEWORK

This appendix briefly presents the used control framework. Assuming that a robot consists of rigid bodies, the equations of motion can be written as

$$\mathbf{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\epsilon}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{\tau}, \tag{32}$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are the joint positions, velocities and accelerations, respectively, $\mathbf{H}(\boldsymbol{q})$ is the inertia matrix, $\mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are the Coriolis and centripetal forces, $\boldsymbol{g}(\boldsymbol{q})$ are the gravity forces and $\boldsymbol{\epsilon}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ are the nonlinearities which are not considered in the rigid body dynamics, e. g. friction. If the robot's inverse dynamic model given by (32) and denoted as $\boldsymbol{f}_{dynamic}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ is known, a possible control approach for

tracking the desired trajectory $(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d)$ is defined as

$$\boldsymbol{\tau}_u = \mathbf{K}(\boldsymbol{q}_d - \boldsymbol{q}) + \mathbf{D}(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{f}_{dynamic}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}). \quad (33)$$

where $\boldsymbol{\tau}_u$ is the commanded torque for joint specific impedance control, $\mathbf{D}$ is the damping matrix, and $\mathbf{K}$ is the diagonal matrix that determines the stiffness of the robot.

If the diagonal elements of $\mathbf{K}$ in (33) are high, then the robot behavior is stiff. This also implies better tracking accuracy of the desired joint trajectories $\boldsymbol{q}_d$. By lowering the values of the matrix $\mathbf{K}$, the robot becomes less stiff i. e. more compliant. However, with compliant behavior tracking errors can rise significantly, as the robot cannot accurately follow the desired trajectory.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 600578 (ACAT).

REFERENCES

[1] C. Breazeal and B. Scassellati, "Robots that imitate humans," *Trends Cogn. Sci.*, vol. 6, no. 11, pp. 481–487, 2002.

[2] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robot. Auton. Syst.*, vol. 47, no. 2, pp. 109–116, 2004.

[3] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends Cogn. Sci.*, vol. 3, no. 6, pp. 233–242, 1999.

[4] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Secaucus, NJ, USA: Springer, 2008, pp. 1371–1394.

[5] S. Schaal, P. Mohajerian, and A. Ijspeert, "Dynamics systems vs. optimal control–a unifying view," *Prog. Brain Res.*, vol. 165, pp. 425–445, 2007.

[6] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural Comput.*, vol. 25, no. 2, pp. 328–373, 2013.

[7] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 800–815, 2010.

[8] D. Forte, A. Gams, J. Morimoto, and A. Ude, "On-line motion synthesis and adaptation using a trajectory database," *Robot. Auton. Syst.*, vol. 60, no. 10, pp. 1327–1339, 2012.

[9] H. Ben Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, Hong Kong, China, 2014, pp. 2831–2837.

[10] A. Schmitz, P. Maiolino, M. Maggiali, L. Natale, G. Cannata, and G. Metta, "Methods and technologies for the implementation of large-scale robot tactile sensors," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 389–400, 2011.

[11] Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo, "Conformable and scalable tactile sensor skin for curved surfaces," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Orlando, FL, USA, 2006, pp. 1348–1353.

[12] M. Frigola, A. Casals, and J. Amat, "Human-robot interaction based on a sensitive bumper skin," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Beijing, China, 2006, pp. 283–287.

[13] A. Albu-Schaffer, O. Eiberger, M. Grebenstein, S. Haddadin, C. Ott, T. Wimbock, S. Wolf, and G. Hirzinger, "Soft robotics," *IEEE Robot. Automat. Mag.*, vol. 15, no. 3, pp. 20–30, 2008.

[14] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *J. Robotic Syst.*, vol. 20, no. 2, pp. 45–63, 2003.

[15] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Appl. Bionics Biomech.*, vol. 5, no. 3, pp. 99–117, 2008.

[16] R. V. Ham, T. G. Sugar, B. Vanderborght, K. W. Hollander, and D. Lefeber, "Compliant actuator designs," *IEEE Robot. Automat. Mag.*, vol. 16, no. 3, pp. 81–94, 2009.

[17] S. Wolf and G. Hirzinger, "A new variable stiffness design: Matching requirements of the next robot generation," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008, pp. 1741–1746.

[18] A. Gams, B. Nemec, A. Ijspeert, and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 816–830, 2014.

[19] T. Kulvicius, M. Biehl, M. J. Aein, M. Tamosiunaite, and F. Wörgötter, "Interaction learning for dynamic movement primitives used in cooperative robotic tasks," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1450–1459, 2013.

[20] R. Paul and B. Shimano, "Compliance and control," in *Proc. Conf. Joint Automatic Control*, West Lafayette, IN, USA, 1976, pp. 694–699.

[21] N. Hogan, "Stable execution of contact tasks using impedance control," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Raleigh, NC, USA, 1987, pp. 1047–1054.

[22] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. New York, NY, USA: Wiley, 2006, vol. 3.

[23] D. Nguyen-Tuong and J. Peters, "Learning robot dynamics for computed torque control using local gaussian processes regression," in *ECSIS Symp. Learning and Adaptive Behaviors for Robotic Systems*, Edinburgh, Scotland, UK, 2008, pp. 59–64.

[24] D. W. Franklin and D. M. Wolpert, "Computational mechanisms of sensorimotor control," *Neuron*, vol. 72, no. 3, pp. 425–442, 2011.

[25] N. S. Pollard, J. K. Hodgins, M. J. Riley, and C. G. Atkeson, "Adapting human motion for the control of a humanoid robot," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Washington, DC, USA, 2002, pp. 1390–1397.

[26] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Und.*, vol. 104, no. 2, pp. 90–126, 2006.

[27] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Commun. ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[28] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1463–1467, 2008.

[29] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. Cambridge, MA, USA: MIT Press, 2006.