# Coupling Movement Primitives: Interaction With the Environment and Bimanual Tasks

Andrej Gams, Bojan Nemec, Auke Jan Ijspeert, *Member, IEEE*, and Aleš Ude

***Abstract*—The framework of dynamic movement primitives (DMPs) contains many favorable properties for the execution of robotic trajectories, such as indirect dependence on time, response to perturbations, and the ability to easily modulate the given trajectories, but the framework in its original form remains constrained to the kinematic aspect of the movement. In this paper, we bridge the gap to dynamic behavior by extending the framework with force/torque feedback. We propose and evaluate a modulation approach that allows interaction with objects and the environment. Through the proposed coupling of originally independent robotic trajectories, the approach also enables the execution of bimanual and tightly coupled cooperative tasks. We apply an iterative learning control algorithm to learn a coupling term, which is applied to the original trajectory in a feed-forward fashion and, thus, modifies the trajectory in accordance to the desired positions or external forces. A stability analysis and results of simulated and real-world experiments using two KUKA LWR arms for bimanual tasks and interaction with the environment are presented. By expanding on the framework of DMPs, we keep all the favorable properties, which is demonstrated with temporal modulation and in a two-agent obstacle avoidance task.**

***Index Terms*—Bimanual operation, cooperative task, dynamic movement primitives (DMPs), interaction with environment.**

## I. Introduction

By moving beyond the structured environment of a manufacturing plant, robots are making their way into the everyday world that people inhabit—offices, hospitals, homes, and other cluttered and uncontrolled environments [1], including the kitchen [2], [3]. A growing portion of robotics research already directly or indirectly deals with all aspects related to complex human environments [4]. If we envision a robotic assistant in a human environment, it will probably use its sensors and existing knowledge to generate trajectories appropriate for the given tasks, for example, by generalization [5]. Despite the possibility to adapt trajectories as they are being executed [6], noise, lack of prior knowledge, and errors in the perception of the environment might not make the trajectories accurate enough for the desired manipulatory actions. The generated trajectory therefore has to be adapted to the task through autonomous exploration or learning. In this paper, we present a new approach to modulation and flexible learning of robot movements, which allow safe interaction with the environment and bimanual or cooperative multiagent tasks.

Trajectory generation depends on the type of encoding approach, and different encoding approaches also allow for different possibilities of modulation, interpolation, and categorization [7]. One of the approaches is the use of splines and wavelets [8], [9]. However, splines are nonautonomous representations with no attractor properties. While effective for imitation learning, they do not allow easy online modulation [10]. Rescaling the splines in space and time for generalization is possible, but it requires to explicitly recompute the spline nodes. Gaussian mixture regression [11] and Gaussian mixture models are another option. A mixture model approach was used in [12] and [13] to estimate the entire attractor landscape of a movement skill from several sample trajectories. To ensure stability of the dynamical system toward an attractor point, a constraint optimization problem has to be solved in a nonconvex optimization landscape. Yet another option is the use of hidden Markov models [14].

We build on dynamic movement primitives (DMPs), first introduced by Ijspeert *et al.* [15], which model attractor behaviors of autonomous nonlinear dynamical systems with the help of statistical learning techniques. DMPs provide means to encode a trajectory as a set of differential equations that can compactly

Manuscript received October 10, 2013; accepted January 31, 2014. Date of publication February 26, 2014; date of current version August 4, 2014. This paper was recommended for publication by Associate Editor C. C. Cheah and Editor G. Oriolo upon evaluation of the reviewers' comments. Initial results on the topic were published in A. Gams, B. Nemec, L. Žlajpah, M. Waechter, A. Ijspeert, T. Asfour, and A. Ude, "Modulation of Motor Primitives using Force Feedback: Interaction with the Environment and Bimanual Tasks," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5629–5635, Tokyo, Japan, 2013. Research leading to these results was supported by the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) grant agreement no. 270273, Xperience and no. 248311, AMARSi, and by SCIEX-NMS$^{CH}$ project 12.018.

A. Gams, B. Nemec, and A. Ude are with the Humanoid and Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jožef Stean Institute, Ljubljana 1000, Slovenia (e-mail: andrej.gams at ijs.si; bojan. nemec at ijs.si; ales.ude at ijs.si).

A. J. Ijspeert is with the Biorobotics Laboratory, Institute of Bioengineering, École Polytechnique Fédérale de Lausanne, Lausanne CH-1015, Switzerland (e-mail: auke.ijspeert at epfl.ch).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. The video shows the application of the proposed algorithm in four distinct tasks. The first part of the video shows the applicability of the algorithm for modulating a given trajectory in order to achieve a desired contact force with the environment. In ten epochs, the algorithm is first applied to achieve the desired contact force 0 N, i.e., to approach and not touch the object, and later 15 N. The second experiment shows the performance of the approach when coupling two trajectories in a bimanual task, where the force along the manipulated object is minimized. In the third part, we show the performance of the approach for minimizing the force along the object when one or both of the arms in a bimanual task have to avoid obstacles. The final, fourth part, shows a bimanual human–robot cooperative task, where the robots and the person have to put a lid with a tight fit on a box. The box is moved and the proposed algorithm is used to adapt the trajectories of the robots with respect to the forces exerted by the person and from contact with the box, so that by the final, eighth epoch, the robots do not need any interference to place the lid on the box.

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/TRO.2014.2304775

represent control policies, while their attractor landscapes can be adapted by only changing a few parameters. The latter can be exploited in several ways, for example, for reinforcement learning [16]–[20], statistical generalization [5], [21], or for combining separate trajectories in a dynamic way [22], [23].

The structure of DMPs enables incorporation of sensory feedback. Modulations can affect either the transformation system, the canonical system, or both systems [10]. The use of sensory feedback was demonstrated in various applications, e.g., modulations affecting the canonical system were demonstrated on different periodic tasks [24]. Other examples include modulating the transformation system for online obstacle avoidance [10], [25] or for introducing an external limit [7]. An example of modulating both systems is the so-called slow-down feedback, which is used to stop the execution of the trajectory [26].

In this paper we propose a new approach to modify trajectories, i.e., the transformation system of the DMP. We propose recording the sensory feedback as the robot moves along a trajectory and then using this feedback to improve the robot's performance the next time it moves down the same trajectory. We do not modify the original trajectory but learn a coupling term, which is fed into the original trajectory similarly to an external limit modulation [7]. The coupling term can either be the real force coupling between two manipulators/agents, or it can virtually represent an external force arising from interaction. The final shape and amplitude of the coupling term are learned in a few iterations using iterative learning control (ILC) [27]. The approach is fast and reliable for tasks which do not vary along the trials. Initial results on the approach were published in [28].

Several reasons speak for the use of iterative learning in the proposed approach. On the one hand, its appeal lies in the similarity to human learning processes, as people may practice a task many times before being able to find correct inputs to accomplish it with such a complex system as the human body [29]. In [27], the concept of ILC is well illustrated by the example of a basketball player, shooting a free throw from a fixed position, who can improve his ability to score by practicing the shot repeatedly. During each shot, the basketball player observes the trajectory of the ball and consciously plans an alteration in the shooting motion for the next attempt. ILC can be applied in exactly the same manner to learning of robotic movements [30], [31].

On the other hand, ILC features several desired properties. Just as with any learning system, ILC incorporates information-rich error signals from previous operations for subsequent iterations. Furthermore, it only adapts the control input, and not the controller, does not require extensive training, and is known to converge fast [27]. Because ILC generates its open-loop control through practice, it is also highly robust to system uncertainties [27] and can be used to achieve perfect tracking, even when the model is uncertain or unknown [32]. The novelty of the proposed approach lies in incorporating the well-defined DMP framework and the ILC into a single, robust system for modification of trajectories based on force feedback, thus surpassing the kinematic domain of the DMPs.

After related work in Section II and the basic review of DMPs in Section III, we present 1) force-based modulation of the DMPs at both velocity and acceleration levels, 2) coupling of DMPs for bimanual tasks (see Section III), 3) learning the open coupling terms with ILC (Section IV), 4) stability analysis and arguments for using both acceleration and velocity levels as compared with acceleration level only modulation (see Section V). Section VI describes interaction and bimanual experiments conducted on two KUKA LWR robots in a bimanual setting, including experiments in cooperation with humans. Section VII shows that DMP properties, such as the modulation of the duration, remain intact when coupling trajectories. The pros and cons of the approach are discussed in Section VIII, and our concluding remarks are in Section IX.

## II. Related Work

Even though controlling rigid robots while in contact with the environment can be difficult, using a force-feedback term to learn and improve task execution was considered in many robotic tasks (see, for example, [33]). The use of force feedback to change the output velocity of a manipulator was reported by Hogan [34]. On the other hand, relatively few papers discuss the use of force feedback in combination with dynamical systems or specifically DMPs.

Modifying periodic DMPs was previously demonstrated for a task of wiping a flat or curved surface [35]. Contrary to the proposed approach, complete trajectory waveforms were modified within a few periods of the task using regression methods. Formally, the approach in [35] did not rely on modulation but on learning of new trajectories, as the trajectories for the whole period of motion were constantly relearned. The approach was expanded on by Ernesti *et al.* [36] to include transient motions.

Learning interaction force skills in presence of compliant external dynamics from human demonstrations using dynamical systems was shown in [37], in which an interaction force encoded in terms of a parameterized time-invariant differential equation based upon the parallel force/position control law was used. Similarly to our proposed approach, it modulates the velocity term of its dynamical system. Applicability was shown only in virtual settings [37].

Pastor *et al.* [4], [38] have demonstrated an approach of modifying DMPs, which, similarly to the proposed method, relies on data from an execution of a discrete task to modify the trajectory in the next, perturbed execution. They implemented a low-level position and force control system that integrates with DMPs at the acceleration level, allowing for reactive and compliant behaviors. The key idea in their approach is that a successful demonstration provides a reference force for the following, possibly perturbed, executions. A controller is used to ensure the same force profile. The approach was applied for grasping of a lamp [4] and a battery operated drill, combined with sequencing to achieve complete tasks [38].

DMPs were modulated for tightly coupled dual-agent tasks by Kulvicius *et al.* [39]. In their approach, virtual forces were used to couple DMPs at acceleration levels and applied Hebbian type learning to minimize the virtual force during the execution.

Vision and touch sensors were used to determine the distance and virtual force between the two agents. An approach for bimanual operation based on dynamical systems and also applicable to DMPs was discussed by Calinon *et al.* [40]. The approach expresses the nonlinear force modulating the movement in the original DMP formulation as additional sets of virtual springs, adding local corrective terms that can swiftly react to perturbations during reproduction.

