This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

Learning Compliant Movement Primitives Through Demonstration and Statistical Generalization

Miha Deniša, Andrej Gams, *Member, IEEE*, Aleš Ude, *Member, IEEE*, and Tadej Petrič

***Abstract*—In this paper, we address the problem of simultaneously achieving low trajectory tracking errors and compliant control without using explicit mathematical models of task dynamics. To achieve this goal, we propose a new movement representation called compliant movement primitives (CMPs), which encodes position trajectory and associated torque profiles and can be learned from a single user demonstration. With the proposed control framework, the robot can remain compliant and consequently safe for humans sharing its workspace, even if high trajectory tracking accuracy is required. We developed a statistical learning approach that can use a database of existing CMPs and compute new ones, adapted for novel task variations. The proposed approach was evaluated on a Kuka LWR-4 robot performing 1) a discrete pick-and-place task with objects of varying weight and 2) a periodic handle turning operation. The evaluation of the discrete task showed a 15-fold decrease of the tracking error while exhibiting compliant behavior compared to the standard feedback control approach. It also indicated no significant rise in the tracking error while using generalized primitives computed by the statistical learning method. With respect to unforeseen collisions, the proposed approach resulted in a 75% drop of contact forces compared to standard feedback control. The periodic task demonstrated on-line use of the proposed approach to accomplish a task of handle turning.**

***Index Terms*—Adaptive system, intelligent robots, learning system, robot control.**

I. Introduction

PROGRAMMING by demonstration [1]–[5] is a widely used approach for acquiring new sensorimotor knowledge by observing humans performing a task. Example movements can be recorded using optical or magnetic marker-based systems [6], [7], vision systems including stereo cameras [8] and RGB-D cameras [9], [10]. A robot can also learn new tasks [11], force constrains [12], and compliance strategies [13] by being physically guided by a human. A movement captured through kinesthetic guiding has the advantage of already being adapted to the robot kinematics and dynamics. If the trajectory is captured in joint space, the imitation also preserves the posture of the robot when it is redundant with respect to the task.

While a single demonstration can be used to learn robot trajectories, for example encoded by dynamic movement primitives (DMPs) [14], [15], multiple demonstrations can be used as well in combination with statistical approaches. Statistical learning is used in order to synthesize an appropriate trajectory for a new task within the training space from a set of recorded movements [16]–[18]. Other representations of movement primitives that can be used with multiple demonstrations include hidden Markov models [19], Gaussian mixture models [20], [21], etc. A sample set of trajectories was also used by [22]. They used it to learn a skill manifold and with it generate trajectories modified to novel constrains.

An important aspect that needs to be considered when using robots in domestic settings or shared workspaces in industry settings is safety. If the *dynamic models* of the *robot* and the *task* are known, feedback gains can be adjusted to obtain the desirable compliance or even to prescribe the desired dynamic behavior of the mechanism [23]–[25]. However, in the case of imitation learning, dynamic models, which include both the dynamics of the robot and the dynamics of the task, are usually not known and can not be easily learned from imitation. Since modeling the task dynamics is usually a difficult and time consuming task, this paper addresses the problem of how to obtain them through autonomous learning and thereby avoid the need for an expert to define them.

A possible way to achieve compliance is through contact detection using an artificial tactile skin. Such skin can be realized as a bumper-based hard shell [26] or as a soft tactile sensing array [27]–[29]. Among the drawbacks of using the artificial skin are high prices, contacts with areas that cannot be covered, and the inherent sensor delay. Lately some work on contact detection was also done using alternative approaches, e. g., resonant frequency tracking [30].

Robots can be made passively compliant by design [31]. Contact forces can be mitigated by reducing the weight and hardness of the robot structure [32]–[34]. Passive compliance can be achieved through an advanced actuator design by implementing elastic elements [35]. Elastic elements can also adapt to a given task. They were extensively researched within the concept of variable stiffness actuators [36]–[39].

Manuscript received November 19, 2014; revised June 24, 2015 and October 6, 2015; accepted November 24, 2015. Recommended by Technical Editor J. Yu. This work was supported in part by the European Community's Seventh Framework Program FP7/2007-2013 (Specific Program Cooperation, Theme 3, Information and Communication Technologies) under Grant 600578, ACAT.

The authors are with the Humanoid and Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jozef Stefan Institute, Ljubljana 1000, Slovenia (e-mail: miha.denisa@ijs.si; andrej.gams@ijs.si; ales.ude@ijs.si; tadej.petric@ijs.si).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org provided by the authors. This material is 13 MB. Contact miha.denisa@ijs.si for further questions about this work.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2015.2510165

1083-4435 © 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

As an alternative to passive approaches, compliance can be achieved also through active torque control strategies. By comparing the actual torques and the required theoretical torques, compliant robot behavior can be achieved [40]–[42]. The main problem with this approach is that besides having access to actual torques through sensors, a correct dynamic model for each task variation must be available in order to calculate optimal torques for robot control. Models of the task dynamics are often not available. As an alternative, different biologically inspired methods were proposed for dynamic robot control. An extensive review of such control mechanisms, from optimal feedback control [43] to the forward models and predictive control [44], was published by Franklin and Wolpert [45].

The main result of this paper is a new concept of compliant movement primitives (CMPs), which is suitable for robots with active torque control. CMPs enable compliant and at the same time accurate execution of various tasks without requiring explicit models of task dynamics. They can be easily learned from user demonstrations. The inspiration for CMPs was derived from the human ability to learn arbitrary dynamical tasks [46]. To surpass the limited applicability of pure imitation, we extended the basic approach with *statistical generalization*, thereby allowing for variations in the task configuration. The viability of CMPs was shown in two distinct experiments, applied for discrete and periodic tasks. The discrete task, depicting pick-and-place, is common for numerous industrial settings. With our approach, it can be utilized in a shared human–robot environment and can also be applied for variable parts, as the robot's compliance can account for minor deviations. The periodic experiment depicts handle turning for a variable load and frequency. Such tasks are common in robot-aided assembly. The experiments were performed on the Kuka LWR-4 robot. Neither of the experiments required explicit dynamical models, which can only be provided by experts.

CMPs are gained by demonstration in a two-step learning process:

1) The desired motion trajectory is demonstrated and encoded as a DMP.
2) The trajectory is 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, denoted here as a Torque Primitive.

We denote the task-related position trajectory supplemented with the corresponding torques as a CMP. Both components of the CMP are used to execute a task. While the torque component is used as a feedforward term, the position trajectory is used to ensure stability 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 and in a compliant manner.

The second part of the paper tackles the learning of CMPs for variations of the desired task. In unconstrained domains such as home environments, people and consequently service robots often need to perform variations of the same task. Quick adaptability to different tasks and their variations is also becoming an important aspect in industrial settings, as many enterprises are customizing ever more products, leading to small batch size production. In our approach, 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. By combining CMPs with statistical learning methods, 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 complex movements and dynamical models, thus allowing for faster modification of robotic operations in assembly, making the process economically viable.

In related work, Nguyen-Tuong and Peters [47] used local Gaussian process regression for online dynamic model learning. This enabled them to improve the accuracy of the model and avoid high feedback gains, usually needed to ensure accurate trajectory tracking [48]. Unlike our approach, the method of Nguyen-Tuong and Peters learns complete robot dynamic models, hence, their approach requires the availability of much more data than our approach, which learns only the task-specific dynamics. With a similar goal in mind, iterative learning control (ILC) [49] was utilized for torque control. Schwarz and Behnke [50] used ILC to learn motor and friction models. Gautier *et al.* [51] proposed an iterative learning identification and control method for dynamic robot control.

The rest of the paper is structured as follows. The proposed control framework is presented in Section II. The next section defines CMPs and presents their learning. Section IV explains the generalization of CMPs using statistical methods. Evaluation is presented in Section V and the discussion is provided in the last section.

II. Control Framework

Assuming that a robot consists of rigid bodies, the equations of motion can be written as

$$\mathbf{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\epsilon}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{1}$$

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. We assume that the robot's inverse dynamic model given by (1) is known and denote it as $\boldsymbol{f}_{\mathrm{dynamic}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$. A possible control approach [31], [52] for tracking the desired position $\boldsymbol{q}_d$ using the inverse dynamic model is defined as

$$\boldsymbol{\tau}_u = \mathbf{K}(\boldsymbol{q}_d - \boldsymbol{q}) + \mathbf{D}(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{f}_{\mathrm{dynamic}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \tag{2}$$

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 $\mathbf{K}$ 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, while using compliant behavior with imprecise or/and incomplete inverse dynamic models tracking errors can rise significantly, as the tracking error reduction of the compliant controller is low. Therefore such a controller, referred to as

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

standard control approach from here on out, cannot compensate for any model imperfections.

To improve the tracking accuracy while maintaining compliant robot behavior, we propose the following controller:

$$\boldsymbol{\tau}_u = \mathbf{K}(\boldsymbol{q}_d - \boldsymbol{q}) + \mathbf{D}(\dot{\boldsymbol{q}}_d - \dot{\boldsymbol{q}}) + \boldsymbol{\tau}_f + \boldsymbol{f}_{\text{dynamic}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \tag{3}$$

where the $\boldsymbol{\tau}_f$ is an additional feedforward torque signal, which compensates for the task-specific dynamics.

The task-specific dynamics, i.e., the dynamics which are task dependent, and therefore, not included in the robot's dynamical model, can be obtained with mathematical modeling of the task [47], [48]. However, mathematical modeling is a difficult and time consuming problem that can only be performed by an expert. Therefore, we propose an alternative solution where the robot autonomously learns the proper task-specific dynamics $\boldsymbol{\tau}_f$ in a controlled environment or under human supervision. Once the task-specific dynamic is learned, the robot can accurately execute the task while being compliant. Compliant behavior implies lower impact forces in case of unexpected collisions. Therefore, the robot can now safely work in unstructured environments and interact with humans.

