

Modulation of Motor Primitives using Force Feedback: Interaction with the Environment and Bimanual Tasks

Andrej Gams^{1,3}, Bojan Nemeč¹, Leon Žlajpah¹, Mirko Wächter², Auke Ijspeert³, Tamim Asfour², Aleš Ude¹

Abstract—The framework of dynamic movement primitives allows the generation of discrete and periodic trajectories, which can be modulated in various aspects. We propose and evaluate a novel modulation approach that includes force feedback and thus allows physical interaction with objects and the environment. The proposed approach also enables the coupling of independently executed robotic trajectories, simplifying 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. The coupling term modifies the trajectory in accordance to either the desired position or external force. The strengths of the approach are shown in bimanual or two-agent obstacle avoidance tasks, where no higher level cognitive reasoning or planning are required. Results of simulated and real-world experiments on the ARMAR-III humanoid robot in interaction and object lifting tasks, and on two KUKA LWR robots in a bimanual setting are presented.

I. INTRODUCTION

A sizeable part of robotics research directly or indirectly deals with all aspects related to complex human environments [1]. Unstructured and changing human environments, such as the kitchen, demand the ability to learn new tasks using sensor feedback in a natural way and without the need for an expert [2]. Besides trajectory generation, 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.

Different ways of encoding trajectories allow for different possibilities of modulation, interpolation, and categorization [3]. There exist different approaches, for example splines and wavelets [4], Gaussian Mixture Regression [5], or Hidden Markov Models [6]. In this paper we build on dynamic movement primitives, first introduced by Ijspeert et al. [7]. DMPs provide means to encode a trajectory as a set of differential equations that can compactly represent control policies, while their attractor landscapes can be adapted by changing only a few parameters. The latter can be exploited in several ways, for example for reinforcement learning [8],

[9], [10], [11], statistical generalization [12], [13], or they can even be combined in a dynamic way [14], [15].

The dynamic systems design of DMPs enables us to incorporate sensory feedback, either in the transformation system, e. g. for on-line obstacle avoidance [16], the canonical system, e. g. for different periodic tasks [17], or both. An example of such is using the so-called slow-down feedback to stop the execution of the trajectory [18], [19].

In this paper we propose a new approach to modifying trajectories. We first record the trajectory data as the robot moves along a defined trajectory, and then use this data to improve its 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 [3]. The coupling term can either be the real force of interaction, or a virtual force defined from the positions of two manipulators/agents. The final waveform and amplitude of the coupling term is learned in a few iterations in an iterative learning control (ILC) [20] manner. Appeal to iterative learning comes from 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 [21].

Use of force feedback and robotic force control are well studied problems [22]. The use of force feedback to change the output velocity of a manipulator was reported by Hogan [23]. However, the DMP framework was, with notable exceptions, thus far mostly constrained to the kinematic domain. Using a force feedback term to learn and improve task execution in the DMP framework was demonstrated for a periodic task of wiping a flat or curved surface [2]. Contrary to the approach in this paper, complete trajectory waveforms were modified within a few periods of the task using regression methods. Similar was reported by Ernesti et al. [24]. Notably, Pastor et al. [1] have implemented a low-level position and force control system that integrates with DMPs at an acceleration level, allowing for reactive and compliant behaviors. The key idea in their approach is that after an execution with a disturbance, they apply a controller on the measured forces and use the output of the controller in a feed-forward manner to realize grasping of an electric torch [1] or a battery drill [25].

In this paper we present: 1) a force-based modulation of the DMP at both velocity and acceleration levels; 2) coupling of DMPs for bimanual tasks (both Section II);

Research leading to these results was supported in part by the EU Seventh Framework Programme under grant agreement no. 270273, Xperience, and by Sciex-NMS^{CH} project 12.018.

¹ Humanoid and Cognitive Robotics Lab, Dept. of Automatics, Biocybernetics and Robotics, Jožef Stean Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia, andrej.gams at ijs.si

² High Performance Humanoid Technologies, Institute for Anthropomatics, Karlsruhe Institute of Technology

³ Biorobotics Laboratory, École Polytechnique Fédérale de Lausanne

3) learning the open coupling terms with iterative learning control (Section III); 4) improved stability when using both levels as compared to only acceleration level modulations (Appendix). Section IV describes experiments conducted on the ARMAR-III robot and two KUKA LWR robots in a bimanual setting. Concluding remarks are in Section V.