## III. Modulating Dynamic Movement Primitives

### A. Dynamic Movement Primitives

DMPs have been thoroughly discussed in the literature [10], [15], [26]. Here, we provide only basic information, which is based on the formulation provided in [5] and [26]. For a single degree of freedom (DOF) denoted by $y$, in our case one of the external task-space coordinates, a DMP is defined by the following system of nonlinear differential equations

$$\tau \dot{z} = \alpha_z(\beta_z(g-y)-z)+f(x) \tag{1}$$

$$\tau \dot{y} = z. \tag{2}$$

Here, $f(x)$ is defined as a linear combination of nonlinear radial basis functions

$$f(x) = \frac{\sum_{i=1}^{N} w_i \Psi_i(x)}{\sum_{i=1}^{N} \Psi_i(x)} x \tag{3}$$

$$\Psi_i(x) = \exp(-h_i(x-c_i)^2) \tag{4}$$

where $c_i$ are the centers of radial basis functions distributed along the trajectory and $h_i > 0.0$ their widths. Provided that parameters $\alpha_z, \beta_z, \tau > 0$, and $\alpha_z = 4\beta_z$, the linear part of the system (1) and (2) is critically damped and has a unique attractor point at $y = g$ and $z = 0$. A phase variable $x$ is used in (1), (3), and (4). It is utilized to avoid direct dependence of $f$ on time. Its dynamics are defined by

$$\tau \dot{x} = -\alpha_x x \tag{5}$$

with initial value $x(0) = 1$. $\alpha_x$ is a positive constant.

The weight vector $\mathbf{w}$, which is composed of weights $w_i$, defines the shape of the encoded trajectory. Learning of the weight vector has been described in [5] and [15]. Multiple DOFs are realized by maintaining separate sets of (1)–(4), while a single canonical system given by (5) is used to synchronize them.

### B. Modulation for Interaction With Environment

DMPs can be modulated online to take dynamic events from the environment into account. Those online modulations are among the most important properties offered by the dynamical systems approach [10]. An example of spatial modulation is including an obstacle avoidance term in (1) [10], [25]

$$\tau \dot{z} = \alpha_z(\beta_z(g-y)-z)+f(x)+C_m \tag{6}$$

where $C_m$ is the modulation term. In this paper, we call this kind of modulation as a modulation at the *acceleration* level.

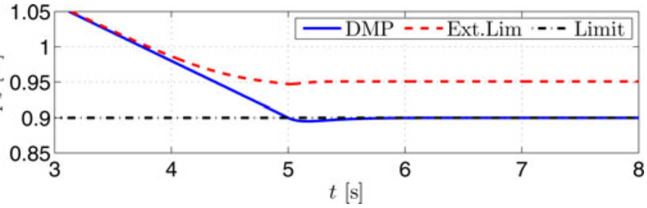

Fig. 1. Response in presence of an external limit according to (7) and (8), depicted in red (dashed), with the limit set at $y_L = 0.9$ m (black dash dot). Original DMP trajectory in solid blue.

Another spatial modulation includes a simple repulsive force to avoid moving beyond a given position in the task space [7]. Such a repulsive force can be specified by modifying (2) into

$$\tau \dot{y} = z + h(y) \tag{7}$$

while leaving (1) in the original form. In this paper, we call this kind of modulation as modulation at the *velocity* level. A simple repulsive force to avoid hitting $y_L$ can be defined as [7]

$$h(y) = -\frac{1}{\gamma(y_L - y)^3} \tag{8}$$

where $y_L$ is the known limit. Modification of a DMP that encodes a straight trajectory from 1.3 to 0.9 m in 5 s, using (7) and (8) and $\gamma = 10^5$, $y_L = 0.9$ m, results in the response shown in red (dashed) in Fig. 1.

The external limit may be static, as shown in Fig. 1, or moving, determined, for example, by vision. In both cases, defining the external limit as in (7) and (8) can prevent the robot from getting into contact with objects in its environment because the repulsive force of the limit acts before actual contact takes place. We therefore propose a modification of the external limit approach, not by changing (7) but by defining a different repulsive force. Instead of using (8), we propose using the measured force $F$, which arises from the interaction with the environment

$$\tau \dot{y} = z + cF(t) \tag{9}$$

where $c$ is a scaling constant. $F(t)$ can either be the real measured force of contact or a virtual force. The virtual force can be defined as (for one DOF)

$$F(t) = kd(t) \tag{10}$$

where $k$ is the object (or environment) stiffness, and $d$ is the depth of penetration into the object.

A slight overshoot of forces upon environment contact appears when using the proposed velocity level modulation. To minimize this overshoot of forces (the error), we add a derivative of the measured force at the acceleration level. Similarly to PD controllers, this additional coupling introduces damping. The equation of a DMP with coupling at both the velocity and acceleration levels becomes

$$\tau \dot{z} = \alpha_z(\beta_z(g-y)-z)+f(x)+c_2\dot{C} \tag{11}$$

$$\tau \dot{y} = z + C \tag{12}$$

$$C = cF(t) \tag{13}$$

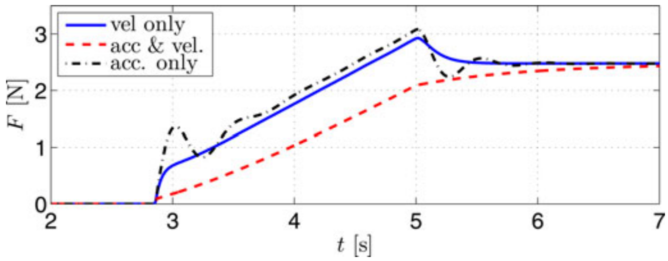

Fig. 2. Simulation results that show the difference of adding a coupling term at the velocity level (the blue solid line), which also includes the derivative of the force at the acceleration level (the red dashed line), or acceleration level only (the black dash-dotted line). The red trajectory does not overshoot, while the blue and black do. Acceleration only modulation produces a more oscillatory response. As discussed in Section V, higher coupling gains are required for acceleration only modulation to achieve the same steady-state value: in this case, 36 times. The trajectory was encoded to start at 1.4 m and end at 0.7 m in 5 s, while an object was encountered at 1 m.

with $c_2$ a scaling constant. Fig. 2 shows the difference of using velocity modulation, both velocity and acceleration with $c_2 = 30$, determined empirically, and only acceleration level modulation. When using velocity modulation only or acceleration modulation only, the force overshoots at time $t = 5$ s. This overshoot results in oscillations in the direction of the force. We show in Section V that adding the coupling term to both velocity and acceleration level is better than only to the acceleration level, because the latter results in significantly larger oscillations in the direction of the force. This performance would impose restrictions on the use of the proposed iterative learning algorithm and would not allow effective learning.

Properly selected scaling factors $c$ and $c_2$ ensure rapid and compliant behavior of the robot. Even so, the force $F$ and therefore the modification of the trajectory only appears after the contact with the environment. We therefore propose to employ an ILC learning algorithm, which takes a few repetitions of the same task to learn the waveform and amplitude of what we call the *coupling term*. Using the coupling term, we can minimize the error for a desired force of contact and, thus, also mitigate the need for tuning the scaling factor. The learning algorithm is explained in detail in Section IV.

### C. Cooperative Dynamic Movement Primitives

The force of contact with the environment can also be the force of contact with another robot and, therefore, used for bimanual or two-agent tasks. Here, it is important to note that a robot with a centralized controller and accurate control for both arms (or for two agents) does not need such modifications. Well-studied approaches for bimanual control exist, for example, [41], [42]. However, given two independently controlled robots, possibly with conflicting trajectories, an approach for motion synchronization is needed. Examples of this are cooperation of two stand-alone robots/agents working together when carrying a large object. In our experiments, we used two independently controlled robot arms for bimanual tasks.

Let us assume two separate trajectories given by two DMPs executed by two robot arms. To keep the desired force between two agents, we introduce a coupling term. For one DOF (for clarity), this coupling term is defined as

$$F_{1,2} = F_d - (F_1 - F_2) \tag{14}$$

where $F_d$ is the desired coupling force, and $F_i$ is the force measured at the end effector of the $i$th agent. In simulation or if the desired distance between the two end effectors is specified instead of the force, we introduce a virtual spring between the end effectors of the arms that alters both trajectories. In this case, the coupling term becomes

$$F_{1,2} = k(d_d - d_a) \tag{15}$$

where $d_d$ is the desired distance between the end effectors, and $d_a$ is the actual difference, while $k$ is the virtual spring constant. Measured force can be used instead of a virtual spring. The force that acts on DMP$_1$ is the opposite of the force acting on DMP$_2$:

$$F_{2,1} = -F_{1,2} = -k(d_d - d_a). \tag{16}$$

We introduce these forces, again scaled by $c$, into each DMP. Equations (17)–(22), shown below, define what we label *cooperative DMPs*:

$$\tau \dot{z}_1 = \alpha_z(\beta_z(g_1 - y_1) - z_1) + f_1(x) + c_2\dot{C}_{1,2} \tag{17}$$

$$\tau \dot{y}_1 = z_1 + C_{1,2} \tag{18}$$

$$C_{1,2} = c\,F_{1,2} \cdot l_{f1} \tag{19}$$

$$\tau \dot{z}_2 = \alpha_z(\beta_z(g_2 - y_2) - z_2) + f_2(x) + c_2\dot{C}_{2,1} \tag{20}$$

$$\tau \dot{y}_2 = z_2 + C_{2,1} \tag{21}$$

$$C_{2,1} = cF_{2,1} \cdot l_{f2}. \tag{22}$$

The variable $l_f$ defines the relation leader–follower. If $l_{f1} = l_{f2}$, then both robots will adapt their trajectories to follow average trajectories at the defined distance $d_d$ between them (within tolerance and after learning, discussed in the next section). On the other hand, if $l_{f1} = 0$ and $l_{f2} = 1$, only DMP$_2$ will change the trajectory to match the trajectory of DMP$_1$, again at the distance $d_d$ and again only after learning. The opposite applies as well. Leader–follower relation can be determined by a higher level planner, which is beyond the scope of this paper. It depends on the needs and circumstances of a specific task.

## IV. Learning Using Previous Sensory Information

To ensure the desired force of contact with the environment, or the desired displacement between two robots, we need to learn the terms $C_{1,2}$ and $C_{2,1}$ in such a way that $F_d = F$, where $F$ is either the real force or defined as in (10) or (15). In the following, we propose an ILC algorithm to learn $C_{1,2}$ and $C_{2,1}$. See a thorough review by Bristow *et al.* [27] for details on ILC. The proposed algorithm avoids the necessity to accurately model the dynamics of the robot and the environment.

In a tightly coupled bimanual task, both arms are physically connected through an object, and we can assume $C_{1,2} = -C_{2,1}$. Thus, we need to learn only one of the two terms. In the following, we denote this term by $C$. Upon the execution of the given task for the first time, the sensors register the resulting force. If the task was to be executed again with no difference, the sensory readings would not change, except for the noise.