III. CMPs and Their Learning

We define a compliant movement as a combination of desired motion trajectories (joint positions) and corresponding joint torque signals

$$\boldsymbol{h}(t) = [\boldsymbol{q}_d(t), \boldsymbol{\tau}_f(t)] \tag{4}$$

where

$$\boldsymbol{q}_d(t) = [q_{d1}(t), wq_{d2}(t), \ldots, q_{dP}(t)]^T, \tag{5}$$

$$\boldsymbol{\tau}_f(t) = [\tau_{d1}(t), \tau_{d2}(t), \ldots, \tau_{dP}(t)]^T. \tag{6}$$

Here P denotes the number of degrees of freedom (DOF). In the proposed approach, motion trajectories $\boldsymbol{q}_d$ are first obtained by human demonstration and encoded as DMPs [14], [15]. The corresponding torques $\boldsymbol{\tau}_f$ are obtained by executing the learned task trajectories with a high-gain feedback controller, which provides information to achieve the required tracking accuracy. These task-specific torques are encoded as a linear combination of radial basis functions. We denote them as torque primitives (TPs). A pair of DMP and TP now describes a compliant movement. We denote such a pair as a CMP.

A. Learning Motion Trajectories

Various techniques can be applied to capture human demonstration. In this paper, we use kinesthetic guiding to capture example motion trajectories

$$\boldsymbol{q}_x(t) = [q_{x1}(t), q_{x2}(t), \ldots, q_{xP}(t)]^T \tag{7}$$

where subscript x denotes examples and P the number of DOF. The complete robot position trajectory is then given as

$$\tilde{\boldsymbol{q}}_x = \{\boldsymbol{q}_x(t_1), \boldsymbol{q}_x(t_2), \ldots, \boldsymbol{q}_x(t_T)\} \tag{8}$$

where T is the total number of captured samples throughout a single demonstration. With multiple demonstrations, a set of example motion trajectories in joint space is given as

$$\boldsymbol{Q}_x = \{\tilde{\boldsymbol{q}}_{xj}, \boldsymbol{c}_{qj}\}_{j=1}^{N} \tag{9}$$

where N denotes the total number of captured examples $\tilde{\boldsymbol{q}}_x$. Each example trajectory successfully accomplishes one variation of the task described through task parameters called queries $\boldsymbol{c}_q$. Task parameters can describe a variety of things, e.g., initial position of a discrete movement in task space that varies in a single dimension $c_q = x_{\text{start}}$, final position of the discrete movement in joints space, $\boldsymbol{c}_q = [q_1, q_2, \ldots, q_P]^T$, Cartesian mean position value of periodic motion in two dimensions, $\boldsymbol{c}_q = [p_x, p_y]^T$, etc. We encode each example trajectory $\tilde{\boldsymbol{q}}_x$ as a DMP [15], [16].

For the sake of completeness, we provide a short overview of the DMP theory [14], [15]. The equations below are valid for a single DOF, and can be used in parallel for multiple DOFs. A nonlinear system of differential equations defines a DMP for discrete (point-to-point) and periodic movements

$$\upsilon\dot{z} = \alpha_z(\beta_z(g - y) - z) + f(s) \tag{10}$$

$$\upsilon\dot{y} = z \tag{11}$$

where the linear part ensures the convergence of y to the desired goal configuration g once f becomes equal to zero. The nonlinear part $f(s)$ modifies the shape of the movement and is defined by a linear combination of L_d radial basis functions [53]

$$f(s) = \frac{\sum_{b=1}^{L_d} w_{qb}\psi_b(s)}{\sum_{b=1}^{L_d} \psi_b(s)} s. \tag{12}$$

Here ψ_b denotes Gaussian basis functions

$$\psi_b(s) = \exp(-d_b(s - c_b)^2) \tag{13}$$

with centers at c_b and widths $d_b > 0$. Note that $f(s)$ is not directly time dependent. Instead, a phase variable s with the initial value $s(0) = 1$ is used

$$\upsilon\dot{s} = -\alpha_s s. \tag{14}$$

The phase variable is useful in case of external perturbations and is common across all DOFs. In our case, the phase is common across the complete CMP, which in addition to a DMP also includes a Torque Primitive (TP).

By defining appropriate constants $\alpha_z, \beta_z, \upsilon > 0$, and $\alpha_s > 0$, the described system is guaranteed to converge to the desired configuration g [15]. If not stated otherwise we used $\alpha_z = 48$, $\beta_z = \alpha_z/4$, $\alpha_s = 2$ and $L_d = 30$. These DMP parameters were set empirically.

To acquire the target signal for learning, (10) and (11) are rewritten as a second-order system

$$\upsilon^2\ddot{y} + \alpha_z\upsilon\dot{y} - \alpha_z\beta_z(g - y) = f(s). \tag{15}$$

By substituting y with the example trajectory $\tilde{\boldsymbol{q}}_{xn}$ of the nth joint and its derivatives $\dot{\tilde{\boldsymbol{q}}}_{xn}$ and $\ddot{\tilde{\boldsymbol{q}}}_{xn}$, the target function is derived as

$$f_{ni} = \upsilon^2\ddot{q}_{xn}(t_i) + \alpha_z\dot{q}_{xn}(t_i) - \alpha_z\beta_z(g - q_{xn}(t_i)), \quad i = 1, \ldots, T \tag{16}$$

where the goal value g is specified by the end value of example trajectory $q_{xn}(t_T)$. The DMP is calculated by solving

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

the overdetermined equation system (16) using regression techniques [15], [16] for each joint trajectory.

Equations (12) and (13) hold for discrete movements. In case of periodic movements, a linear combination of Gaussian functions $f(s)$ is replaced by a linear combination of periodic basis functions [15] given by

$$f(\phi) = \frac{\sum_{b=1}^{L_p} w_{qb}\Gamma_b(\phi)}{\sum_{b=1}^{L_p}\Gamma_b(\phi)} r \tag{17}$$

$$\Gamma_b(\phi) = \exp(h_b(\cos(\phi - c_b) - 1)) \tag{18}$$

where r is the amplitude of the oscillator and $h_b > 0$. In our case, we used $L_p = 30$ and $h_b = 2.5\,L_p$. By replacing υ with $\upsilon = 1/\Omega$ in (10) and (11), we obtain

$$\dot{z} = \Omega(\alpha_z(\beta_z(g - y) - z) + f(\phi)) \tag{19}$$

$$\dot{y} = \Omega z. \tag{20}$$

Here, the discrete phase s was replaced by periodic phase ϕ, which is determined by the phase oscillator

$$\dot{\phi} = \Omega \tag{21}$$

where Ω is the frequency of oscillations.

Encoding of the periodic trajectories is done in a similar way as for discrete trajectories (15), (16), except that in (19), (21), frequency of the movement Ω is needed. Ideally, Ω is determined automatically from the data. Details on the automatic determination of frequency of periodic movements are provided in [54] and [55].

B. Learning Corresponding Torque Signals

In order to obtain the corresponding task-specific torque samples

$$\tilde{\boldsymbol{\tau}}_x = \{\boldsymbol{\tau}_x(t_1), \boldsymbol{\tau}_x(t_2), \ldots, \boldsymbol{\tau}_x(t_T)\} \tag{22}$$

$$\boldsymbol{\tau}_x(t) = [\tau_{x1}(t), \tau_{x2}(t), \ldots, \tau_{xP}(t)]^T \tag{23}$$

example movements $\tilde{\boldsymbol{q}}_x$, encoded as DMPs, are executed in a controlled environment or under human supervision. For that a standard control approach, i.e., a high-gain feedback controller

$$\boldsymbol{\tau}_u = \mathbf{K}(\boldsymbol{q}_x - \boldsymbol{q}) + \mathbf{D}(\dot{\boldsymbol{q}}_x - \dot{\boldsymbol{q}}) + \boldsymbol{f}_{\text{dynamic}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \tag{24}$$

is used. The needed velocities $\dot{\boldsymbol{q}}_x$ are also provided by the DMP. It should be noted here that in this step high values of $\mathbf{K}$ are needed in order to ensure the required tracking accuracy. We also assume the robot is capable of accurately tracking the desired trajectory. At this stage, the task specific dynamical model is being *learned* by recording torques provided by the controller. If the values $\mathbf{K}$ are low and the dynamical model is imprecise and/or does not incorporate current task dynamics, the controller will not be able to track the desired trajectory $\boldsymbol{q}_x$ and faulty torques will be recorded. Task-specific torques $\boldsymbol{\tau}_x$ are gained by subtracting the known robot's dynamics $\boldsymbol{f}_{\text{dynamic}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ from actual measured torques $\boldsymbol{\tau}_m$ at robot's joints

$$\boldsymbol{\tau}_x = \boldsymbol{\tau}_m - \boldsymbol{f}_{\text{dynamic}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}). \tag{25}$$

Here, we assume that the robot's dynamic model is known and the corresponding torques are used to mitigate its errors and compensate for task-specific dynamics. If the robot's model is not known, the controller given by (3) changes to

$$\boldsymbol{\tau}_u = \mathbf{K}(\boldsymbol{q}_x - \boldsymbol{q}) + \mathbf{D}(\dot{\boldsymbol{q}}_x - \dot{\boldsymbol{q}}) \tag{26}$$

and the torques for learning would be defined as

$$\boldsymbol{\tau}_x = \boldsymbol{\tau}_m \tag{27}$$

which would now compensate for the task-specific dynamics and the robot's own dynamics.

Each demonstrated motion $\tilde{\boldsymbol{q}}_x$ can be executed multiple times under different task conditions $\boldsymbol{c}_\tau$ to gain compliant movement trajectories, which have different corresponding task-specific torques, but the same position trajectory. For example, a position trajectory of the movement can be executed at varying velocities and produce compliant movements with different corresponding torques. Another example is moving objects of varying mass $c_\tau = m$ over the same position trajectory, i.e., the corresponding torques differ for each different mass. Thus, we need to obtain a set of example torque signals

$$\boldsymbol{T}_x = \{\tilde{\boldsymbol{\tau}}_{xk}, \boldsymbol{c}_{\tau k}\}_{k=1}^{NM} \tag{28}$$

where M is the number of times each of the N example motion trajectory $\tilde{\boldsymbol{q}}_{xj}$ was executed with varying conditions $\boldsymbol{c}_\tau$.