II. MODULATING DYNAMIC MOVEMENT PRIMITIVES

A. Dynamic Movement Primitives

DMPs have been thoroughly discussed in the literature [7], [18], [19]. In the following we provide only the basic information, based on the formulation in [18], [12]. 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

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

$$\tau \dot{y} = z. \quad (2)$$

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

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

$$\Psi_i(x) = \exp\left(-h_i(x - c_i)^2\right), \quad (4)$$

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

$$\tau \dot{x} = -\alpha_x x, \quad (5)$$

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

The weight vector \mathbf{w} , composed of weights w_i , defines the shape of the encoded trajectory. [7] and [12] describe the learning of the weight vector. Multiple DOFs are realized by maintaining separate sets of eqs. (1) – (4), while a single canonical system given by (5) is used to synchronize them.

B. Modulation

Online modulations are among the most important properties offered by the dynamical systems approach [19]. An example of spatial modulation is including an obstacle avoidance term in Eq. (1) [16], [19]

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

where C_m is the modulation term. In this paper we term this kind of modulation as a modulation at the *acceleration* level. On the other hand, a simple repulsive force $h(y)$ to avoid moving beyond a given position in the task space [3] can be specified by modifying Eq. (2) into

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

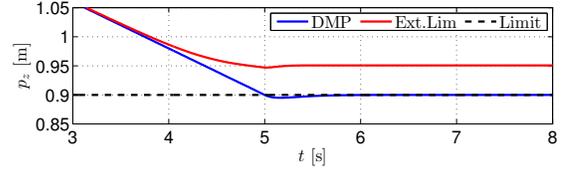


Fig. 1. Response in presence of an external limit according to Eq. (7) and (8), with the limit set at $y_L = 0.9\text{m}$. The red trajectory does not reach the target position because the repulsive force acts before the actual limit.

while leaving Eq. (1) in the original form. In this paper we term this as a modulation at the *velocity* level. A simple repulsive force to avoid hitting y_L can be defined as [3]

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

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

Defining the repulsive force as in Eq. (8) prevents the robot from getting into contact with objects and the environment and therefore, it cannot be used for manipulation tasks. We therefore propose a modification of the approach by defining a different repulsive force, yet keeping Eq. (7) in its current form. Instead of using Eq. (8), we propose using the measured force F , which arises from the interaction with the environment

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

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

$$F(t) = kd(t), \quad (10)$$

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

Using only Eq. (7), i.e. only velocity level modulation, results in an overshoot of forces upon environment contact, see Fig. 2. 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

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

$$\tau \dot{y} = z + C, \quad (12)$$

$$C = cF(t), \quad (13)$$

with c and c_2 scaling constants. Fig. 2 shows the effect of damping, with $c_2 = 30$, determined empirically. The bottom plot shows that the resulting force overshoots at $t = 5$ s. This overshoot results in oscillations in the direction of the force. We show in the Appendix that adding the coupling term to both velocity and acceleration level is better than only to the acceleration level, because adding the coupling term only to the acceleration level results in significantly larger

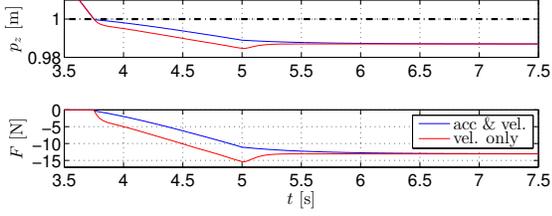


Fig. 2. Simulation results that show the difference of adding a coupling term at the velocity level (red) or also including the derivative of the force at the acceleration level (blue). The trajectory was encoded to start at 1.3m and end at 0.9m in 5 s. An obstacle is encountered at 1m, plotted in the top plot by the black dash-dot line. The top plot shows the positions and the bottom plot the resulting forces. The obstacle was intentionally not stiff ($k = 1000\text{N/m}$) to emphasize the difference.

oscillations in the direction of the force. Such performance would impose great restrictions on the use of the proposed iterative learning algorithm.

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. To minimize the force of contact we propose to employ an ILC learning algorithm, which takes a few repetitions of the exact same task to learn the waveform and amplitude of what we call the *coupling term*. Using the coupling term, we can minimize the error or force of contact and thus also mitigate the need for tuning the scaling factors. The learning algorithm is explained in detail in Section III.

C. Cooperative DMPs

Well studied control approaches for two robot arms with a central controller exist, i.e. [26]. However, given a control architecture where each robot is controlled on its own, possibly with conflicting trajectories, an approach for motion synchronization is needed. An example of this is cooperation of two stand-alone robots/agents working together when carrying a large object. We used two independently controlled robot arms for bimanual tasks.