Therefore, we propose that the second time the task is executed, the sensor measurements from the first attempt are fed into the trajectory generation in a feed-forward manner. The learning update for the coupling terms is then defined as suggested by the ILC theory [27]:

$$C_i = c\,e_i + F_{c,i} \tag{23}$$

$$F_{c,i} = Q(F_{c,i-1} + L\,c\,\dot{e}_{i-1}) \tag{24}$$

$$e_i = F_d - F_i \tag{25}$$

where index $i$ denotes the $i$th epoch, $c$ is the force gain, $e_i$ is the coupling force error calculated from the difference of the desired coupling force $F_d$ and the measured coupling force $F_i = F_{1,i} - F_{2,i}$, $F_{c,i}$ is the learned coupling force term, and $Q$ and $L$ are positive scalars. The coupling term given by (23) is known as current iteration learning control, since it incorporates instantaneous feedback in the first term and learning update in the second term. The tunable parameters are $Q$, $L$, and $c$. In all our experiments, we use $Q = 0.99$, $L = 1$, and $c = 0.5$. In the learning and subsequent execution of the learned movement, we use the coupling term $C_i$ instead of $C$ for interaction with the environment. Similarly, we use $C_i$ instead of $cF_{1,2}$ in (19) and $-C_i$ instead of $cF_{2,1}$ in (22). Note that if the desired force $F_d = 0$, (23) takes the form $C_i = -cF + F_{c,i}$, which matches (9) in the first iteration ($i = 0$), when the learned coupling force is $F_c = 0$.

While the force depends on the execution of the trajectory and thus time, there is no need to encode the learned coupling force $F_c$ as a vector of time-stamps and values. Just like $f(x)$, we represent $F_c$ as a linear combination of radial basis functions along phase $x$:

$$F_c(x) = \frac{\sum_{j=1}^{M} a_j \Psi_j(x)}{\sum_{j=1}^{M} \Psi_j(x)} x. \tag{26}$$

To calculate the weight parameters after the $i$th epoch, we use

$$\mathbf{f} = \begin{bmatrix} F_{c,i}(x_0) \\ \dots \\ F_{c,i}(x_T) \end{bmatrix}, \ \mathbf{a} = \begin{bmatrix} a_0 \\ \dots \\ a_M \end{bmatrix} \tag{27}$$

where $x_j = x(t_j)$ and $t_j$ denotes the $j$th time sample. Writing

$$\mathbf{X} = \begin{bmatrix} \dfrac{\Psi_1(x_0)}{\sum_{j=1}^{M} \Psi_j(x_0)} x_0 & \dots & \dfrac{\Psi_M(x_1)}{\sum_{j=1}^{M} \Psi_j(x_0)} x_0 \\ \dots & \dots & \dots \\ \dfrac{\Psi_1(x_T)}{\sum_{j=1}^{M} \Psi_j(x_T)} x_T & \dots & \dfrac{\Psi_M(x_T)}{\sum_{j=1}^{M} \Psi_j(x_T)} x_T \end{bmatrix} \tag{28}$$

we need to solve the following set of linear equations:

$$\mathbf{X}\mathbf{a} = \mathbf{f}. \tag{29}$$

The parameters $\mathbf{a}$ are calculated in a least-squares sense. Several advantages speak in favor of encoding the coupling term in this manner. For example, the nonlinear encoding acts as a filter [7] and thus cancels out the sensor noise. The main advantage is that this coupling term depends on the same canonical system as the trajectories.

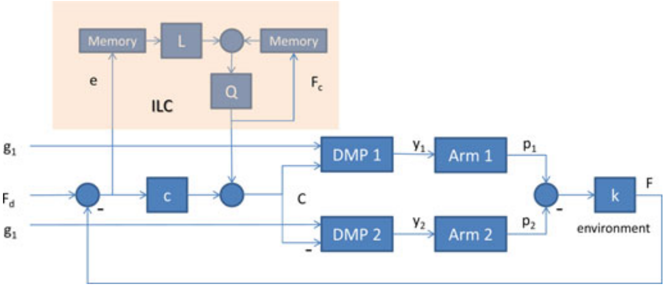

Fig. 3. MISO structure of the cooperative DMP system. Note that the coupling comes from the force $F$, which is (in simulation) defined as the scaled distance between the robots/agents, i.e., $F_d = kd_d$. The shaded region marks the ILC.

Note that separate canonical systems and, therefore, phases can be used for the predefined motion of the robot, given by $y$, and the coupling force $F_c$. At the end of the predefined motion, the phase $x$ reaches practically 0, and only the linear part of the DMP remains active. In order for $F_c$ not to go to 0 at the same time, a separate phase variable has to be kept while learning and later applying $F_c$. Since we are dealing with discrete, finite motions, eventually, both run out.

## V. Stability

### A. Stability of Coupled Dynamic Movement Primitives

Even though single DMPs are stable [10], the stability of coupled DMPs, given by (17)–(22), cannot be guaranteed without further analysis. Cooperative two robot/agent DMPs change the system from single-input–single-output (SISO) into a multiple-input–single-output (MISO) system, since their outputs are subtracted. Fig. 3 shows the resulting MISO system structure. The coupling comes from the force, which depends on the positions of the two robots as given in (15), where the actual distance is $d_a = p_1 - p_2$, with $p_1$ and $p_2$ being the positions of the two arms. In our theoretical analysis, we assume that the robot tracks the desired trajectory perfectly, i.e., $p_1 = y_1$, $p_2 = y_2$; thus, $d_a = y_1 - y_2$. For the given stable DMP parameters, the gain $c$ of the coupling term determines the behavior of the MISO system. Using the virtual spring formulation (15), we can derive the state-space system (30) and (31) from (17)–(22) with the applied feedback $C_{1,2} = -C_{2,1} = k(d_d - d_a)$, $l_{f,1} = l_{f,2} = 1$, (see the Appendix)

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{30}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t). \tag{31}$$

The system matrices for the controllable canonical form are given by

$$\mathbf{A} = \begin{bmatrix} -\dfrac{\alpha_z\tau + 2ck(c_2 + \tau)}{\tau^2} & 1 \\ -\dfrac{\alpha_z\beta_z\tau + 2ck}{\tau^2} & 0 \end{bmatrix} \tag{32}$$

$$\mathbf{B} = \begin{bmatrix} \dfrac{2k(c_2 + \tau)}{\tau^2} & \dfrac{2ck(c_2 + \tau)}{\tau^2} & 0 \\ \dfrac{2k\alpha_z}{\tau^2} & \dfrac{2ck\alpha_z}{\tau^2} & \dfrac{k}{\tau^2} \end{bmatrix} \tag{33}$$

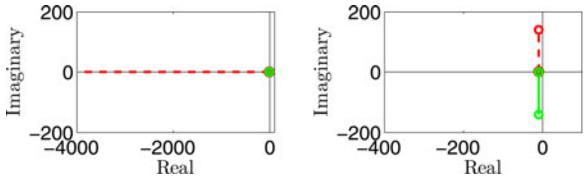

Fig. 4. Root locus plot of the coupled DMP structure with modulation at an acceleration level (right) and both the velocity and acceleration levels (left). We varied the gain $ck$ from 0 to 10 000, while the gain $c_2$ was fixed at 1. The system has two poles denoted with red (dashed) and green. The full circles denote roots at $ck = 0$ and the empty circles at $ck = 10\,000$.

$$\mathbf{C} = [1 \quad 0]. \tag{34}$$

The input vector $\mathbf{u}$ and the scalar output $y$ in (32) are given as $\mathbf{u} = [F_c, F_d, \alpha_z\beta_z(g_1 - g_2) + f_1(x) - f_2(x)]^T$ and $y = F$, respectively (see Fig. 3). The state vector is defined as

$$\mathbf{x} = \begin{bmatrix} F \\ \dot{F} - \dfrac{\alpha_z\tau + 2ck(c_2 + \tau)}{\tau^2}F \end{bmatrix}. \tag{35}$$

Since the nonlinear parts $f_1(x)$ and $f_2(x)$ in (17) and (20) are bounded and tend to zero as the phase tends to zero, it is sufficient to prove the stability and attractor properties of the linear part of system (17)–(22). We assume environment stiffness as defined by (10).

The eigenvalues of $\mathbf{A}$ determine the stability and convergence of the linear part of differential equation system (30). These eigenvalues are given as

$$\lambda_{1,2} = \frac{1}{2}\Bigg( - \frac{\alpha_z\tau + 2ck(c_2 + \tau)}{\tau^2} \pm \sqrt{\left(\frac{\alpha_z\tau + 2ck(c_2 + \tau)}{\tau^2}\right)^2 - 4\frac{\alpha_z\beta_z\tau + 2ck}{\tau^2}}\Bigg). \tag{36}$$

Since all parameters $\alpha_z, \beta_z, c, k, c_2$, and $\tau$ are positive, the eigenvalues $\lambda_{1,2}$ are negative for all cases in which both eigenvalues are real numbers. It can also happen that the eigenvalues are complex numbers, but in such cases, the real part of both eigenvalues is again negative. This means that system (30) converges to a unique attractor point for all positive parameter values. We obtain complex eigenvalues only for some rather unlikely values, e.g., for very large values of $\tau$.

We compared the performance of the proposed velocity–acceleration scheme with a scheme that uses only acceleration feedback in (11) but no feedback in (12). Root-locus in Fig. 4 shows that both schemes remain stable with increasing gain $ck$, but also clearly shows the main difference between them. The imaginary part of conjugate-complex eigenvalues increases only in the case when acceleration level modulation is used, whereas it remains close to zero when modulating both velocity and acceleration levels. The results clearly support the proposed velocity and acceleration level scheme, where the response is always damped, whereas the convergence is slower for modulation at acceleration level only.

### B. Stability and Convergence of the Learning Algorithm

Besides the stability and convergence to the attractor point of the coupled DMP structure, we also need to prove the stability and convergence of the proposed learning algorithm, as defined by (23)–(25). In the following, we rely on the stability analysis provided by the ILC framework. A general form of ILC is defined as [27]

$$F_{c,i+1}(j) = Q(F_{c,i}(j) + Le_{i+1}(j+1)) \tag{37}$$

where $F_{c,i}(j)$ is the control input of the $i$th epoch, $e_i(j) = F_d - F_i(j)$, $Q$ is the filtering of the last control input, $L$ is the learning parameter, and $j$ denotes the time sample. Equation (24) is exactly the same as (37) with the exception that this is a discrete time implementation, where the error derivative was replaced by the sample shift $j + 1$. The aim of the stability analysis is to find the range of parameters of the learning parameter $L$, where the ILC remains stable. For that, state-space matrices given by (30) and (31) have to be in a discrete time form [43]. Note that from the ILC perspective, the input is the learned coupling force $F_c$. Therefore, a suitable discrete time representation of our system is

$$\mathbf{x}(j+1) = \hat{\mathbf{A}}\mathbf{x}(j) + \hat{\mathbf{B}}_1 F_c(j) + \hat{\mathbf{B}}_2 F_d + \hat{\mathbf{B}}_3 \Phi(j) \tag{38}$$

$$y(j) = \hat{\mathbf{C}}\mathbf{x}(j) \tag{39}$$

where input matrices $\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2, \hat{\mathbf{B}}_3$ are formed from the first, second, and the third column of the input matrix $\mathbf{B}$, respectively [see (33)]. Hat symbol $(\hat{\cdot})$ denotes the discrete-time counterpart of the continuous-time system matrices. An input signal $\Phi(j)$ is defined as $\alpha_z\beta_z(g_1 - g_2) + f_1(x(j)) - f_2(x(j))$. A commonly accepted framework to examine the stability of a discrete-time plant controlled with ILC in a time domain is a lifted or a supervector representation of the system dynamics [27], [44]. The supervector representation of the discrete-time system results in $T + 1$ dimensional input and output vectors and $(T + 1) \times (T + 1)$ system matrix