A linear combination of basis functions is used to encode task-specific torques $\tilde{\boldsymbol{\tau}}_x$ as a TP. For discrete movement, the torques for one DOF are given by

$$\tau_x(s) = \begin{cases} \dfrac{\sum_{b=1}^{L_d} w_{\tau b}\psi_b(s)}{\sum_{b=1}^{L_d}\psi_b(s)} & s \geq s_\epsilon \\ \tau_x(s_\epsilon) & s < s_\epsilon \end{cases} \tag{29}$$

where s_ϵ denotes the final value of the phase variable for the encoded learned torque signal. In this way, the final torque value is maintained, even if CMPs are executed beyond the final learned point. For periodic movements, the torques are given by

$$\tau_x(\phi) = \frac{\sum_{b=1}^{L_p} w_{\tau b}\Gamma_b(\phi)}{\sum_{b=1}^{L_p}\Gamma_b(\phi)}. \tag{30}$$

As with (16), regression techniques are used in order to compute the TPs $\tau_x(s)$ and $\tau_x(\phi)$ by solving a simplified equation system:

$$f_{ni} = \tau_{xn}(t_i),\ i = 1, \ldots, T. \tag{31}$$

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

$$\begin{aligned} \boldsymbol{H}_x = \ & \{\{\tilde{\boldsymbol{q}}_{x1}, \tilde{\boldsymbol{\tau}}_{x1}\}, \ldots, \{\tilde{\boldsymbol{q}}_{x1}, \tilde{\boldsymbol{\tau}}_{xM}\}, \\ & \{\tilde{\boldsymbol{q}}_{x2}, \tilde{\boldsymbol{\tau}}_{x(M+1)}\}, \ldots, \{\tilde{\boldsymbol{q}}_{x2}, \tilde{\boldsymbol{\tau}}_{x(2M)}\}, \ldots, \\ & \{\tilde{\boldsymbol{q}}_{xN}, \tilde{\boldsymbol{\tau}}_{x((N-1)M+1)}\}, \ldots, \{\tilde{\boldsymbol{q}}_{xN}, \tilde{\boldsymbol{\tau}}_{x(NM)}\}\} \end{aligned} \tag{32}$$

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

$$\begin{aligned} \boldsymbol{C}_x = \ & \{[\boldsymbol{c}_{q1}^T, \boldsymbol{c}_{\tau 1}^T]^T, \ldots, [\boldsymbol{c}_{q1}^T, \boldsymbol{c}_{\tau M}^T]^T, \\ & [\boldsymbol{c}_{q2}^T, \boldsymbol{c}_{\tau M+1}^T]^T, \ldots, [\boldsymbol{c}_{q2}^T, \boldsymbol{c}_{\tau 2M}^T]^T, \ldots, \\ & [\boldsymbol{c}_{qN}^T, \boldsymbol{c}_{\tau (N-1)M+1}^T]^T, \ldots, [\boldsymbol{c}_{qj}^T, \boldsymbol{c}_{\tau NM}^T]^T\}. \end{aligned} \tag{33}$$

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

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

$$\boldsymbol{H}_x^{\mathrm{CMP}} = \{\boldsymbol{w}_{q\,k}, \boldsymbol{g}_{q\,k}, \boldsymbol{w}_{\tau\,k}, \upsilon_k, \boldsymbol{c}_k\},\ k = 1, \ldots, NM \quad (34)$$

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 $\boldsymbol{c}_k = [\boldsymbol{c}_{q\,j}^{\ T}, \boldsymbol{c}_{\tau\,k}^{\ T}]^T$.

IV. Statistical Generalization of CMPs

For most tasks, there is an infinite number of task variations, hence each possible task variation cannot be learned in advance. However, if the learned CMPs transition smoothly between each other as a function of query points $\boldsymbol{c}_k$, statistical learning techniques can be used to predict optimal movements for new query points, i.e., task descriptors $\boldsymbol{c}$. This allows for executing the task at an arbitrary configuration as long as the task descriptor remains within the training space. For example, statistically we can compute the necessary torques to compliantly turn a handle even for task configurations that were not directly demonstrated.

More formally, our goal is to learn a function

$$\boldsymbol{F}_{\boldsymbol{H}_x^{\mathrm{CMP}}} : \boldsymbol{c} \longmapsto [\boldsymbol{w}_q^{\ T}, \boldsymbol{g}_q^T, \boldsymbol{w}_\tau^{\ T}, \upsilon]^T \quad (35)$$

given the training data (34), with outputs being defined as $[\boldsymbol{w}_{q\,k}^{\ T}, \boldsymbol{g}_{q\,k}^T, \boldsymbol{w}_{\tau\,k}^{\ T}, \upsilon_k]^T$ and inputs as $\boldsymbol{c}_k$. By the above definition, $\boldsymbol{F}_{\boldsymbol{H}_x^{\mathrm{CMP}}}(\boldsymbol{c})$ computes the appropriate CMP parameters at the given task descriptor $\boldsymbol{c}$. For each component function of the vector valued function $\boldsymbol{F}_{\boldsymbol{H}_x^{\mathrm{CMP}}}(\boldsymbol{c})$, we employ Gaussian process regression [56] to estimate it from the training data.

Let r be one of the components of the output vectors $[\boldsymbol{w}_q^{\ T}, \boldsymbol{g}_q^T, \boldsymbol{w}_\tau^{\ T}, \upsilon]^T$. For this one component, training data (34) can be written as $\{r_k, \boldsymbol{c}_k\}_{k=1}^{NM}$, where r_k are now scalar valued outputs. Let $\boldsymbol{r} = [r_1, \ldots, r_{NM}]^T$ be the vector combining all the training outputs. Let's also assume that we obtain a new set of inputs $\boldsymbol{c}^* = [\boldsymbol{c}_1^{*T}, \ldots, \boldsymbol{c}_K^{*T}]^T$ for which the corresponding outputs $\boldsymbol{r}^* = [r_1^*, \ldots, r_K^*]^T$ should be computed. Assuming that the mean of training outputs $\{r_k\}$ is zero[1], Gaussian process regression can be applied to compute $\boldsymbol{r}^*$ as follows:

$$\boldsymbol{r}^* = \boldsymbol{\Sigma}\left(\boldsymbol{C}^*, \boldsymbol{C}\right) \cdot \left[\boldsymbol{\Sigma}\left(\boldsymbol{C}, \boldsymbol{C}\right) + \sigma_n^2 \mathbf{I}\right]^{-1} \boldsymbol{r}. \quad (36)$$

Here $\boldsymbol{C} = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{NM}\}, \boldsymbol{C}^* = \{\boldsymbol{c}_1^*, \ldots, \boldsymbol{c}_K^*\}$, σ_n is the noise variance of the output data and

$$\begin{aligned}\boldsymbol{\Sigma}(\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K\}, \{\boldsymbol{c}'_1, \ldots, \boldsymbol{c}'_{K'}\}) = \\ = \begin{bmatrix} \mathrm{cov}(\boldsymbol{c}_1, \boldsymbol{c}'_1) & \ldots & \mathrm{cov}(\boldsymbol{c}_1, \boldsymbol{c}'_{K'}) \\ \vdots & \ldots & \vdots \\ \mathrm{cov}(\boldsymbol{c}_K, \boldsymbol{c}'_1) & \ldots & \mathrm{cov}(\boldsymbol{c}_K, \boldsymbol{c}'_{K'}) \end{bmatrix}\end{aligned} \quad (37)$$

$$\mathrm{cov}(\boldsymbol{c}_i, \boldsymbol{c}'_j) = \sigma_f^2 \exp\left(-\frac{\|\boldsymbol{c}_i - \boldsymbol{c}'_j\|^2}{2l^2}\right) \quad (38)$$

where σ_f is the signal variance and l the characteristic length-scale, i.e., roughly the distance that one has to move in the input space before the value of the output signal changes significantly. σ_n, σ_f, and l are called hyperparameters and should be estimated from the training data. This can be accomplished by maximizing the following log marginal likelihood:

$$\begin{aligned}\log\left(p(\boldsymbol{r}|\boldsymbol{C}, \sigma_l, \sigma_f, l)\right) = -\frac{1}{2}\boldsymbol{r}^T\left[\boldsymbol{\Sigma}(\boldsymbol{C}, \boldsymbol{C}) + \sigma_n^2\mathbf{I}\right]^{-1}\boldsymbol{r} \\ -\frac{1}{2}\log\left(\det\left[\boldsymbol{\Sigma}(\boldsymbol{C}, \boldsymbol{C}) + \sigma_n^2\mathbf{I}\right]\right) - \frac{NM}{2}\log 2\pi. \end{aligned} \quad (39)$$

This optimization can be performed by any of the standard nonlinear optimization routines.

The most computationally expensive part of Gaussian process regression is the calculation of the hyperparameters $\{\sigma_n, \sigma_f, l\}$, which is done by minimizing (39), and the calculation of the inverse matrix $[\boldsymbol{\Sigma}(\mathbf{C}, \mathbf{C}) + \sigma_n^2\mathbf{I}]^{-1}$. However, these calculations can be done offline as they depend only on the training data $\boldsymbol{H}_x^{\mathrm{CMP}}$. Once the Gaussian processes have been *trained* for each component function of the training set (34), new CMPs for arbitrary queries $\boldsymbol{c}^*$ can be calculated by simple matrix multiplications. This can easily be done in real time.

A. Stability of the Generalized CMPs

The basic assumption of our approach is that the reproduction of the demonstrated movements with a high-gain feedback controller (24) is stable. For the execution of the generalized movements, the original gains are reduced to achieve compliance, which further diminishes the risk of the system becoming unstable. Note also that in the case of discrete movements, the feedforward term $\tau_x(s)$ from (29) tends to the value that maintains the robot at the final configuration at the end of the movement. Thus, the output of the controller combines (24) with low gains, and the final value of encoded $\tilde{\boldsymbol{\tau}}_x$, thereby ensuring that the robot is stable at the end of the movement.

Generalization of CMPs with (35) results in feedforward terms that are bounded and smoothly transition between the feedforward terms obtained by high-gain feedback control, which by our initial assumption all resulted in stable movements. It is therefore intuitive that the generalized CMPs that smoothly transition between them as the function of task descriptors are stable as well. It should be noted, however, that a rigorous proof would require us to make various assumptions about the task and admissible perturbations. Such an analysis is beyond the scope of this paper.