Let us assume that two trajectories, both given by DMPs, are executed by two different robot arms. By introducing a virtual spring between the end effectors of both arms, we introduce a force that alters both trajectories. For one DOF, this force is defined as

$$F_{1,2} = k(d_d - d_a), \quad (14)$$

where d_d is the desired distance between the robots (tips) and d_a is the actual, measured difference. k is the virtual spring constant. Measured force can be used instead of a virtual spring. The force that acts on DMP_1 is opposite to the force acting on DMP_2

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

We introduce these forces, again scaled by c , into each DMP.

Eq. (16) – (21) define what we label *cooperative DMPs*:

$$\tau \dot{z}_1 = \alpha_z(\beta_z(g_1 - y_1) - z_1) + f_1(x) + c_2 \dot{C}_{1,2}, \quad (16)$$

$$\tau y_1 = z_1 + C_{1,2}, \quad (17)$$

$$C_{1,2} = c F_{1,2} \cdot l_{f1}, \quad (18)$$

$$\tau \dot{z}_2 = \alpha_z(\beta_z(g_2 - y_2) - z_2) + f_2(x) + c_2 \dot{C}_{2,1}, \quad (19)$$

$$\tau y_2 = z_2 + C_{2,1}, \quad (20)$$

$$C_{2,1} = c F_{2,1} \cdot l_{f2}. \quad (21)$$

The variable l_f defines the relation leader-follower. If $l_{f1} = l_{f2} = 1$, then both robots will adapt their trajectories, reaching an equilibrium between the two. They will, in fact, both follow the average trajectory, within tolerance (after learning discussed in the next section), but at the defined distance d_d . 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. Vice-versa applies as well. Leader-follower relation can be determined by a higher level planner, which is beyond the scope of this paper. In general, it depends on the needs and circumstances of a specific task.

III. ITERATIVE LEARNING CONTROL

The coupling terms $C_{1,2}$ and $C_{2,1}$ need to be learned in such a way that the robots maintain the desired force and/or displacement, i.e. $F_d = F$, where F is defined as in (10) or (14). In the following we propose an ILC-based algorithm to learn $C_{1,2}$ and $C_{2,1}$. The proposed algorithm avoids the necessity to accurately model the dynamics of the robot and the environment. See a thorough review by Bristow et al. [20] for more details on ILC and for the stability analysis of an ILC algorithm.

Upon the execution of the given task for the first time, the sensors register the resulting force. If the task were to be executed again without any difference, the sensory readings would not change but 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 (C_i stands for either $C_{1,2}$, i or $C_{2,1}$ i) is then defined as suggested by the ILC theory [20]

$$C_i = c e_{F,i} + F_{c,i}, \quad (22)$$

$$F_{c,i} = Q(F_{c,i-1} + L c \dot{e}_{F,i-1}) \quad (23)$$

$$e_{F,i} = F_{d,i} - F_i \quad (24)$$

where index i denotes the i -th epoch, c is the force gain, e_F is the force error calculated from the difference of the desired F_d and measured F forces, F_c is the learned coupling force, and Q and L are positive scalars. The coupling term given by Eq (22) is known as current iteration ILC, 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 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 in Eq. (13) instead of C for interaction

with the environment. Similarly we use C_i instead of $cF_{1,2}$ in Eq. (18) and $-C_i$ instead of $cF_{2,1}$ in Eq. (21).

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, the same as $f(x)$ in Eq. (1), with x a phase variable, see [12] for details. Several advantages speak in favor of encoding the coupling term in this manner. For example, the nonlinear encoding acts as a filter [3] and thus cancels out the sensor noise. The main advantage is that the amplitude depends on the same canonical system as the trajectories.

IV. EVALUATION

A. Contact with the Environment

We performed several simulated numerical and real world experiments. The real world experiments were performed on the ARMAR-III robot [27] at 30 Hz.

We applied the proposed algorithm to minimize the force upon impact ($F_{d,i} = 0$) with a cardboard box as shown in Fig. 4. The movement was repeated 10 times. Fig. 3 shows the results. The resulting forces are reduced with each epoch and they tend to appear later during the movement, as shown in the bottom plot. Low repeatability of the object (deformable cardboard box) can account for inconsistencies in the appearance of the force. The trajectory was defined as a minimum jerk trajectory going from 1.35m to 1m in 15 s, modified upon impact with the small box at roughly 1.08m. 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 would squish the box. A force sensor in the wrist of the robot was used to measure the contact force. Even though the resulting force does not reach exactly 0N, the results show a great reduction in the force. The error can be attributed to deformability of the box, repeatability of the execution, low control loop frequency (30 Hz), and very noisy force measurements.

The algorithm can be applied to produce a desired force of contact, see Eq. (24). Fig. 5 shows the results of applying the algorithm to produce a desired force of contact at $F_d = 4N$.