$$\begin{bmatrix} F_i(0) \\ F_i(1) \\ F_i(2) \\ \vdots \\ F_i(T) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \hat{\mathbf{C}}\hat{\mathbf{B}}_1 & 0 & \cdots & 0 \\ \hat{\mathbf{C}}\hat{\mathbf{A}}\hat{\mathbf{B}}_1 & \hat{\mathbf{C}}\hat{\mathbf{B}}_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{C}}\hat{\mathbf{A}}^{T-1}\hat{\mathbf{B}}_1 & \hat{\mathbf{C}}\hat{\mathbf{A}}^{T-2}\hat{\mathbf{B}}_1 & \cdots & 0 \end{bmatrix} \begin{bmatrix} F_{c,i}(0) \\ F_{c,i}(1) \\ F_{c,i}(2) \\ \vdots \\ F_{c,i}(T) \end{bmatrix} + \begin{bmatrix} r(0) \\ r(1) \\ r(2) \\ \vdots \\ r(T) \end{bmatrix}$$

$$\begin{bmatrix} r(0) \\ r(1) \\ r(2) \\ \vdots \\ r(T) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{C}} \\ \hat{\mathbf{C}}\hat{\mathbf{A}} \\ \hat{\mathbf{C}}\hat{\mathbf{A}}^2 \\ \vdots \\ \hat{\mathbf{C}}\hat{\mathbf{A}}^T \end{bmatrix} \begin{bmatrix} F(0) \\ 0 \end{bmatrix}$$

$$+\begin{bmatrix} 0 & \cdots & 0 \\ \hat{\mathbf{C}}\hat{\mathbf{B}}_2 & \cdots & 0 \\ \hat{\mathbf{C}}\hat{\mathbf{A}}\hat{\mathbf{B}}_2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \hat{\mathbf{C}}\hat{\mathbf{A}}^{T-1}\hat{\mathbf{B}}_2 & \cdots & 0 \end{bmatrix}\begin{bmatrix} F_d \\ F_d \\ F_d \\ \vdots \\ F_d \end{bmatrix} + \begin{bmatrix} 0 & \cdots & 0 \\ \hat{\mathbf{C}}\hat{\mathbf{B}}_3 & \cdots & 0 \\ \hat{\mathbf{C}}\hat{\mathbf{A}}\hat{\mathbf{B}}_3 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \hat{\mathbf{C}}\hat{\mathbf{A}}^{T-1}\hat{\mathbf{B}}_3 & \cdots & 0 \end{bmatrix}\begin{bmatrix} \Phi(x(0)) \\ \Phi(x(1)) \\ \Phi(x(2)) \\ \vdots \\ \Phi(x(T)) \end{bmatrix} \quad (40)$$

where the contribution of initial condition $F(0)$ and inputs $F_d, \Phi(x(t))$ are treated as an exogenous signal $\mathbf{r}$, and $T$ is the number of time samples. Note that inputs $\Phi(x(t))$ are bounded because the nonlinear part of DMP [$f_1(x)$ and $f_2(x)$] is given as linear combination of radial basis functions, which are bounded. The system matrix in (40), which we denote by $\mathbf{P}$, is a lower triangular Toeplitz matrix, where the coefficients are Markov parameters [27]. With the supervector notation of the ILC matrices $\mathbf{Q}$ and $\mathbf{L}$ and by inserting (40) into (37), we obtain

$$\mathbf{F}_{c,i+1}(j) = \mathbf{Q}(\mathbf{I} - \mathbf{LP})\mathbf{F}_{c,i}(j) + \mathbf{QL}(\mathbf{F}_d - \mathbf{r}) \quad (41)$$

where $i$ denotes the iteration index of the learning controller, $\mathbf{I}$ is a diagonal matrix of dimension $(T+1) \times (T+1)$, and $\mathbf{F}_d = [F_d, \ldots, F_d]^{\mathrm{T}}$. The ILC system is asymptotically stable if and only if [27]

$$\rho(\mathbf{Q}(\mathbf{I} - \mathbf{LP})) < 1 \quad (42)$$

where $\rho$ denotes the maximum absolute value of the matrix eigenvalue. If the ILC system is asymptotically stable, the asymptotic error when $T$ tends to infinity is

$$\mathbf{e}_\infty = [\mathbf{I} - \mathbf{P}[\mathbf{I} - \mathbf{Q}(\mathbf{I} - \mathbf{LP})]^{-1}\mathbf{QL}](\mathbf{F}_d - \mathbf{r}). \quad (43)$$

For $\mathbf{Q} = \mathbf{I}$, the error $e_i(j)$ will converge to 0. Applying an ILC controller with $\mathbf{Q} = 0.99\mathbf{I}$ and $\mathbf{L} = L\mathbf{I}$ to the coupled DMP system shows that the stability can be guaranteed for the learning controller gains within the range $L = [0, 2.09]$, where $T \leq 500$, and the gain $ck$ was set to 100. Given these settings and $L = 1$, $g_1 - g_2 = 5$, $f_1(x) - f_2(x) = 0$, $F_d = 0$, the force $F$ will converge to the desired force $F_d$ with maximal error norm $0.062813N$.

In the ILC stability analysis, we assumed that given equal control signals, the plant always returns the same outputs, which cannot be always guaranteed in the case of changing environmental dynamics. A key question is therefore whether or not the proposed scheme remains asymptotically stable to plant perturbations. In [27] and [44], it was shown that the ILC is inherently stable to the plant dynamics variation. If we want to further increase the robustness to plant perturbations, the most direct way is to decrease the $\mathbf{Q}$ filter gain. On the other hand, decreasing $\mathbf{Q}$ increases the steady-state learning error. As a consequence, the selection of $\mathbf{Q}$ is a tradeoff between performance and robustness. Robustness was experimentally evaluated, and the results are presented in the next section.

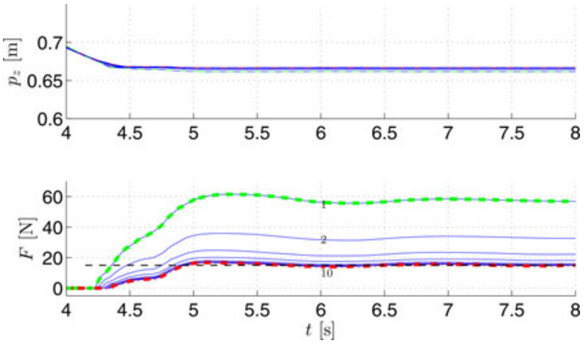

Fig. 5. Real-world results of adaptation to environment. (Top) Trajectories of motion, with the trajectory of the first epoch in green (dashed) and the final (the tenth) epoch in red (dash-dot). (Bottom) Measured forces with the force of the first epoch in green (dashed) and the last being the tenth in red (dash-dot). The black dashed line shows the desired force that is set to appear after the impact is detected.

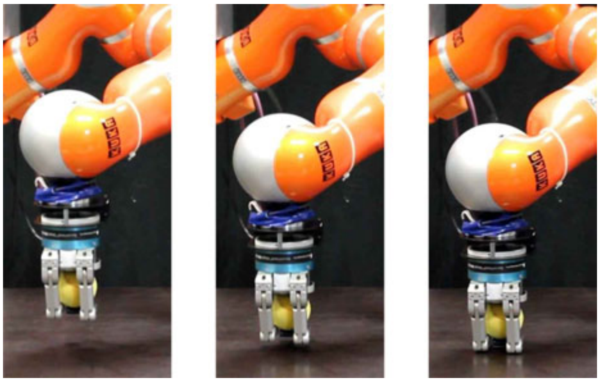

Fig. 6. Image sequence showing the collision of the robot with the table.

# VI. Evaluation

## A. Experimental Setup

We performed several simulated and real-world experiments. The real-world experiments were performed on two KUKA LWR arms with 7 DOFs each. Both arms are shown in Fig. 16. Trajectory calculation was performed on a client computer using MATLAB/Simulink. The desired task space coordinates were sent to a server computer at 200 Hz via UDP. The server computer, running an xPC Target application at 2 kHz sent these commands to the KUKA controller, utilizing KUKA Fast Research Interface. It also sent the measured actual robot positions and forces back to the client PC.

## B. Contact With the Environment

Contact with the environment is crucial for many robotic tasks. It needs to be safe for both the robot and the environment, which consequently means that the forces should be kept low. We applied the proposed algorithm to produce a desired force of contact $F = 15$ N upon impact with a table (see Fig. 6). The movement was repeated ten times. Fig. 5 shows the results of the real-world experiment. The top plot shows the trajectories of all epochs. The original trajectory was defined to start at 1.0 m and end at 0.6 m in 5 s. The table was at just under 0.67 m. The

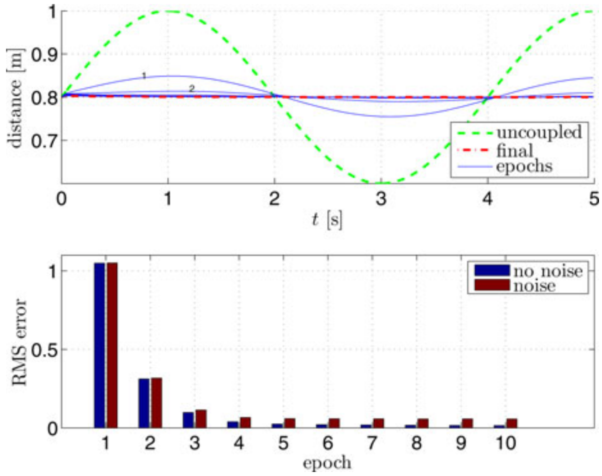

Fig. 7. Simulated results of cooperative DMPs. (Top) Distance between the robots, where the green (dashed) line marks the distance during the trajectory of uncoupled robots, the red (dash-dot) line the distance after ten epochs, and the blue lines the distance in each epoch; epochs 1 and 2 are marked with numbers. (Bottom) RMS error after each epoch for trajectories with and without noise.

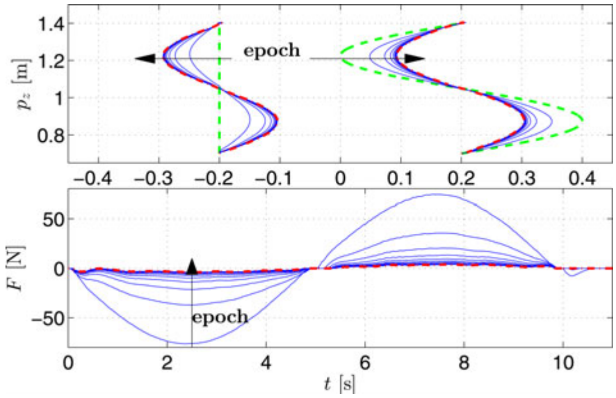

Fig. 8. Real-world experimental results for cooperative DMPs. Note that the $p_{y,L}$ and $p_{y,R}$ trajectories are for presentation purposes depicted at ±0.2 m, although the trajectories were performed at ±0.4 m. The top $p_y$–$p_z$ plot shows the trajectories, with the green dashed lines showing the original trajectories and the red dash-dot lines showing the final trajectories. The bottom plot shows the resulting force and the final as a red dash-dot line.

trajectory in the first epoch is in green (dashed) and the trajectory after learning in red (dash-dot). While the position trajectories practically overlap, the initial and final force trajectories are considerably different. Note that here it is crucial that the DMP was modulated with the measured force already in the first epoch; otherwise, the resulting forces would be far greater and could damage the robot. The bottom plot shows the forces, with the force of the first epoch in green and the final force after 10 epochs in red. The reference force is set to appear after the impact is detected. Note that this is later anticipated by the ILC and that the trajectory is altered before the actual contact.