V. Evaluation

To evaluate our approach, we used a Kuka LWR-4 arm with a mounted BarrettHand BH8-280. Throughout the experiments, CMPs consisted of joint trajectories and torques for all seven of the robot's DOF. Although the robot's dynamical model is not strictly necessary for the proposed approach, we made use of the dynamical model $\boldsymbol{f}_{\mathrm{dynamic}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ as provided by Kuka. The motivation behind our approach is to enable the robot to first, learn those aspects of the robot's dynamical model that are not contained in $\boldsymbol{f}_{\mathrm{dynamic}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, and second, acquire task-specific dynamical models that are often difficult or even

[1] In general, the mean $\bar{r} = \frac{1}{NM}\sum_{k=1}^{NM} r_k$ is not equal to zero and should be subtracted from the training data. $\bar{r}$ should later be added to the estimated output values (36).

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

TABLE I
MAXIMUM ERRORS FOR TASK EXECUTIONS USING DIFFERENT CONTROL APPROACHES UNDER VARYING STIFFNESS SETTINGS

		Stiffness setting k_s [N · m/rad]							
		10	25	50	125	250	500	1000	2000
M_{robot}	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)
M_{task}	m_1	95.9	42.8	24.1	10.2	7.15	6.68	6.41	6.39
	m_2	93.5	44.3	25.7	11.1	7.43	6.88	6.55	6.37
	m_3	142	59.9	34.2	17.4	10.1	7.45	6.82	6.51
	m_4	101	49.5	29	16.6	10.3	7.15	6.57	6.4
	m_5	162	71.2	40.3	20.2	12.2	7.92	7	6.62
	mean	119 (31)	53.6 (11.9)	30.7 (6.64)	15.1 (4.3)	9.43 (2.11)	7.22 (0.49)	6.67 (0.235)	6.46 (0.103)
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. All error values are in millimeters.

impossible to specify. In this section, we show that the tracking error of CMPs and generalized CMPs is comparable to standard approaches that use mathematically defined dynamical models in those rare cases when such models can be provided. Three different approaches are compared: a standard control approach (2) using the robot's dynamical model (denoted by M_{robot}); the same control approach (2) using a dynamical model enhanced with a task-specific point mass model at the top of the robot (denoted by M_{task}); and the proposed control approach (3) using CMPs (denoted by CMP).

Evaluation can be divided into four parts: 1) Compliant movement evaluation was done at various stiffness settings by observing tracking errors while executing a simple pick and place task using different control approaches; 2) collision setup was used to compare robot behavior to unforeseen impacts while using different stiffness settings and control strategies; 3) discrete CMPs and their generalization over one- and two-dimensional queries was evaluated w.r.t. tracking errors on a pick and place scenario; and 4) Periodic CMPs and generalized CMPs were evaluated using a hard-to-model task of raising a height-adjustable table by rotating its handle.

A. Compliant Movement Evaluation

CMPs were evaluated by comparing the three approaches: M_{robot}, M_{task}, and CMP. Experimental setup can be seen in Fig. 1. By moving an object of varying weight, dynamics of the task change and the inverse dynamic model would need to be adapted. We avoided doing that mathematically by using the proposed approach.

A movement which resulted in the hand weight being moved from the initial position to the final position was demonstrated by kinesthetic guiding. The movement was then executed five times using feedback controller (24) with high gains, which ensures high tracking accuracy. For each repetition, the mass of

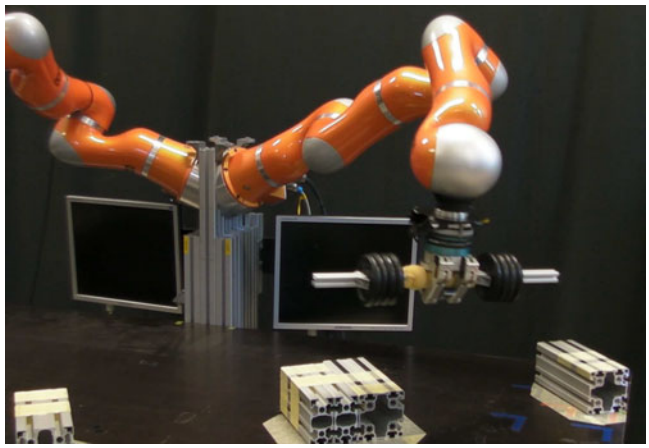

Fig. 1. Experimental setup for the discrete pick and place task. The robot picks up the hand-weight on the right, carries it to the left, and releases it.

the object was changed

$$c_s = \{m_1, m_2, m_3, m_4, m_5\} = \{0.5, 1.5, 2.5, 3.5, 4.5\} \text{ kg} \quad (40)$$

and different corresponding torques were obtained. As the Kuka robot has torque sensors in each joint, actual joint torques were recorded at this step. The motion trajectories and corresponding torques were encoded as CMPs

$$\boldsymbol{H}_s^{\text{CMP}} = \{\boldsymbol{w}_{q\,k}, \boldsymbol{g}_{q\,k}, \boldsymbol{w}_{\tau\,k}, v_k, \boldsymbol{c}_k\},\ k = 1, \ldots, 5. \quad (41)$$

All learned CMPs were executed while using eight different stiffness settings

$$k_s = \{10, 25, 50, 125, 250, 500, 1000, 2000\} \text{ Nm/rad} \quad (42)$$

where k_s are the diagonal elements of the stiffness matrix $\mathbf{K} = k_s\mathbf{I}$, which is used in (3) and (24). These values were selected in order to cover a wide specter of compliance exhibited by the

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

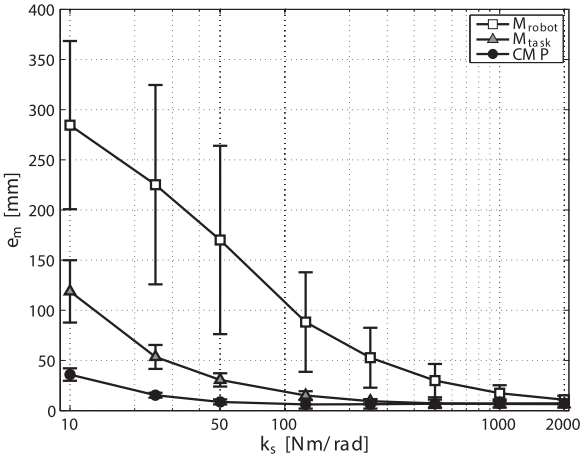

Fig. 2. Mean and standard deviation of task's maximum error e_m. The bottom line shows the mean and standard deviation for the proposed system with task-specific feedforward torques (CMP). The top line shows the mean and standard deviation for the standard control approach using Kuka's dynamical model ($\mathrm{M}_{\mathrm{robot}}$), while the middle line shows errors obtained while using a task specific model ($\mathrm{M}_{\mathrm{task}}$).

Kuka LWR-4 robot and are used commonly for all DOFs. The maximum error of each task execution was defined as

$$e_m = \max_t(||\boldsymbol{p}_a(t) - \boldsymbol{p}_d(t)||). \tag{43}$$

Here $\boldsymbol{p}_a(t)$ is the measured robot position on the trajectory and $\boldsymbol{p}_d(t)$ the desired position, both given in the Cartesian space. The task was then executed for each object mass and stiffness setting using all three control approaches ($\mathrm{M}_{\mathrm{robot}}$, $\mathrm{M}_{\mathrm{task}}$, CMP).

Table I shows the mean and standard deviation of e_m over all object weights c_s for each stiffness setting k_s. While the top two sections present results gained by using the standard control approach with two different models ($\mathrm{M}_{\mathrm{robot}}$ and $\mathrm{M}_{\mathrm{task}}$), the bottom section presents errors arising while executing CMPs. The same results are presented in Fig. 2. The tracking error is quite high when $\mathrm{M}_{\mathrm{robot}}$ is used. If M_{task} is used instead, errors drop significantly. We can also observe that errors drop even further if CMPs are used to execute the task. We can also detect the point where the errors start to increase notably, i.e., at stiffness values lower than 50, $k_s < 50$ N $\cdot$ m/rad. Based on these results, the *low gain* stiffness value in all successive experiments was set to $k_s = 50$ N $\cdot$ m/rad. It should be noted that the approach using point mass models $\mathrm{M}_{\mathrm{task}}$ was the only approach with knowledge of the object mass. The proposed CMPs do not need any prior information about the task after the task-specific feedforward torques have been learned.

Example joint torques, presented in Fig. 3, show contributions of different torque components during CMP learning and execution. In the left column, where different torque components during CMP learning are shown, the effect of different masses can be seen. When the mass gets bigger, and is not included in the dynamical model, the contribution of the feedback torque gets larger, as the robots controller needs to compensate for tracking errors. This torque is then stored and encoded in the CMP (25). The right column shows torques while executing the learned CMP (3) at high stiffness values ($k_s = 1000\,\mathrm{N}\cdot$ m/rad). The additional feedforward component, i.e., CMP torque, marked with a dashed red line, compensates for task specific dynamics and minimizes the feedback torque component produced by the robot's controller τ_{fb}. While the feedback term is significant during learning, it drops to minimal values during CMP execution. This enables low feedback gains k_s while maintaining high tracking accuracy without the need for task-specific dynamical models.

B. Collision Evaluation

In the next experiment, we evaluated the CMPs while unexpectedly colliding with an object. The robot performed a simple downward movement while an obstacle was blocking its path. Previously, the movement was executed without the obstacle in order to learn the corresponding task-specific torques. A soft object was used, as to avoid extreme forces and possible robot damage while exhibiting stiff movement. The movement into the obstacle was performed three times using: 1) standard control ($\mathrm{M}_{\mathrm{robot}}$) with high gains ($k_s = 1000\,\mathrm{N}\cdot$ m/rad); 2) standard control ($\mathrm{M}_{\mathrm{robot}}$) with low stiffness value ($k_s = 50\,\mathrm{N}\cdot$ m/rad); and 3) CMPs with low stiffness value ($k_s = 50\,\mathrm{N}\cdot$ m/rad). Fig. 4 shows three series of snapshots for each task execution. Fig. 5 depicts the plots of TCP tracking errors $\boldsymbol{p}_a(t) - \boldsymbol{p}_d(t)$, measured tool center point forces and positions along the z axis, respectively. In Figs. 4 and 5, we can observe that while using the high-gain control, the robot simply tracks the desired trajectory and penetrates into the obstacle. While the tracking error remains low all through the movement, the forces rise significantly after the collision. When using the standard low-stiffness control ($k_s = 50\,\mathrm{N}\cdot$ m/rad), forces do not rise as much even after the collision, but the system is unable to track the trajectory and the errors are high throughout the whole movement (see Fig. 5). Poor tracking can also be seen in the middle four snapshots of Fig. 4. The last series of snapshots show the movement while using low stiffness settings ($k_s = 50\,\mathrm{N}\cdot$ m/rad) and CMPs. In Fig. 5, we can observe low tracking error before the collision, which indicates good trajectory tracking despite low stiffness settings. After the collision the error rises, as can be expected and as it is desirable for a compliant behavior. The object remains almost nondeformed as the forces rise to approximately just $1/4$ of the ones exhibited while using standard high-gain control. This experiment demonstrate that CMPs combine the advantages of high-gain feedback control (high tracking accuracy) and compliant behavior (low forces caused by unexpected collisions) while eliminating their disadvantages (high contact forces and poor tracking), all without using analytical models of task dynamics. All three executions and corresponding results can also be seen in the accompanying video.