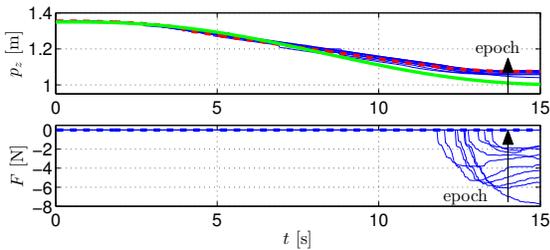


Fig. 3. Minimization of the contact force after 10 repetitions of a real robot movement that collides with a slightly deformable cardboard box. The green line depicts the original desired trajectory and the blue lines are the performed trajectories. The final trajectory is marked with a red dotted line. The bottom plot shows the measured forces occurring during the contact. The dotted line shows the desired force.

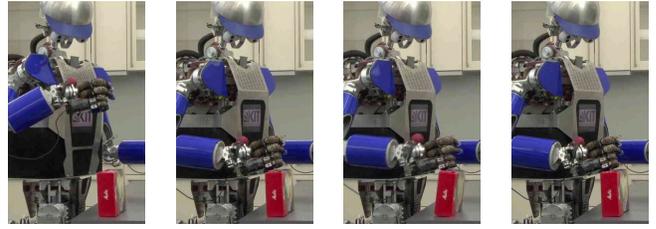


Fig. 4. Image sequence showing the collision of the ARMAR-III robot's hand robot with a slightly deformable cardboard box on the table. The third image shows the deformation of the box after the first epoch. The last image shows that contact after the 10th, final epoch. The image shows that there is considerably less deformation of the cardboard box.

The desired force was defined to appear only after the impact.

B. Bimanual Tasks

We applied the proposed method to couple trajectories of the arms of the ARMAR-III robot to lift a cardboard box. The original trajectories of the arms defined a forward-upward motion, which ended at different heights and distances from the body for each arm. The left arm was to end 0.2m higher and 0.14m more forward than the right arm. Note that we treated the arms of the robot as two separate agents, without a central controller. In order to use the given trajectories for the task of lifting a box, for example to place it on a shelf, we applied our algorithm to couple the trajectories in p_z (up-down) and p_y (forward-backward) directions. After the learning, both arms reached a common height and distance, effectively maintaining the orientation of the box while lifting it. Fig. 7 shows the distances between the arms during the execution of the trajectories, estimated using direct kinematics. The movements were repeated 5 times. The dotted lines show the original uncoupled trajectories while the red lines show the final distance between the arms in both directions. Note that both arms adapted their trajectories, as can be observed from Fig. 6.

The leftmost image in Fig. 6 shows the final position of the uncoupled trajectory. The other five images show the learning progress (left to right) of the cardboard box position

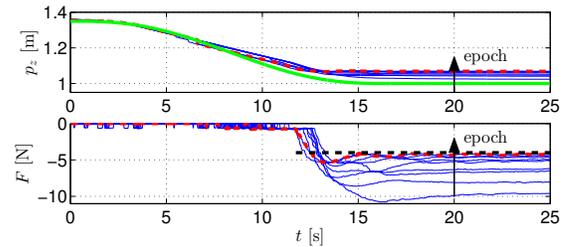


Fig. 5. Applying the learning algorithm to produce a desired force of contact at $F_d = 4N$ (measured force is -4N). The movement was repeated 9 times. The desired (green) and real trajectories are shown in the top plot. The final trajectory is depicted in red. The bottom plot shows the measured forces occurring during the contact (blue), while the red dotted line depicts the final force. The black dotted line shows the desired force, set to appear only after the contact.

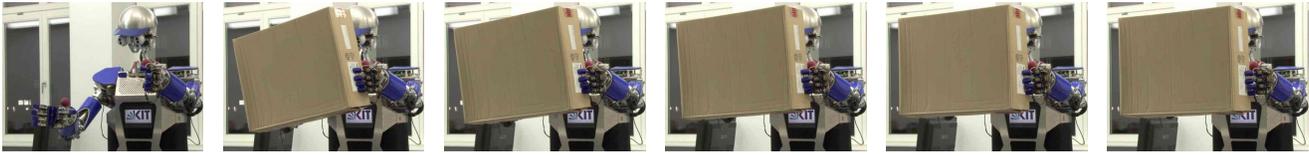


Fig. 6. Images showing the final position of the box in the bimanual manipulation task with the ARMAR-III robot. The first image shows the position of the arms after the execution of the original, uncoupled trajectories. Note that they are at different heights. The following images show the end positions of the box, held with the arms, after each epoch. The final image shows the position after the learning. Note that the arms have reached almost the desired, parallel position and therefore the box is not tilted anymore.

after each learning epoch.