### *C. Bimanual Tasks*

We applied the proposed approach to couple two trajectories in simulation. For the left robot, the original trajectory was equal to $p_{x,L} = 0.75$ m, $p_{y,L} = 0.4 + 0.2\sin(t\pi/2)$ m, and $p_{z,L} = (0.7 + 0.12t)$ m. For the right robot, the trajectory was defined as $p_{x,R} = 0.75$ m, $p_{y,R} = 0.4$ m, and $p_{z,R} = (0.7 + 0.14t)$ m. The desired distance between the robots was set to $d_d = 0.8$ m, and the virtual coupling spring constant to 20 N/m. The top plot in Fig. 7 shows the distance between the robots along the trajectories for each of the ten learning epochs. It is evident that the error is considerably reduced after very few epochs; epochs 1 and 2 are marked. The final distance along the trajectory between the robots, marked with red, has a maximal error of less than 0.003 m, appearing at the very start of the motion.

The bottom plot shows the root-mean-square (RMS) error of the distance for the cases of ideal trajectories (no noise) and for the case when noise of distance estimation was added to show the robustness. The noise was added on the position of the arms, with the maximal noise amplitude at 0.01 m. We can see that the approach is hardly affected by the noise.

We performed a similar experiment on the real robots, which we tightly coupled by both of them rigidly holding a stick. The motions were the same as for the virtual experiment with the difference in the $p_z$ direction, which are defined as the same for both robots at $p_{z,L} = p_{z,R} = (0.7 + 0.07t)$ m. The duration of motion was set to 10 s. A full sinusoidal wave was performed by the left robot with $p_{y,L} = 0.4 + 0.2\sin(t\pi/5)$ m. The task was to modify the trajectories so that the force along the stick on the robots will be minimal. Seven learning epochs were conducted. Fig. 8 shows the results. The top plot shows the trajectories in a $p_y - p_z$ plot. The green dashed lines show the original uncoupled trajectories and the red dash-dot lines the final trajectories. The bottom plot shows the resulting measured force. Both robots adapted, i.e., $l_{f1} = l_{f2} = 1$. The force scaling factor was set empirically.

The two image sequences shown in Fig. 9 compare the execution of independent and cooperative DMPs, where cooperative DMPs were learned in seven epochs. The top row shows the execution of the original, independent trajectories, where we can see that the distance between the robot end effectors is changing, which can be observed from the length of the stick on the right side of the right robot. The bottom row shows the execution of cooperative trajectories, where the distance between the robots is kept constant. Fig. 8 shows the resulting forces along the stick.

In another experiment, we coupled two independently controlled KUKA LWR robots and combined the task with adaptation to external, human interference, which is not fully repeatable due to the human in the loop. The task demanded that the robots—together with the human—place a lid on a wooden box, the robots holding one side and the human another side. The fit of the lid was very tight. The initial trajectories were learned by demonstration. Then, the box was moved 12 cm in the $-p_x$ (backward) and the 7 cm in $p_y$ (robot's right) directions from the demonstrated position, and our proposed approach was used to correct for the misplacement of the box in eight epochs. $p_x$ and $p_y$ directions were corrected, while $p_z$ (up-down) was not. The task was to minimize the force the robots exert to each other and toward the human, i.e., $F_{d,x,y} = 0$ for both robots. Fig. 10 shows the resulting forces and positions of both robots.

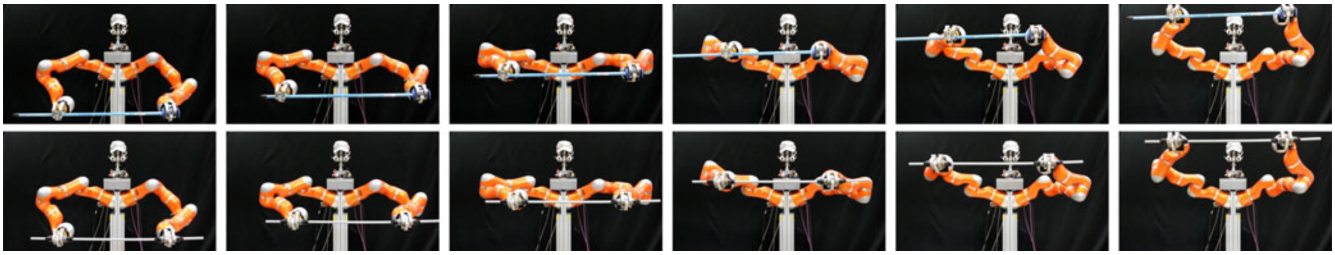

Fig. 9. Execution of a task as performed by two independent DMPs (top row) and the execution with cooperative DMPS after learning (bottom row).

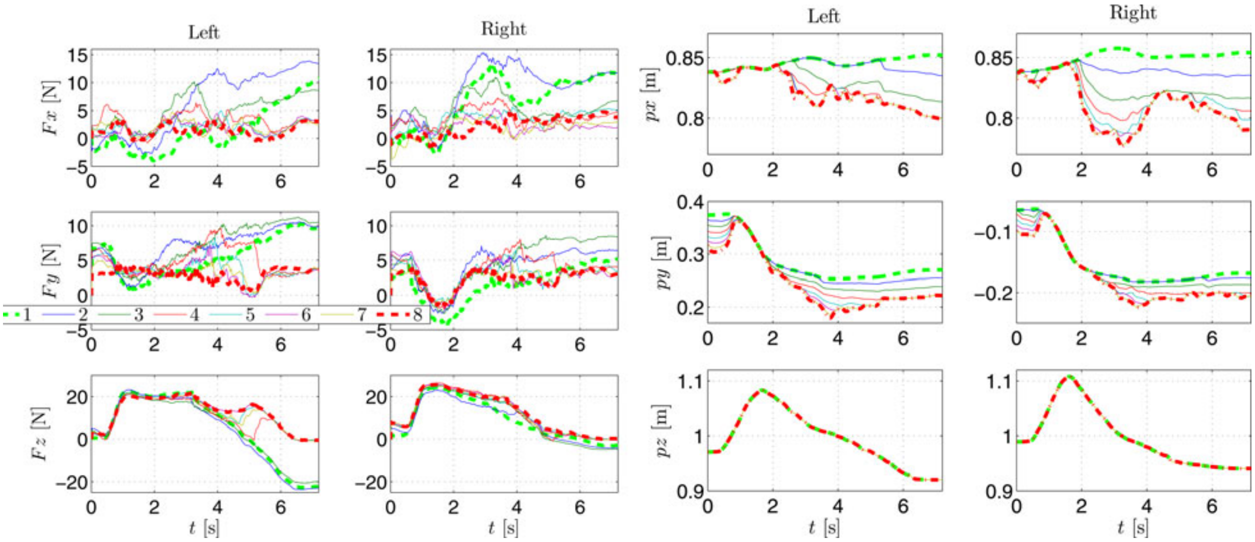

Fig. 10. Results of adaptation of force in both robots in the left figure and of position in the right figure. The legend denotes the colors of separate epochs for all 12 plots. The first is marked with green dashed and the last with red dash-dot line.

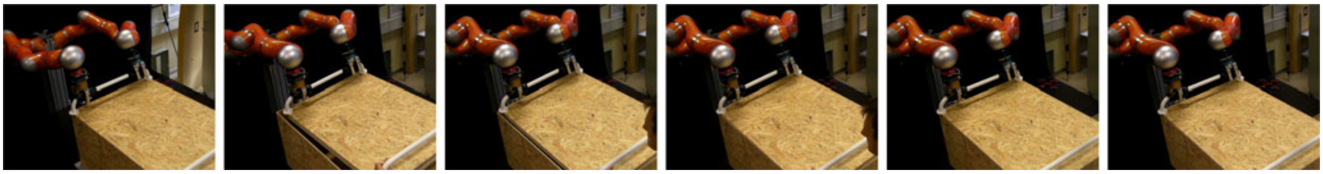

Fig. 11. Position of the box, lid, and the robots in the demonstration in the leftmost picture. Positions after each epoch, with epoch number increasing toward the right. Only five epochs are shown because the pictures of the final position after epoch five are practically identical.

We can see from the force plots in Fig. 10 that the person had to push on the lid in the first three epochs (see $F_y$ plots). We can also see in the left $F_z$ plot that the lid did not fit in the box in the first four epochs. Once it did, the forces were reduced to desired values at 0 N. Negative $F_z$ is the force the human is exerting on the lid, while positive values indicate simply the weight of the lid. Fig. 11 shows the position of the lid after the demonstration and after the first five epochs. The experiment is also depicted in the accompanying video.

### D. *Effect of Changes in Epochs*

In a real-world scenario with two independent agents performing a cooperative task, a systematic error of the system could influence the repeatability of the execution during epochs. Examples of such could be the slipping of the wheels of wheeled robots cooperatively moving an object, faulty sensors, miscalibration . . . In the experiment two robots are coupled to maintain a common distance of 0.4 m. The original trajectory of the right robot is a straight vertical line, and of the left robot a sinusoidal line. The left robot is the leader and does not adapt trajectories. It also drifts in the first five epochs ($i = 1, \ldots, 5$), which we simulate with $y_L(t) = 0.4 + \sin(t\pi/2)0.2 + (i-1)0.005t + (i-1)\sin(t\pi/2)0.002$. After the fifth epoch, the drifting stops. We also simulate noise.

Fig. 12 shows the trajectories of both robots, with original trajectories in dashed green and final trajectories in dashed red. Adaptation through all epochs can be clearly seen. The final trajectories practically maintain the desired distance at all times, as can be seen in the RMS error results in Fig. 13, which also depicts RMS errors for using only a feedback controller for coupling, while the leader robot drifts, i.e., in the first five epochs. The results clearly show the advantage of using ILC, even if the trajectories are smoothly changing from trial to trial.