C. Discrete CMP Generalization

The experimental setup used to evaluate generalized discrete CMPs was identical to the one used in Section V-A (see also Fig. 1). The evaluation was done in two parts. In the first part, we evaluated generalization performed over a 1-D query and over a range of different stiffness settings, while in the second part, we evaluated generalization over a 2-D query. We assume accurate queries are given to the system as input data.

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

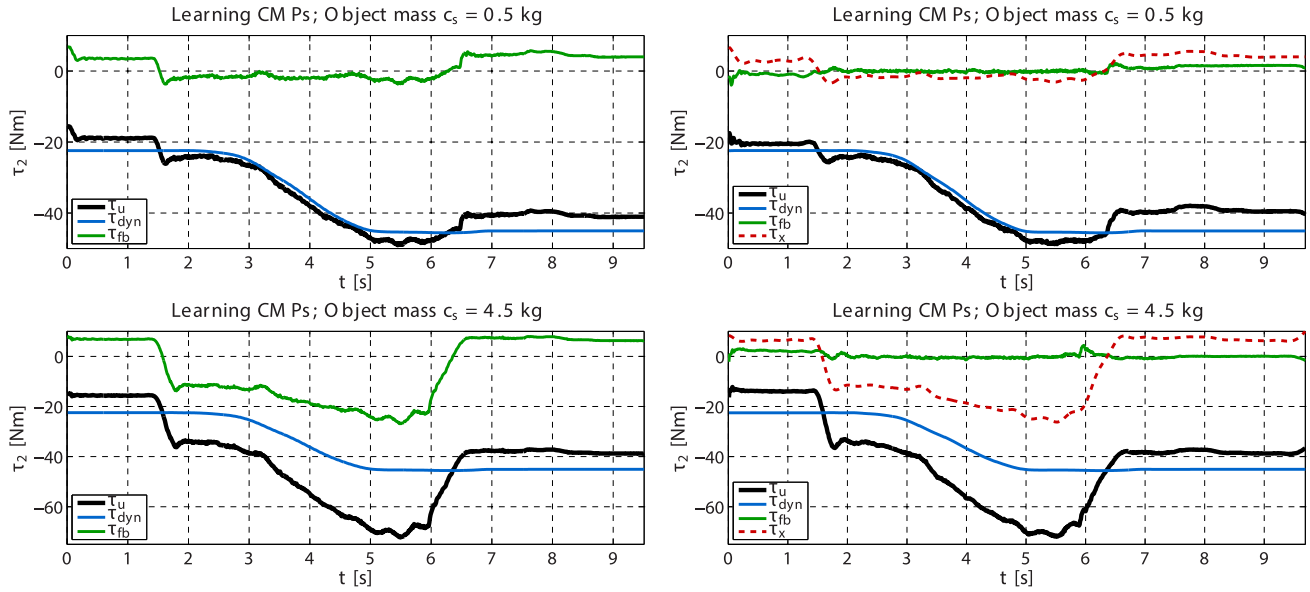

Fig. 3. Example joint torque values during CMP learning and execution. Graphs in the left column show different torque components while learning CMPs, i.e., corresponding torque learning, during a pick and place task of an object with mass of 0.5 and 4.5 kg (24–25). While the thick black line denotes the complete joint's measured torque $\tau_m = \tau_u$, blue and green represent its components, the torques contributed by Kuka's own dynamical model $\tau_{\text{dyn}} = f_{\text{dynamic}}(q, \dot{q}, \ddot{q})$ and the learned and stored feedback torque $\tau_x = \tau_{\text{fb}}$, respectively. Graphs in the right column show example joint's torque components during CMP execution with high stiffness settings $k_s = 1000\,\text{N}\cdot$ m/rad and varying object masses (3). Again, the thick black line represents complete torque of the example joint τ_u and the blue and green line torques contributed from Kuka's own dynamical model $\tau_{\text{dyn}} = f_{\text{dynamic}}(q, \dot{q}, \ddot{q})$ and the feedback torque contributed by the robot's controller τ_{fb}. The dashed red line shows additional feedforward toque encoded in the CMP τ_x. It can be observed that the contribution from the robots controller drops significantly during CMP execution. This enables us to lower the stiffness, i.e., the feedback gain, and compliantly move the robot while maintaining high tracking accuracy (without the need for task-specific dynamical models).

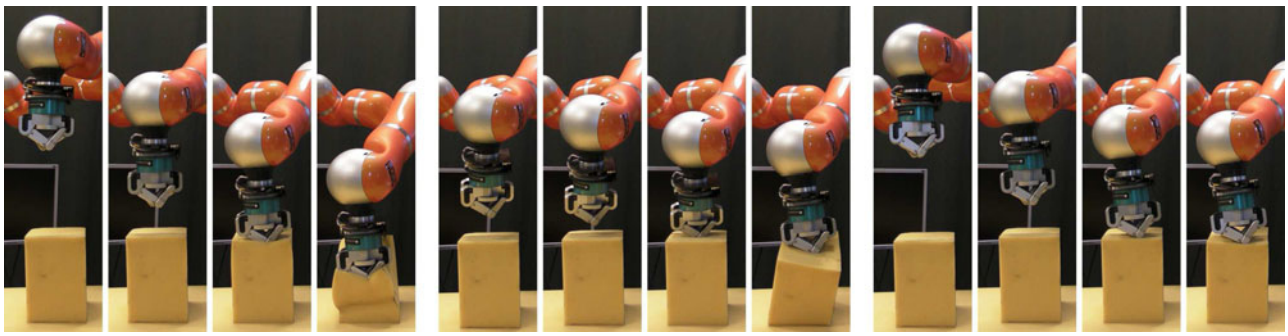

Fig. 4. Robot colliding with an object with different stiffness settings and control approaches. In the first series of four images, the robot moved in a stiff manner ($k_s = 1000\,\text{N}\cdot$ m/rad). The second set of images shows a collision while executing a compliant movement ($k_s = 50\,\text{N}\cdot$ m/rad). Both of the movements were executed while using a standard closed-loop control. The last set of images shows the collision while moving in a compliant manner using CMPs, given by (3), where the stiffness values were set to $k_s = 50\,\text{N}\cdot$ m/rad.

The previously obtained set of CMPs (41) was used for generalization over a 1-D query, i.e., a varying object mass c_s (40). As described in Section IV, statistical methods were used to *train* Gaussian process regression, which was then used to calculate appropriate CMPs. The generalized CMPs could move an object with arbitrary mass within the training space defined by example set (41). New, generalized CMPs were executed for nine different queries, covering the demonstrated as well as in-between weights. For each query, the task was executed with eight different stiffness settings. Fig. 6 compares the maximum tracking errors e_m of the generalized CMPs to the maximum tracking errors e_m obtained when executing the originally demonstrated CMPs. We can observe a slight, but not statistically significant increase in the tracking error at the lower stiffness values, i.e., when the system is more susceptible to inaccuracies in generalized torque signals.

The second part of this section focuses on CMPs generalized over a 2-D query and their tracking accuracy w.r.t. query points. Throughout this experiment, a stiffness setting of $k_s = 50\ \text{N}\cdot\text{m/rad}$ was used. In addition to varying the object mass, the final robot position was also changing. The task was to move the hand weight to different final positions. The queries were defined as

$$\boldsymbol{c} = [c_g, c_m]^T \tag{44}$$

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

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

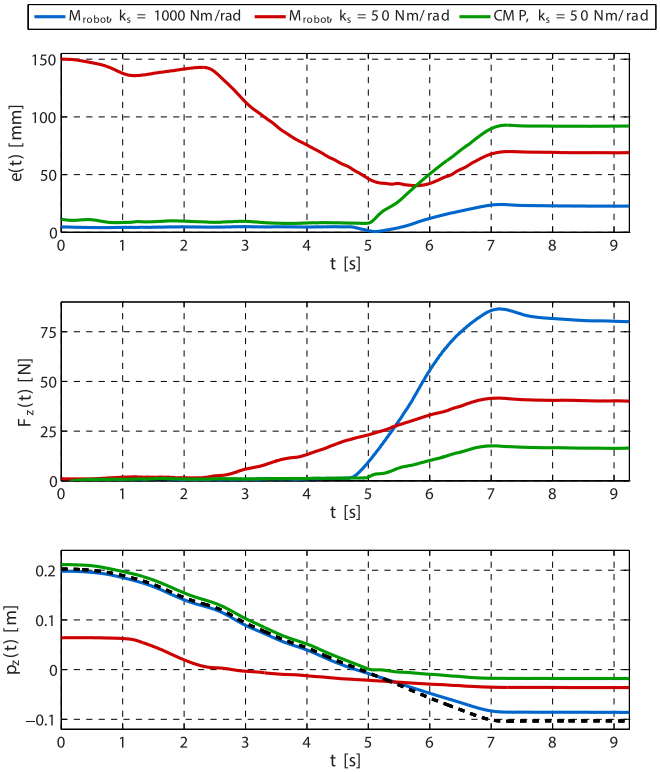

Fig. 5. Robot colliding with an object while using different stiffness settings and control approaches. The graphs present collision trajectories and forces under two different stiffness settings ($k_s = 1000\,\mathrm{N}\cdot$ m/rad and $k_s = 50\,\mathrm{N}\cdot$ m/rad) and with two different control approaches ($\mathrm{M_{robot}}$ 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.