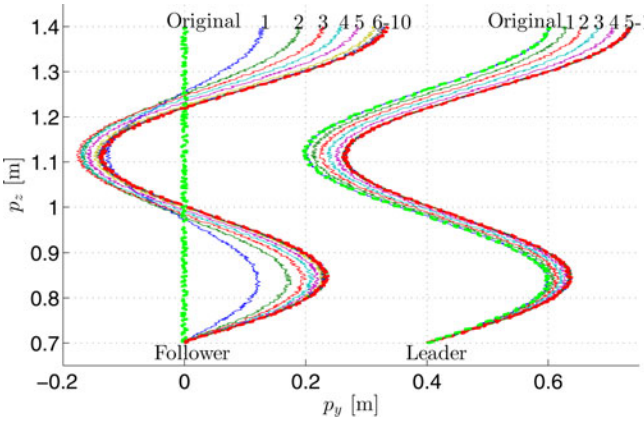

Fig. 12. Simulated results for cooperative DMPs. The leader robot does not adapt its trajectories. It also drifts in the first five epochs. The follower robot, which starts from a straight trajectory, adapts in every epoch and converges to the final trajectory after the drift of the leader robot stops. The RMS error for the follower robot trajectories is shown in Fig. 13.

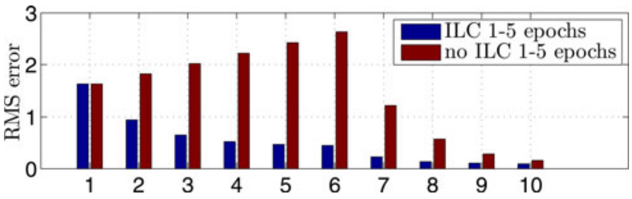

Fig. 13. RMS error for simulated trajectories from Fig 12. Despite the drift, the RMS error is being clearly reduced. The RMS error is far bigger if only a feedback controller (no ILC) is used.

## E. *Obstacle Avoidance in Bimanual Tasks*

When one of the arms in a bimanual task encounters an obstacle, both arms have to adapt. Similar experiments but with acceleration level coupling and Hebbian-type learning of a filter gain were discussed in [39]. If the arms are coupled using our proposed approach, the feedback term will move the arm that does not encounter an obstacle, but considerable force will appear between the tightly coupled arms. If the obstacle is repeatable over epochs, the proposed ILC approach can be used to minimize the force. In our experiments, we applied the same online obstacle avoidance algorithm, as presented in [10] and [25]. We used it to control the left robot arm.

The cooperative DMPs were set in a leader–follower relation, the left robot being the leader ($l_{f,L} = 0$). From the start, both robots had identical 20 s vertical trajectories ($y_{L,R} = \pm 0.4$ m), but the left robot encountered an obstacle at $p_x = 0.7$ m, $p_y = 0.45$ m, and $p_z = 1$ m and, therefore, had to apply obstacle avoidance. Our proposed algorithm was utilized to minimize the forces between the robots, rigidly connected with a stick. Fig. 14 shows the results of learning to minimize the forces between the robots in seven epochs. The top plot shows the $p_y - p_z$ trajectory plot. The trajectories are for presentation purposes again depicted at $y_{L,R} \pm 0.2$ m, but they were executed at $y_{L,R} = \pm 0.4$ m. The dashed green lines show the original trajectories. The black lines connecting the robots show the connecting stick every 5 s. The bottom plot shows the resulting forces between the robots, in the $p_y$ direction (blue), and the resulting torques around the global $z$-(vertical) axis.

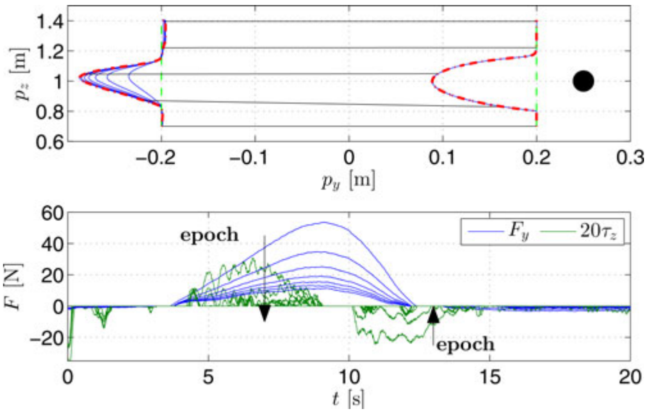

Fig. 14. Real-world results of obstacle avoidance with the right robot following the left one. The top plot shows the $p_y$–$p_z$ trajectories, where the original trajectories are depicted with green dashed lines. The bottom plot shows the resulting forces and the resulting torques (scaled 20 times for presentation purposes).

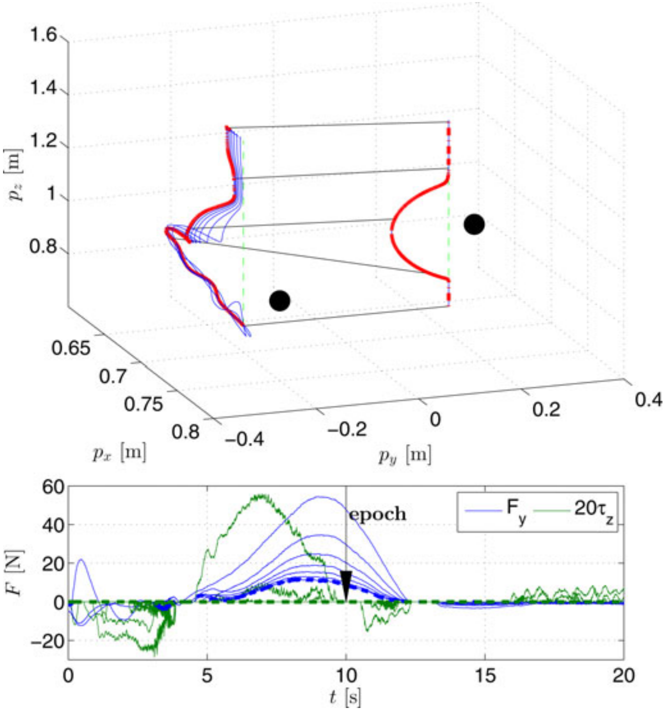

Fig. 15. Real-world results of double obstacle avoidance of cooperative DMP trajectories. The left robot is the leader, which encounters an obstacle, but the follower also encounters an obstacle. The top plot shows the final, i.e., the 7th plot marked with a red dash-dot line. Dashed green lines show the original trajectories. The bottom plot shows the resulting force in the $p_y$-direction and the resulting torque (scaled 20 times for presentation purposes) around the world $z$-axis. Dashed lines show the forces and torques in the final epoch.

Fig. 15 shows the results of a similar real-world experiment, where additionally the right robot encounters an obstacle at $p_x = 0.75$ m, $p_y = -0.4$ m, and $p_z = 0.9$ m. The obstacle is set so that the robot must avoid it in the $-x$ direction. The resulting movement leads to a rotation of the stick between the robots, namely, around the world $z$-axis. The rotation of the

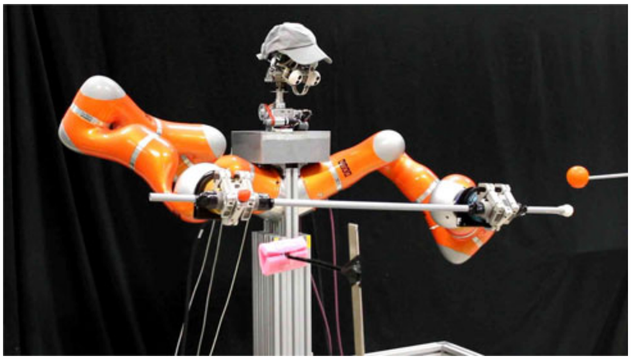

Fig. 16. Simultaneous avoidance of two obstacles. The leader robot arm (left) encounters an obstacle to its left side (orange ball). Before that, the follower robot encounters an obstacle in front of itself (pink foam).

object was a direct result of cooperation, and no higher level planners were applied. The results indicate the ability of the algorithm to provide trajectories that can guide wide objects through narrow passages, e.g., a long board through a door, with no higher level planning. Note that the coupling between the robots was in all task space degrees of freedom. In the top 3-D plot,we can also notice the initial oscillations. These are the result of both obstacle avoidance and cooperative terms acting on the trajectory of the right robot. The oscillations disappear by the final, seventh epoch, marked with red. The bottom plot shows the resulting forces in the $p_y$-direction and the resulting torque around the world $z$-axis. Fig. 16 shows the two robots avoiding two obstacles. Note that in a simulated scenario, the robots could get stuck in a local minimum, where both the obstacle avoidance and the coupling terms would provide the same, excluding modulation values. In a real-world scenario that is unlikely, even more so due to the fact that the coupling acts on both the velocity and acceleration.

## VII. Modulating the Duration of Coupled Trajectories

The introduction of the coupling term does not affect other DMP modulation properties, as was already demonstrated with obstacle avoidance. In the following, we show how we can modulate the duration of coupled DMPs.

The property of not being directly dependent on time but on the phase $x$ of the movement allows the modulation of a DMP trajectory duration by changing a single parameter, i.e., parameter $\tau$. Coupled DMPs preserve this property if a simple scaling factor is added to (11) or (17) and (20), respectively.

For example, if $\tau_{\mathrm{new}} = 2\tau$ and the rest of the DMP parameters remain unchanged, the new DMP trajectory will take twice as long to execute. Other than duration, given correct initial conditions (position and velocity), the trajectory will remain unchanged. The coupling term $F_c$, which couples the trajectory to the environment or another robot, also depends on the phase $x$; see (26). Let us assume that the same $\tau$ governs the duration of both the trajectory and the coupling term $F_c$. By changing the duration of the coupled trajectory with $\tau_{\mathrm{new}} = 2\tau$, the behavior of the robot will remain the same, even though $F_c$ was learned for $\tau$, with the only change in (11), where the term $c_2\dot{C}$ must be changed to $\frac{\tau_{\mathrm{new}}}{\tau}c_2\dot{C}$.

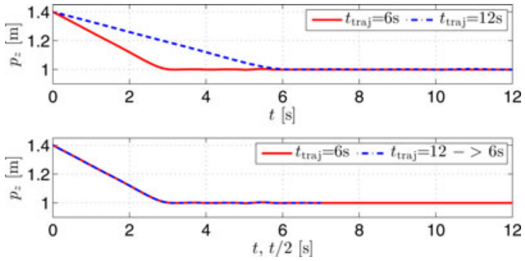

Fig. 17. Simulation results obtained by modulating the duration of an environment-coupled DMP. The original trajectory and the trajectory with the changed duration are in the top plot. Both trajectories, where the modulated trajectory time scale is changed with the same ratio as the duration, are presented in the bottom plot.

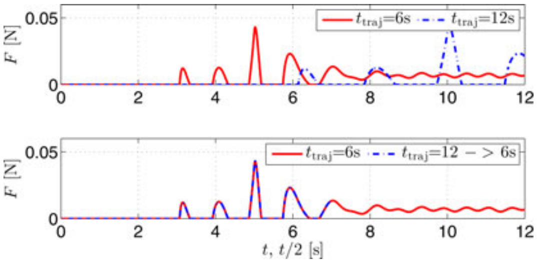

Fig. 18. Simulation force results obtained by modulating the duration of an environment-coupled DMP. The measured force during the execution of the original trajectory and the force measured along the trajectory with the changed duration are in the top plot. Both forces, where the modulated trajectory time scale is changed with the same ratio as the duration, are presented in the bottom plot.