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

$$\boldsymbol{Q}_x = \{\tilde{\boldsymbol{q}}_{xj}, c_{gj}\}_{j=1}^{6} \tag{45}$$

was tracked five 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 to 4.5 kg. Altogether, 30 example pairs of motion trajectories and corresponding torques $\{\tilde{\boldsymbol{q}}_x, \tilde{\boldsymbol{\tau}}_x\}$ 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

$$\boldsymbol{H}_x^{\mathrm{CMP}} = \{\boldsymbol{w}_{q\,k}, \boldsymbol{g}_{qk}, \boldsymbol{w}_{\tau\,k}, \upsilon_k, \boldsymbol{c}_k\},\ k = 1, \ldots, 30. \tag{46}$$

Example CMPs were exploited as described in Section IV to learn the generalization function (35) by Gaussian process regression, which was then used 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 discrete compliant tasks, we executed 99 compliant pick-and-place movements to

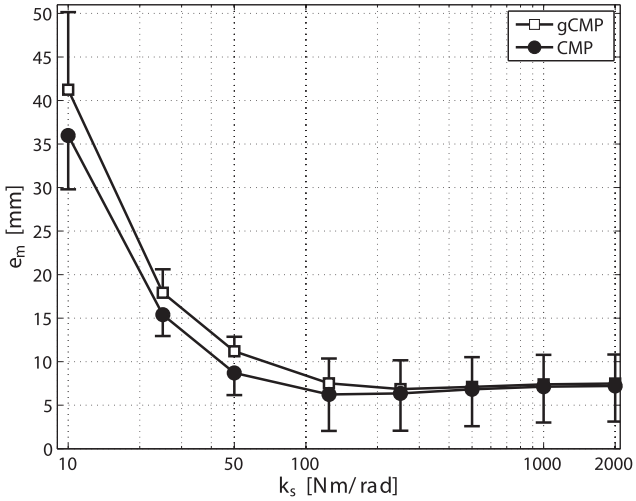

Fig. 6. Mean and standard deviation of discrete task's maximum error e_m. the top line shows the mean and standard deviation while executing CMPs generalized over a one dimension query (gCMP). The bottom line, showing the maximal mean and standard deviation while executing learned CMPs (CMP), was gained in the previous experiment (see Fig. 2) and is included for comparison.

different positions and with varying masses. These new CMPs covered the whole training space, including 30 training and 69 new query points. The goal height varied by a step of 22.5 mm, while the object mass was changed in steps of 0.5 kg. Example executions of generalized CMPs can be seen in the accompanying video.

For each task execution, maximum error e_m was calculated using (43). Table II shows the maximum errors for each generalized CMP. Results are also shown in Fig. 7 and in the accompanying video. Note that the tracking error is slightly larger when 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.

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

$$m_e = 18.2(4.4)\,\mathrm{mm}. \tag{47}$$

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 (see Table I) or generalized CMPs using a 1-D query . This small rise in the tracking error can be attributed to errors introduced through corresponding torques by statistical generalization.

D. Periodic CMPs

In the last scenario, the proposed algorithm was evaluated on a task where the robot was holding a handle of a height-adjustable table as shown in Fig. 8. In order to gain a proper inverse dynamical model $\boldsymbol{f}_{\mathrm{dynamic}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ analytically, the table would need to be mathematically modeled. Due to the structure of the table, this would be a complex and time consuming task.

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

TABLE II
MAXIMUM ERRORS FOR GENERALIZED CMPS UNDER VARYING QUERIES

		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	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
mass	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
c_m [kg]	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

All error values are in millimeters.

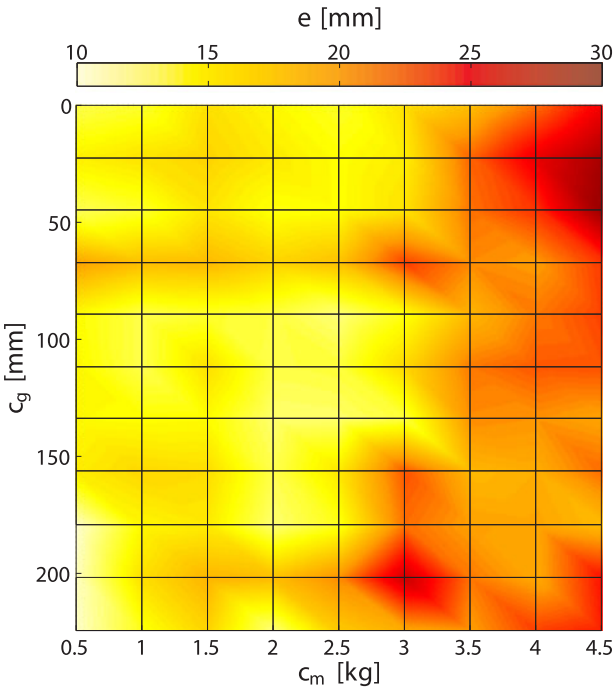

Fig. 7. Maximum errors e_m that resulted from executing generalized CMPs computed at different 2-D queries.

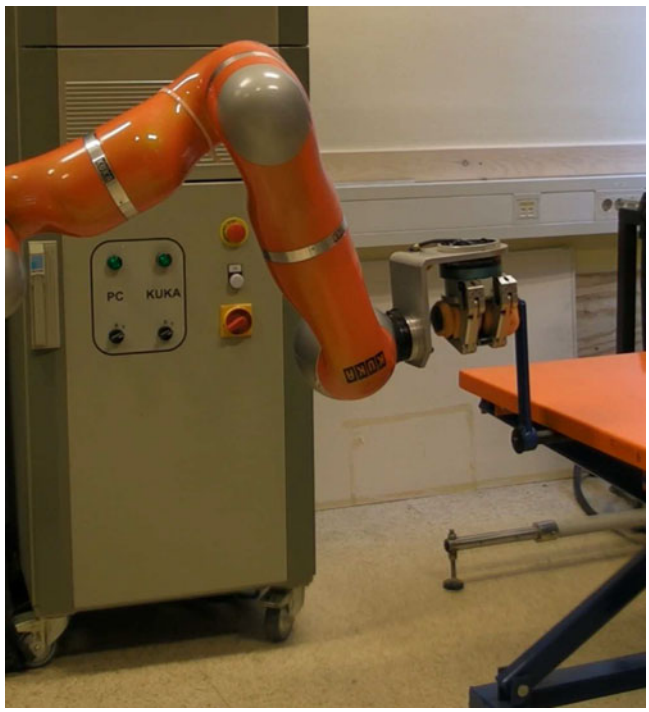

Fig. 8. Experimental setup for periodic tasks.

Instead, we used the proposed approach to gain the task-specific dynamics, i.e., appropriate torques for each task variation. The variations in the task of turning the handle were defined by two queries

$$\boldsymbol{c} = [c_h, c_\omega]^T. \tag{48}$$

The first query c_h defines the height of the table, and the second c_ω defines the frequency of rotation. The example motion trajectories

$$\boldsymbol{Q}_x = \{\tilde{\boldsymbol{q}}_{xj}, c_{hj}\}_{j=1}^{5} \tag{49}$$

were obtained by kinesthetic guiding at example table heights c_h, varying by 80 mm. We recorded only five example trajectories, which were then executed by a robot at five different frequencies c_ω, ranging from 0.4 to 2 rad/s, using a high-gain feedback controller. This resulted in 25 pairs of motion trajectories and corresponding torques $\{\tilde{\boldsymbol{q}}_x, \tilde{\boldsymbol{\tau}}_x\}$, obtained by executing variations of recorded motions demonstrated by a human. They cover all combinations of example query points $[c_h, c_\omega]$. In this way, we obtained an example set of 25 periodic CMPs

$$\boldsymbol{H}_x^{\mathrm{CMP}} = \{\boldsymbol{w}_{q\,k}, \boldsymbol{g}_{qk}, \boldsymbol{w}_{\tau\,k}, \upsilon_k, \boldsymbol{c}_k\},\ k = 1, \ldots, 25. \tag{50}$$

This training data was used to estimate the generalization function (35) using Gaussian process regression. Once the appropriate hyperparameters were obtained, the robot was able to compliantly turn the handle at arbitrary table height and frequency throughout the whole training area, while calculating the appropriate CMPs online. To show that the system can calculate the CMPs in real time, the frequency query was defined as a saw-like trajectory, while the changing table height was