Fig. 17 shows the results of modulating $\tau$ for the case of interaction with the environment. A DMP trajectory was encoded to reach from $p_z = 1.4$ m to $p_z = 0.7$ m, and the coupling term $F_c(x)$ learned to stop at the obstacle at $p_z = 1$ m but not press on it ($F_d = 0$). The top plot shows the trajectory for $\tau = 6$ in red, and the temporally modulated trajectory for $\tau_{\mathrm{new}} = 2\tau = 12$ in dashed blue. The bottom plot shows the same trajectories, but the modulated (dashed blue) trajectory is plotted against $t/2$, i.e., the time axis is squeezed. The trajectories match perfectly. The same initial position and velocity conditions were applied for both trajectories. Fig. 18 shows the force results for the same scenario, also showing a perfect matching.

## VIII. Discussion

In this paper, we have shown that the proposed approach can effectively be utilized to achieve the desired force contact behavior for both interaction with the environment and cooperative/bimanual tasks. Because it generates an internal environment model, i.e., learns the predictive coupling term to achieve the desired behavior, it can prevent hard contacts with the environment, which can arise during pure feedback control. The robot learns to anticipate when a contact will occur and prepares appropriately. The sensory feedback is always present in the system and assures that the robot gradually adapts to a different configuration, should the need arise. In the following, we discuss some issues of the proposed method and compare it with similar approaches in the literature. We also briefly discuss nonstationary conditions.

The current-iteration ILC algorithm requires two important tunable parameters, namely $Q$ and $L$. The force gain parameter $c$ determines the responsiveness of the system in the first iteration when the coupling term $F_c$ equals 0. Parameters were tuned using *heuristic* approach [45], where the choice of $Q$ is a tradeoff between the stability region and steady-state error. $L$ is calculated according to (42). As stated in the literature [27], a combination of ILC and feedback controller can also be applied to reject noise and nonrepeating disturbances. We showed in Figs. 7, 12, and 13 that the approach is beneficial in the presence of both noise and systematic errors.

Considering that our approach can modify any trajectory to achieve the desired behavior, the approach is in its essence general. While it is true that it requires a few repetitions of the task to learn the behavior, the same can be claimed for any learning scenario. Since many objects in a human home, such as furniture, are stationary, the learning process needs to take place only once for a specific task. The feedback controller and continuous learning over the epochs can account for changes in the environment. When considering the algorithm for nonstationary conditions, for example, when operating in contact with a human or when dealing with moving objects, the proposed algorithm is also applicable, as we have shown in our experiments.

Contrary to our approach, which changes the reference trajectory to achieve the desired interaction dynamics, Cheah and Wang [46] showed how to apply ILC to learn the target impedance model. To improve the stability of interaction, Yang *et al.* [47] moved beyond standard ILC and proposed a learning controller for interaction tasks by adapting feed-forward force and impedance. The advantage of changing reference trajectories—like in our study—is that we can anticipate contacts before they arise. We have shown in our experiments that the proposed approach can cope with gradual changes in the environment.

With respect to similar algorithms in the literature, the algorithm in [38] uses a similar setting. A trajectory is executed once and the measured signals are used as referential signals for the controller. This is also the fundamental difference to our approach, as the first execution gives the reference, while in our case, the reference can be anything, applied to any trajectory. While the papers [4] and [38] show impressive disturbance rejection results, true generalization remains subject to the first, successful, and referential execution.

In the approach by Pastor *et al.* [38], the output of the controller is fed into the acceleration level of the DMP to generate an improved movement. Notably, the measured force is equivalent to acceleration, and therefore, it makes sense to couple the DMP at the acceleration level. However, we have shown that using only the acceleration level of the DMP for coupling results in greater oscillations in the direction of the coupling force. As we can see from the root-locus plot of the coupled system (see Fig. 4), coupling at the velocity level results in better damping of the system. In view of the proposed ILC algorithm, this has an effect on the stability of learning. If only a feedback controller at the velocity level is used, a certain error is expected to appear in case of changed conditions every time, even if the error is repeatable in consecutive motions. This is demonstrated in the obstacle avoidance task, as shown in Fig. 14. Since the original trajectories are perfectly parallel, the first epoch, while $F_c = 0$, can be considered the same as the approach by Pastor *et al.* [4] but with coupling at both the velocity and acceleration levels. If no learning was present, the measured force between the robots would remain the same throughout the epochs. Reducing the force in consecutive executions is the real advantage of the proposed algorithm.

Kulvicius *et al.* [39] proposed to couple DMPs at the acceleration level. In this paper, we demonstrated that it is beneficial to couple DMPs at velocity and acceleration level. Their approach uses a modified DMP representation, which is explicitly dependent on time. In our study, we keep DMPs phase dependent, which allowed us to implement velocity scaling (see Section VII). Most importantly, instead of learning a predictive term (26), Kulvicius *et a*l. [39] applies Hebbian-type learning to determine a filter gain, which uses coupling force error as the input. Such a formulation cannot learn to anticipate coupling forces across learning epochs. Using the well-defined ILC framework from control theory, we were also able to prove the convergence and stability of the proposed scheme with no linearization assumptions.

## IX. Conclusion

In this paper, we presented a new approach for learning coupling terms for interactive and cooperative DMPs. Introducing force feedback into the well-defined framework of the DMPs is one of the key advantages of the proposed approach. It enables learning of coupling terms that establish desired contact forces with the environment and the adaptation of trajectories for cooperative task execution, essentially bridging the gap from the purely kinematic domain of the DMPs to dynamic behavior.

We have shown that both the coupling and the learning algorithms are stable; that it is important that the coupling terms are added at the appropriate level, i.e., velocity and acceleration; that it is robust to noise and systematic errors; and that it can be applied to use real force feedback. The latter was demonstrated in a number of simulation and real-world experiments, where the approach was applied to actual interaction and bimanual cooperation tasks, including cooperation with a human. The low number of learning epochs also makes online learning of the coupled/interactive trajectories a viable possibility.

## Appendix

In the following, we will derive state-space representation for the coupled DMP system. By writing (17) and (18) and (20) and (21) as second-order equations, we get

$$\ddot{y}_1 + \frac{\alpha_z}{\tau}\dot{y}_1 + \frac{\alpha_z\beta_z}{\tau^2}y_1 = \frac{c_2+\tau}{\tau^2}\dot{C}_{1,2} + \frac{\alpha_z}{\tau^2}C_{1,2} + \frac{\alpha_z\beta_z}{\tau^2}g_1 + \frac{f_1(x)}{\tau^2} \tag{44}$$

$$\ddot{y_2} + \frac{\alpha_z}{\tau}\dot{y_2} + \frac{\alpha_z\beta_z}{\tau^2}y_2 = \frac{c_2+\tau}{\tau^2}\dot{C}_{2,1} + \frac{\alpha_z}{\tau^2}C_{2,1} + \frac{\alpha_z\beta_z}{\tau^2}g_2 + \frac{f_2(x)}{\tau^2}. \tag{45}$$

Applying the Laplace transform [43] to both differential equations yields

$$(s^2 + a_1 s + a_2)\mathcal{Y}_1 = (b_1 s + b_2)\mathcal{C}_{1,2} + b_3\mathcal{X}_1 \tag{46}$$

$$(s^2 + a_1 s + a_2)\mathcal{Y}_2 = (b_1 s + b_2)\mathcal{C}_{2,1} + b_3\mathcal{X}_2 \tag{47}$$

where signals are now Laplace transform of the continuous time signals assuming zero initial conditions[1] ($\mathcal{Y} = \mathcal{L}(y), \mathcal{C} = \mathcal{L}(C), \mathcal{F} = \mathcal{L}(F), \mathcal{X}_i = \mathcal{L}(\alpha_z\beta_z g_i + f_i(x))$), and parameters $a_1, a_2, b_1, b_2, b_3$ are

$$a_1 = \frac{\alpha_z}{\tau},\ a_2 = \frac{\alpha_z\beta_z}{\tau^2}$$

$$b_1 = \frac{c_2+\tau}{\tau^2},\ b_2 = \frac{\alpha_z}{\tau^2},\ b_3 = \frac{1}{\tau^2}.$$

Let us rewrite both outputs in transfer function notation

$$\mathcal{Y}_1 = H_1\mathcal{C}_{1,2} + H_2\mathcal{X}_1 \tag{48}$$

$$\mathcal{Y}_2 = H_1\mathcal{C}_{2,1} + H_2\mathcal{X}_3 \tag{49}$$

where $H_1 = (b_1 s + b_2)/(s^2 + a_1 s + a_2)$, and $H_2 = b_3/(s^2 + a_1 s + a_2)$. Subtracting (49) from (48) and multiplying the result by $k$, we obtain

$$\mathcal{F} = k(\mathcal{Y}_1 - \mathcal{Y}_2) = k(H_1(\mathcal{C}_{1,2} - \mathcal{C}_{2,1}) + H_2(\mathcal{X}_1 - \mathcal{X}_2)). \tag{50}$$

Now we can apply feedback. First, we subtract (19) and (22), assume $l_{f1} = l_{f2} = 1$, and use (23) and (25) to get the relation

$$\mathcal{C}_{1,2} - \mathcal{C}_{2,1} = 2(\mathcal{F}_c + c\mathcal{F}_{1,2}) = 2\mathcal{F}_c + 2c(\mathcal{F}_d - \mathcal{F}) \tag{51}$$

and insert the result into (50):

$$\mathcal{F} = 2kH_1\mathcal{F}_c + 2ckH_1(\mathcal{F}_d - \mathcal{F}) + kH_2(\mathcal{X}_1 - \mathcal{X}_2). \tag{52}$$

Solving for $\mathcal{F}$ yields

$$\mathcal{F} = \frac{2kH_1}{1+2ckH_1}\mathcal{F}_c + \frac{2ckH_1}{1+2ckH_1}\mathcal{F}_d + \frac{kH_2}{1+2ckH_1}(\mathcal{X}_1 - \mathcal{X}_2). \tag{53}$$

The output force is therefore the sum of three transfer functions multiplied with inputs $\mathcal{F}_c, \mathcal{F}_d$, and $\mathcal{X}_1 - \mathcal{X}_2$. By inserting the definitions of $H_1$ and $H_2$ back into (54), we obtain

$$\mathcal{F} = \frac{2k(b_1 s + b_2)}{s^2 + (a_1 + 2ckb_1)s + (a_2 + 2ckb_2)}\mathcal{F}_c + \frac{2ck(b_1 s + b_2)}{s^2 + (a_1 + 2ckb_1)s + (a_2 + 2ckb_2)}\mathcal{F}_d + \frac{k}{s^2 + (a_1 + 2ckb_1)s + (a_2 + 2ckb_2)}(\mathcal{X}_1 - \mathcal{X}_2). \tag{54}$$

Finally, we substitute $a_1, a_2, b_1, b_2, b_3$ and rewrite the transfer function (54) in a controllable canonical state-space form and introduce initial conditions

$$\mathbf{A} = \begin{bmatrix} -\frac{\alpha_z\tau + 2ck(c_2+\tau)}{\tau^2} & 1 \\ -\frac{\alpha_z\beta_z\tau + 2ck}{\tau^2} & 0 \end{bmatrix} \tag{55}$$

$$\mathbf{B} = \begin{bmatrix} \frac{2k(c_2+\tau)}{\tau^2} & \frac{2ck(c_2+\tau)}{\tau^2} & 0 \\ \frac{2k\alpha_z}{\tau^2} & \frac{2ck\alpha_z}{\tau^2} & \frac{k}{\tau^2} \end{bmatrix} \tag{56}$$

$$\mathbf{C} = [1 \quad 0] \tag{57}$$

$$\mathbf{x}(0) = \begin{bmatrix} F(0) \\ \dot{F}(0) - \frac{\alpha_z\tau + 2ck(c_2+\tau)}{\tau^2}F(0) \end{bmatrix} \tag{58}$$

$$F(0) = k(y_1(0) - y_2(0)). \tag{59}$$

[1] We achieve this by subtracting the initial values from the corresponding signals.