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

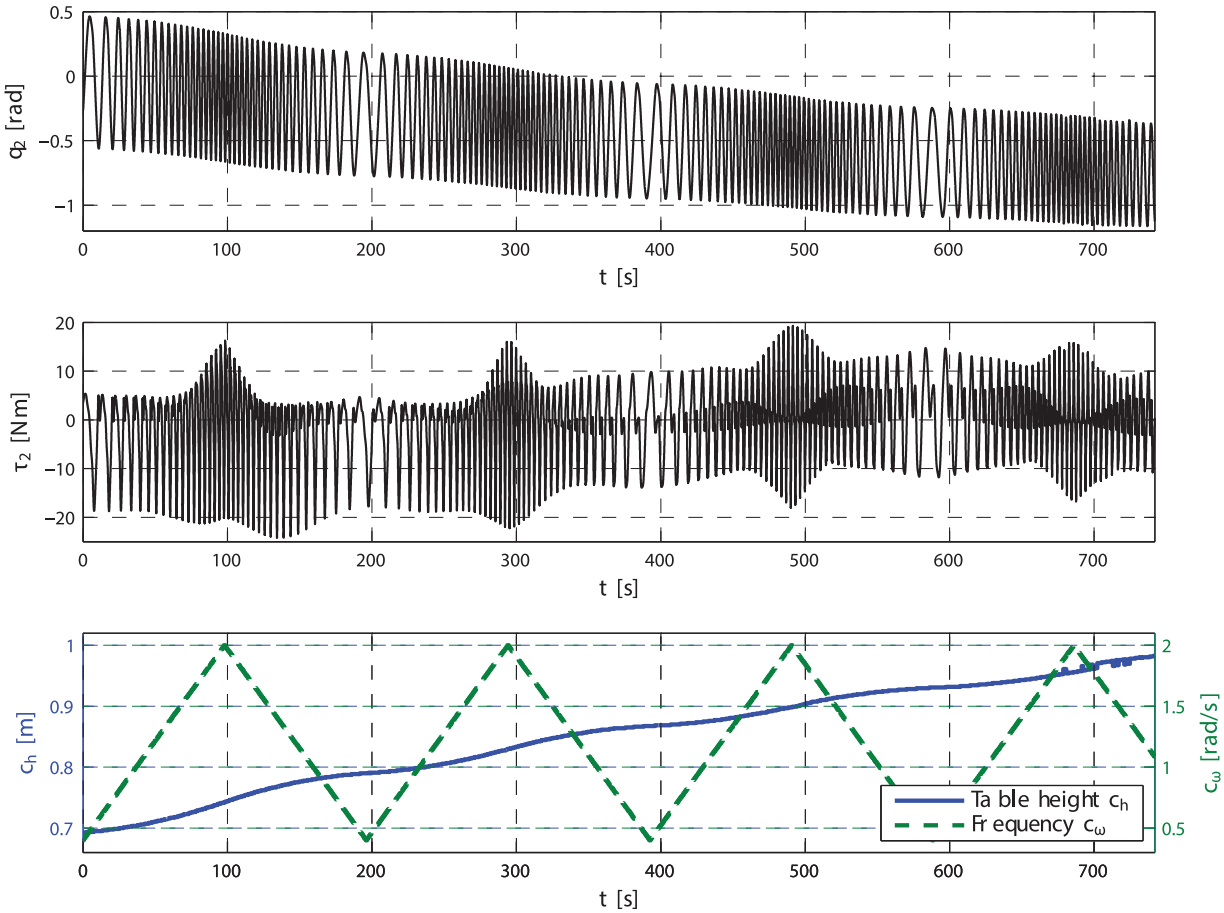

Fig. 9. Top and middle plot show two example joint and torque trajectories, respectively (example second joint). The bottom plot shows both queries throughout the experiment.

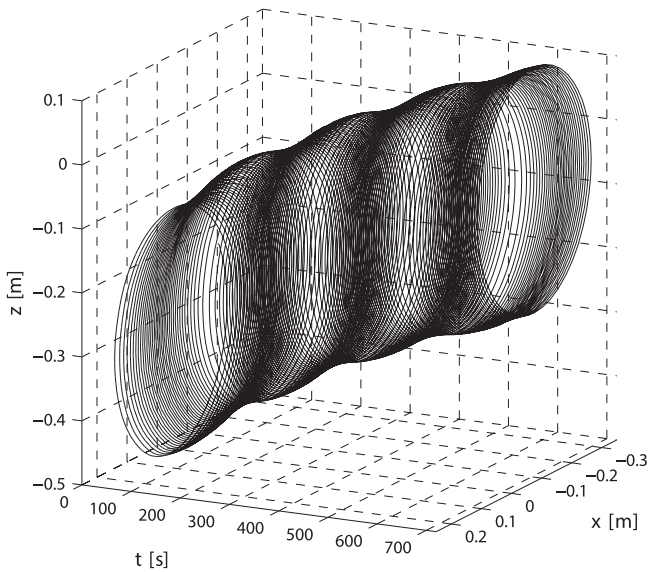

Fig. 10. Task space trajectory in two crucial dimensions (plane of rotation) as a function of time.

estimated by vision. Note that the height of the center of rotation of the handle changes while the handle is rotated. This is clearly seen in the last part of the accompanying video.

The results are shown in Fig. 9. It is evident that the proposed system adapts to both queries smoothly and continuously.

Note that the DMP frequency parameter only allows frequency adaptation for motion trajectories. Corresponding torque signals are usually too complex to be modified by a single parameter. However, with the proposed statistical approach, they can by modified appropriately online. Another important aspect is the resulting task space trajectory, which is shown in Fig. 10, where x and z coordinates (plane of rotation) are shown as the function of time. Here, we can see the adaptation of CMPs to the changes of task descriptors, i.e., the height of the table and the frequency of turning. The results are also shown in the video.

VI. Discussion

The paper presents and evaluates the new concept of CMPs and the accompanying techniques for their execution, learning, and generalization. The proposed approach combines the advantages of stiff movements (high tracking accuracy) while exhibiting low contact forces of compliant movements without the need for task specific dynamic models. The experiments showed CMPs are simply and promptly applied to various tasks. Only a few human demonstrations are needed to successfully execute a dynamically varying discrete or period task in a compliant manner.

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

While the low number of needed human demonstrations and no need for experts make CMPs suited for service tasks in home environments, the approach is primarily intended for industrial applications. Small and medium enterprises (SMEs), adapting to the current market, handle ever smaller batches of more diverse products. Because of long setup times and the need for programming experts, a standard robot cell is an impractical and financially unsound solution for SMEs. Adaptability of CMPs eliminates these shortcomings and makes the use of robots viable for small batches of varying products. For example, a pick and place task is quite common in an industrial setting. A shared workspace and objects of varying mass and size typically require long setup times for each new product. If we have a shared workspace and objects of varying mass, a long setup time would be needed for each new object when using standard approaches. As was shown in our experiments, this can be easily solved with CMPs. Another example industrial task, similar to the selected periodic evaluation task, is turning of an object in assembly. Furthermore, the approach is applicable for future service robots, similar to valve turning in the recent DARPA challenge [57]. As the initial demonstration for each new product can be quickly done by nonexperts, high contact forces due to product abnormalities and errors in positions is avoided as the robot exhibits high compliance.

Compliant behavior would usually be achieved through high precision dynamical models describing each specific task. With the rising complexity of humanoid robots, accurate dynamic models also become more complex. Real physical systems include a variety of unknown nonlinearities, such as friction, gearboxes, elastic, and pneumatic elements, etc. These aspects can degrade the dynamical model which consequently effects the tracking accuracy [48]. Tracking errors can be mitigated with high feedback gains, but that in turn raises stiffness and unforeseen impact forces. Even more critically than the aforementioned issues that can degrade the accuracy of dynamical models, tasks can also introduce complex dynamics, which is hard or even impossible to model analytically. Moreover, these models are task specific, hence, each new task requires a different dynamical model. The periodic task of handle turning presented in this paper is an example of a complex, hard to model, dynamic task. With this in mind, the presented CMPs are an alternative to a standard approach using mathematically defined dynamical models. The proposed approach can take into account the aspects of the robot and task dynamics that were not considered when specifying the mathematical dynamical model.

In the second step of the proposed approach, the trajectories are executed with high-gain feedback loop, which enforces task-specific dynamics. Note here that we assume that the robot is accurately tracking the desired trajectory. The executed torque profiles are recorded and used for training TPs. The combination of TP and DMP defines a CMP. While the proposed approach does not require prior dynamical model, such model can be used if available. In our experiments, Kuka LWR's own dynamical model was available, but the CMP still mitigated any model inaccuracies and provided the unmodeled aspects of the task dynamics. In essence, learned TPs are used to compensate for dynamic model errors and task-specific interaction forces.

The stiff behavior when learning task-specific dynamics in the second step requires a controlled environment to prevent damage to the robot. Instead of executing the trajectory in a stiff manner, task specific dynamics could alternatively be gained by reinforcement learning [58], [59]. This is, however, a much lengthier process than the approach proposed in this paper.

While a rigorous analysis of the stability of generalized CMPs is beyond the scope of this paper and would also not be general but task dependent, we provided an intuitive stability analysis in Section IV.A. As long as the task descriptors remain within the training space, the generalized CMPs smoothly transition between the demonstrated CMPs, which are assumed to be stable. It is therefore intuitive that the generalized CMPs are stable as well. In our experiments, the generalized CMPs were always stable unless extreme perturbations were applied to the robot. The development of an additional adaptive controller, possibly combined with DMP phase stopping techniques [60], will be addressed in our future work.

Another approach of ensuring required tracking accuracy is through variable impedance control [23]–[25]. While variable impedance controllers—just as our approach—vary the gain, the core of their approach is in essence quite different. This field of research mainly focuses on achieving desired dynamic behavior by adjusting feedback gains to higher values when necessary. While a (imperfect) robot's dynamical model is assumed, gain schedules, i.e., stiffness setting trajectories, are learned [23], [24]. On the other hand, we focus on achieving compliance throughout the task and simply reduce the gain—to a value we have statistically determined to show no significant change in the behavior. While there are similarities between our study and the study being done in the field of variable impedance control, our paper focuses on learning task-specific, i.e., corresponding feedforward torques while minimizing stiffness to a minimum throughout the movement.

In order to avoid CMP demonstration for each task variant, we proposed to use statistical techniques. They can generalize a set of example CMPs to new task variants, which proves useful if a robot needs to execute a similar task several times under varying conditions, e.g., everyday tasks in domestic settings. In general, task descriptors/queries are perceived by a sensory system both during learning and during task execution. If there are any errors in the sensor readings, they do not affect the performance of the system as long as they are the same in both phases. They are included in the database when learning and automatically compensate themselves when the task is executed. But if the sensors readings vary, the errors will not be compensated and the task will fail. This is the case for any learning approach as well as for standard control approaches that use analytically defined dynamical models.

Our evaluation showed the viability of the proposed approach. The results comparing CMPs to standard approaches at various stiffness settings show the viability of CMPs as the tracking accuracy was higher compared to standard approaches. The magnitude of error stayed in the same area even at low stiffness values. Evaluation also showed that while colliding with an obstacle, CMPs produce low impact forces of compliant movements, while at the same time exhibiting high tracking

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

accuracy characteristic for stiff motion resulting from the classic high-gain feedback control.

Generalized CMPs were also evaluated. By using Gaussian process regression, we were able to generalize CMPs to any set of queries within the training space. By evaluating the approach on a discrete pick-and-place task we showed that the tracking error introduced by generalization remains small. In the experiment involving periodic motion, a hard-to-model task of raising a height-adjustable table was demonstrated. We showed that the proposed approach was able to calculate the appropriate CMPs in real time. While alternative generalization techniques to Gaussian process regression might reduce the errors introduced by the application of the technique originally proposed in [17], many of them would not be able to adapt CMPs online due to the significant rise in the calculation time [16].

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] D. C. Bentivegna, C. G. Atkeson, and G. Cheng, "Learning tasks from observation and practice," *Robot. Auton. Syst.*, vol. 47, no. 2, pp. 163–169, 2004.

[5] 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.

[6] A. Ude, C. G. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robot. Auton. Syst.*, vol. 47, no. 2, pp. 93–108, 2004.

[7] 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. Robot. Autom.*, Washington, DC, USA, 2002, pp. 1390–1397.

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

[9] 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.

[10] M. Stommel, M. Beetz, and W. Xu, "Model-free detection, encoding, retrieval, and visualization of human poses from kinect data," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 865–875, Apr. 2015.

[11] 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, Dec. 2008.

[12] L. Rozo, S. Calinon, and D. Caldwell, "Learning force and position constraints in human–robot cooperative transportation," in *Proc. IEEE Int. Symp. Robot Human Interactive Commun.*, Edinburgh, Scotland, U.K., 2014, pp. 619–624.

[13] L. Rozo, S. Calinon, D. Caldwell, P. Jiménez Schlegl, and C. Torras, "Learning collaborative impedance-based robot behaviors," in *Proc. AAAI Conf. Artificial Intell.*, Bellevue, WA, USA, 2013, pp. 1422–1428.

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

[15] 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.

[16] 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, Oct. 2010.

[17] 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.

[18] H. Ben Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human–robot cooperation tasks," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, 2014, pp. 2831–2837.

[19] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, "Embodied symbol emergence based on mimesis theory," *Int. J. Robot. Res.*, vol. 23, nos. 4/5, pp. 363–377, 2004.

[20] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation," *IEEE Robot. Autom. Mag*, vol. 17, no. 2, pp. 44–54, Jun. 2010.

[21] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 943–957, Oct. 2011.

[22] I. Havoutis and S. Ramamoorthy, "Motion planning and reactive control on learnt skill manifolds," *Int. J. Robot. Res.*, vol. 32, nos. 9/10, pp. 1120–1150, 2013.

[23] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 820–833, 2011.

[24] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human–robot interaction," *IEEE Trans. Haptics*, vol. 7, no. 3, pp. 367–380, Jul.–Sep. 2014.

[25] G. Ganesh, N. Jarrassé, S. Haddadin, A. Albu-Schaeffer, and E. Burdet, "A versatile biomimetic controller for contact tooling and haptic exploration," in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Paul, MN, USA, 2012, pp. 3329–3334.

[26] M. Frigola, A. Casals, and J. Amat, "Human–robot interaction based on a sensitive bumper skin," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Beijing, China, 2006, pp. 283–287.

[27] 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, Jun. 2011.

[28] Y. Ohmura, Y. Kuniyoshi, and A. Nagakubo, "Conformable and scalable tactile sensor skin for curved surfaces," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, USA, 2006, pp. 1348–1353.

[29] P. Mittendorfer and G. Cheng, "Humanoid multi-modal tactile sensing modules," *IEEE Trans. Robot.*, vol. 27, no. 3, pp. 401–410, Jun. 2011.

[30] S. Backus and A. Dollar, "Robust resonant frequency-based contact detection with applications in robotic reaching and grasping," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 5, pp. 1552–1561, Oct. 2014.

[31] 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, Sep. 2008.

[32] C. Laschi, M. Cianchetti, B. Mazzolai, L. Margheri, M. Follador, and P. Dario, "Soft robot arm inspired by the octopus," *Adv. Robot.*, vol. 26, no. 7, pp. 709–727, 2012.

[33] 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.

[34] 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.

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

[36] A. Bicchi and G. Tonietti, "Fast and 'soft-arm' tactics [robot arm design]," *IEEE Robot. Automat. Mag*, vol. 11, no. 2, pp. 22–33, Jun. 2004.

[37] S. A. Migliore, E. A. Brown, and S. P. DeWeerth, "Biologically inspired joint stiffness control," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 4508–4513.

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

[39] A. Jafari, N. G. Tsagarakis, I. Sardellitti, and D. G. Caldwell, "A new actuator with adjustable stiffness based on a variable ratio lever mechanism," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 1, pp. 55–63, Feb. 2014.

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

[41] J. K. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *Proc. IEEE Conf. Decision Control Including Symp. Adaptive Processes*, Albuquerque, NM, USA, 1980, pp. 95–100.

[42] N. Hogan, "Stable execution of contact tasks using impedance control," in *Proc. IEEE Int. Conf. Robot. Autom.*, Raleigh, NC, USA, 1987, pp. 1047–1054.

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2015.2510165, IEEE/ASME Transactions on Mechatronics

[43] C. G. Atkeson and J. M. Hollerbach, "Kinematic features of unrestrained vertical arm movements," *J. Neurosci.*, vol. 5, no. 9, pp. 2318–2330, 1985.
[44] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural Netw.*, vol. 11, no. 7, pp. 1317–1329, 1998.
[45] D. W. Franklin and D. M. Wolpert, "Computational mechanisms of sensorimotor control," *Neuron*, vol. 72, no. 3, pp. 425–442, 2011.
[46] J. W. Krakauer, M.-F. Ghilardi, and C. Ghez, "Independent learning of internal models for kinematic and dynamic control of reaching," *Nature Neurosci.*, vol. 2, no. 11, pp. 1026–1031, 1999.
[47] D. Nguyen-Tuong and J. Peters, "Learning robot dynamics for computed torque control using local gaussian processes regression," in *Proc. ECSIS Symp. Learning Adaptive Behav. Robot. Syst.*, Edinburgh, Scotland, U.K., 2008, pp. 59–64.
[48] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, vol. 3. New York, NY, USA: Wiley, 2006.
[49] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Syst.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.
[50] M. Schwarz and S. Behnke, "Compliant robot behavior using servo actuator models identified by iterative learning control," in *Proc. 17th RoboCup Int. Symp.*, Eindhoven, The Netherlands, 2013, pp. 207–218.
[51] M. Gautier, A. Jubien, and A. Janot, "Iterative learning identification and computed torque control of robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, 2013, pp. 3419–3424.
[52] A. Albu-Schaffer, C. Ott, and G. Hirzinger, "A passivity based cartesian impedance controller for flexible joint robots-part ii: Full state feedback, impedance design and experiments," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, USA, 2004, pp. 2666–2672.
[53] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 1. New York, NY, USA: Springer, 2006.
[54] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarčič, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Auton. Robot.*, vol. 27, no. 1, pp. 3–23, 2009.
[55] T. Petrič, A. Gams, A. J. Ijspeert, and L. Žlajpah, "On-line frequency adaptation and movement imitation for rhythmic robotic tasks," *Int. J. Robot. Res.*, vol. 30, no. 14, pp. 1775–1788, 2011.
[56] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[57] (2015). [Online]. Available: http://www.theroboticschallenge.org/
[58] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Auton. Robot.*, vol. 33, no. 4, pp. 361–379, 2012.
[59] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, pp. 1238–1274, 2013.
[60] A. Ude, B. Nemec, T. Petrič, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, 2014, pp. 2997–3004.

Miha Deniša received the Diploma degree in electrical engineering from the University of Ljubljana, Ljubljana, Slovenia, in 2010. He is currently working toward the Ph.D. degree at the same institution.

He is a Researcher at the Humanoid and Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia. His research interests include humanoid robotics, imitation learning, and motion databases.

Andrej Gams (M'12) received the Diploma degree in electrical engineering in 2004, and the Ph.D. degree in robotics from the University of Ljubljana, Ljubljana, Slovenia, in 2009.

He is currently a Research Associate with the Department for Automatics, Biocybernetics, and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia. He was a Postdoctoral Researcher with the Biorobotics Laboratory, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, during 2012–2013. He was a Visiting Researcher at the ATR Computational Neuroscience Laboratories in the summers of 2009 and 2014. His research interests include imitation learning, control of periodic tasks, and humanoid cognition.

Dr. Gams received the Best Paper Research Award at the 19th International Workshop in the Alpe-Adria-Danube region in 2010 and the Best Scientific Paper Award at the 23rd International Conference on Robotics in Alpe-Adria-Danube Regions in 2014. He also received the Jožef Stefan Golden Emblem Award for his Ph.D. dissertation in 2012. He received the SCIEX NMS-CH Fellowship for postdoctoral studies at EPFL, Switzerland, in 2012. He has been a Program Committee Member of several conferences, including Humanoids 2011.

Aleš Ude (M'03) received the Diploma degree in applied mathematics from the University of Ljubljana, Ljubljana, Slovenia, and the Ph.D. degree from the Faculty of Informatics, University of Karlsruhe, Karlsruhe, Germany.

He is/was a Principal Investigator or Coordinator of several EU projects in the area of robotics including PACO-PLUS, Xperience, IntellAct, ACAT, and ReconCell. He is currently the Head of Department of Automatics, Biocybernetics, and Robotics, Jožef Stefan Institute, Ljubljana. He is also associated with the ATR Computational Neuroscience Laboratories, Kyoto, Japan. His research interests include autonomous robot learning, imitation learning, humanoid robot vision, humanoid cognition, and humanoid robotics in general.

Dr. Ude received the Science and Technology Agency Fellowship for postdoctoral studies in ERATO Kawato Dynamic Brain Project, Japan.

Tadej Petrič received the M.Sc. degree in electrical engineering from the University of Maribor, Maribor, Slovenia, in 2008. His M.Sc. covered modeling and robotic control of underactuated dynamic system. In 2013, he received the D.Sc. degree in robotics from the Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia. He performed a part of his doctoral research at the Department of Robotic Systems for Dynamic Control of Legged Humanoid Robots, German Aerospace Center (DLR), Oberpfaffenhofen, Germany.

In 2013, he was a Visiting Researcher at ATR Computational Neuroscience Laboratories in Japan. In 2015, he was a Postdoctoral Fellow working with Prof. Auke Ijspeert in the Biorobotics Laboratory, Swiss Federal Institute of Technology in Lausanne, Switzerland. He is currently a Research Associate at the Department for Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia. His current research includes the design of biologically plausible robot controllers that achieve robustness and adaptation to changing environments comparable to that found in humans.

Dr. Petrič received the Prof. Dr. Vratislav Bedjanič award in 2008 for his work.

1083-4435 (c) 2015 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.