## References

[1] O. Khatib, K. Yokoi, O. Brock, K. Chang, and A. Casal, "Robots in human environments," in *Proc. 1st Workshop Robot Motion Control*, 1999, pp. 213–221.

[2] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2006, pp. 169–175.

[3] R. B. Rusu, B. Gerkey, and M. Beetz, "Robots in the kitchen: Exploiting ubiquitous sensing and actuation," *Robot. Auton. Syst.*, vol. 56, no. 10, pp. 844–856, 2008.

[4] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, 2011, pp. 365–371.

[5] 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.

[6] 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.

[7] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarčič, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Auton. Robots*, vol. 27, no. 1, pp. 3–23, 2009.

[8] A. Ude, C. G. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robot. Auton. Syst.*, vol. 47, no. 2–3, pp. 93–108, 2004.

[9] Y. Wada and M. Kawato, "A via-point time optimization algorithm for complex sequential trajectory formation," *Neural Netw.*, vol. 17, no. 3, pp. 353–364, 2004.

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

[11] S. Calinon, F. D'halluin, E. L. Sauser, D. G. Caldwell, and A. G. Billard, "Learning and reproduction of gestures by imitation: An approach based on hidden Markov model and Gaussian mixture regression," *IEEE Robot. Autom. Mag.*, vol. 17, no. 2, pp. 44–54, Jun. 2010.

[12] S. Khansari-Zadeh and A. Billard, "Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Taipei, Taiwan, 2010, pp. 2676–2683.

[13] E. Gribovskaya, S. Khansari-Zadeh, and A. Billard, "Learning non-linear multivariate dynamics of motion in robotic manipulators," *Int. J. Robot. Res.*, vol. 30, no. 1, pp. 80–117, 2011.

[14] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, "Embodied symbol emergence based on mimesis theory," *Int. J. Robot. Res.*, vol. 23, no. 4–5, pp. 363–377, 2004.

[15] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Washington, DC, USA, 2002, vol. 2, pp. 1398–1403.

[16] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Netw.*, vol. 21, pp. 682–697, 2008.

[17] J. Kober and J. Peters, "Learning motor primitives for robotics," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 2112–2118.

[18] F. Stulp, E. A. Theodorou, and S. Schaal, "Reinforcement learning with sequences of motion primitives for robust manipulation," *IEEE Trans. Robot.*, vol. 28, no. 6, pp. 1360–1370, Dec. 2012.

[19] M. Tamosiunaite, B. Nemec, A. Ude, and F. Wörgötter, "Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives," *Robot. Auton. Syst.*, vol. 59, no. 11, pp. 910–922, 2011.

[20] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Auton. Robots*, vol. 33, pp. 361–379, 2012.

[21] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Learning parametric dynamic movement primitives from multiple demonstrations," *Neural Netw.*, vol. 24, no. 5, pp. 493–500, 2011.

[22] B. Nemec and A. Ude, "Action sequencing using dynamic movement primitives," *Robotica*, vol. 30, no. 5, pp. 837–846, 2012.

[23] T. Kulvicius, K. Ning, M. Tamosiunaite, and F. Wörgötter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 145–157, Feb. 2012.

[24] 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.

[25] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 2587–2592.

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

[27] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Syst. Mag.*, vol. 26, no. 3, pp. 96–114, Jun. 2006.

[28] A. Gams, B. Nemec, L. Žlajpah, M. Waechter, A. Ijspeert, T. Asfour, and A. Ude, "Modulation of motor primitives using force feedback: Interaction with the environment and bimanual tasks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 5629–5635.

[29] G. Heinzinger, D. Fenwick, B. Paden, and F. Miyazaki, "Stability of learning control with disturbances and uncertain initial conditions," *IEEE Trans. Automat. Control*, vol. 37, no. 1, pp. 110–114, Jan. 1992.

[30] J. Wallén, M. Norrlöf, and S. Gunnarsson, "Arm-side evaluation of ILC applied to a six-degrees-of-freedom industrial robot," in *Proc. 17th World Congr. Int. Federat. Automat. Contr.*, 2008, pp. 13 450–13 455.

[31] A. Tayebi and S. Islam, "Adaptive iterative learning control for robot manipulators: Experimental results," *Control Eng. Pract.*, vol. 14, no. 7, pp. 843–851, 2006.

[32] H.-S. Ahn, Y.-Q. Chen, and K. Moore, "Iterative learning control: Brief survey and categorization," *IEEE Trans. Syst., Man, Cybern. C: Appl. Rev.*, vol. 37, no. 6, pp. 1099–1121, Nov. 2007.

[33] L. Villani and J. De Schutter, "Force control," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. New York, NY, USA: Springer-Verlag, 2008.

[34] N. Hogan, "Impedance control: An approach to manipulation I II III," *J. Dyn. Syst., Meas. Control*, vol. 107, no. 1, pp. 1–24, 1985.

[35] A. Gams, M. Do, A. Ude, T. Asfour, and R. Dillmann, "On-line periodic movement and force-profile learning for adaptation to new surfaces," in *Proc. 10th IEEE-RAS Int. Conf. Humanoid Robots*, Nashville, TN, USA, 2010, pp. 560–565.

[36] J. Ernesti, L. Righetti, M. Do, T. Asfour, and S. Schaal, "Encoding of periodic and their transient motions by a single dynamic movement primitive," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2012, pp. 57–64.

[37] V. Koropouli, D. Lee, and S. Hirche, "Learning interaction control policies by demonstration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 344–349.

[38] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, "Towards associative skill memories," in *Proc. 12th IEEE-RAS Int. Conf. Humanoid Robots*, 2012, pp. 309–315.

[39] T. Kulvicius, M. Biehl, M. J. Aein, M. Tamosiunaite, and F. Wörgötter, "Interaction learning for dynamic movement primitives used in cooperative robotic tasks," *Robot. Auton. Syst.*, vol. 61, pp. 1450–1459, 2013.

[40] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *Proc. 12th IEEE-RAS Int. Conf. Humanoid Robots*, 2012, pp. 323–329.

[41] P. Chiacchio, S. Chiaverini, and B. Siciliano, "Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system," *J. Dyn. Sys., Meas., Control*, vol. 118, no. 4, pp. 691–697, 1996.

[42] N. Likar, B. Nemec, and L. Žlajpah, "Virtual mechanism approach for dual-arm manipulation," in *Robotica*, 2013, DOI: http://dx.doi.org/10.1017/S0263574713000763

[43] M. Gopal, *Modern Control System Theory*. New York, NY, USA: Wiley, 1993.

[44] K. Moore, Y. Chen, and H.-S. Ahn, "Iterative learning control: A tutorial and big picture view," in *Proc. 45th IEEE Conf. Decis. Contr.*, San Diego, CA, USA, 2006, pp. 2352–2357.

[45] M. Norrlöf and S. Gunnarsson, "Experimental comparison of some classical iterative learning control algorithms," *IEEE Trans. Robot. Autom.*, vol. 8, no. 4, pp. 636–641, Aug. 2002.

[46] C.-C. Cheah and D. Wang, "Learning impedance control for robotic manipulators," *IEEE Trans. Robot. Autom.*, vol. 14, no. 3, pp. 452–465, Jun. 1998.

[47] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schäffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions," *IEEE Trans. Robot.*, vol. 27, no. 5, pp. 918–930, Oct. 2011.

**Andrej Gams** received the Diploma degree in electrical engineering in 2004, and the Ph.D. degree in robotics from the University of Ljubljana, Slovenia, in 2009.

He is currently a Research Fellow with the Department of Automatics, Biocybernetics, and Robotics, Jožef Stefan Institute, Ljubljana. He was a Postdoctoral Researcher with the Biorobotics Laboratory, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, during 2012–2013. 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 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.

**Bojan Nemec** received the Diploma degree in electrical engineering and the M.Sc. and Ph.D. degrees in robotics from the University of Ljubljana, Slovenia, in 1979, 1982, and 1988.

He is currently a Senior Research Associate with the Department of Automatics, Biocybernetics, and Robotics, Jožef Stefan Institute, University of Ljubljana. He spent his sabbatical leave with the Institute for Real-Time Computer Systems and Robotics, University of Karlsruhe, Germany, in 1993. His research interests include robot control, robot learning, service robotics, and sports biomechanics.

Dr. Nemec received the NTF award for best paper in the category of Entertainment Robots and Systems at IROS 2009. He has been a program committee member of several conferences and general chair of the Workshop in the Alpe-Adria-Danube region in 2013.

**Auke Jan Ijspeert** (M'00) received the B.Sc. and M.Sc. degrees in physics from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, and the Ph.D. degree in artificial intelligence from the University of Edinburgh, Edinburgh, U.K, in 1999.

He is currently a Professor with the EPFL, where he is also the Head of the Biorobotics Laboratory. His research interests include the intersection between robotics, computational neuroscience, nonlinear dynamical systems, and applied machine learning. He is interested in using numerical simulations and robots to get a better understanding of sensorimotor coordination in animals and in using inspiration from biology to design novel types of robots and adaptive controllers.

Dr. Ijspeert received the Best Paper Award at the IEEE International Conference on Robotics and Automation in 2002, the Industrial Robot Highly Commended Award at the Eighth International Conference on Climbing and Walking Robots in 2005, and the Best Paper Award at the IEEE-RAS International Conference on Humanoid Robots in 2007. He has been a program committee member of more than 40 conferences.

**Aleš Ude** received the Diploma degree in applied mathematics from the University of Ljubljana, Slovenia, in 1990 and a Dr.-Eng. sciences degree from the Faculty of Informatics, University of Karlsruhe, Karlsruhe, Germany, in 1995.

He is currently the Head of the Humanoid and Cognitive Robotics Lab, 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, perception of human activity, humanoid cognition, and humanoid robotics in general.

Dr. Ude received the Science and Technology Agency fellowship for postdoctoral studies from the ERATO Kawato Dynamic Brain Project, Kyoto, and several fellowships from the Japan Trust International Research Cooperation Programme. He was a General Chair of the 2011 Humanoids conference. In recent years, he has been a member of program committees of all major robotics conferences, including the IEEE International Conference on Robotics and Automation, Intelligent Robots and Systems, and Really Simple Syndication.