In this case study we showed that the arms can be coupled using a virtual force, calculated from the distance of the arms as given in Eq. (14). Besides showing the applicability of the approach using positions, practical reasons would not allow the use of the force sensor for this case. The contact of the robot and the box was not rigid – the box was changing the orientation in the arms, while the arms themselves were not, which resulted in slight sliding and different force profiles in each epoch. A rigid contact, for example when gripping for the handles, would allow the use of real force feedback. In the following case studies we used real force feedback.

C. Obstacle Avoidance in Bimanual Tasks

The framework of DMPs allows additional modulation of the trajectories according to some external feedback, even if the trajectories are coupled. Let us consider a task of carrying a large object with two independently controlled agents/robots, where one of the robots encounters an obstacle and has to adapt its trajectory in order to avoid collision. The change of trajectory results in a difference in the relative position to the other robot and therefore a force along the manipulated object. Our proposed learning algorithm can be applied to minimize the force along the object in a few iterations when one of the robots has to adapt the trajectory, for example for obstacle avoidance.

In our experiments we used two KUKA LWR robots with 7 DOFs, but again without a common controller. The robots were carrying a rigid object and their grip on the object was strong enough to be considered rigid. The robots are shown in Fig. 9. We applied the same online obstacle avoidance algorithm as presented in [16]. We applied it to the left robot.

The cooperative DMPs were set in a leader-follower relation, the left robot being the leader ($l_{f,L} = 0$). From

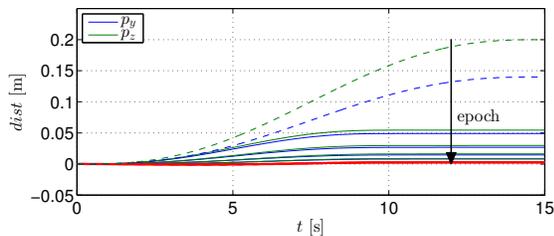


Fig. 7. The distance between the arms in p_y and p_z directions during each epoch of learning for bimanual box manipulation. The dotted lines show the original, uncoupled trajectories, while the red line show the distance after the final, 5th epoch.

the start both robots had identical 20 s vertical trajectories ($y_{LR} = \pm 0.4m$), but the left robot encounters an obstacle at $p_x = 0.7m$, $p_y = 0.45m$, $p_z = 1m$ and therefore has to apply obstacle avoidance. Our proposed algorithm was utilized to minimize the forces between the rigidly connected robots. Fig. 8 shows the results of learning to minimize the forces between the robots after 7 movements. The top plot shows the $p_y - p_z$ trajectory plot. The trajectories are for presentation purposes depicted at $y_{L,R} \pm 0.2m$, but they were executed at $y_{L,R} = \pm 0.4m$. The dotted red lines show the original trajectories. The black line connecting the robots show the connecting stick every 5 s. The bottom plot shows the resulting forces between the robots, in p_y direction (blue), and the resulting torques around the global z (vertical) axis.

Fig. 10 shows the results of a similar real world experiment, where also the right robot encounters an obstacle at $p_x = 0.75m$, $p_y = -0.4m$, $p_z = 0.9m$. The obstacle is set so that the robot must avoid it in the $-p_x$ direction. The resulting movement leads to a rotation of the stick between the robots around the world z axis. This was a direct result of cooperation and no higher level planners were applied. The results indicate the ability of the algorithm to provide trajectories which can guide wide objects through narrow passages, e. g. a long board through a door, without any higher-level planning. In the top 3-D plot we can also notice the initial oscillations. These are the result of obstacle avoidance and cooperative terms acting on the trajectory of the right robot. The oscillations disappear by the final, 7th

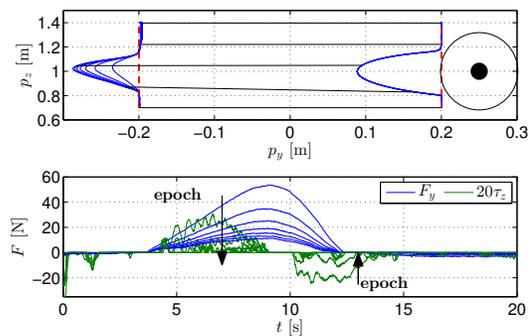


Fig. 8. Real world results of obstacle avoidance with the right robot following the left one. The top plot shows the $p_y - p_z$ trajectories, the original trajectories depicted with red dotted lines. The bottom plot shows the resulting forces and the resulting torques (scaled 20 times for presentation purposes).



Fig. 9. Two obstacles avoidance. The leader robot (left arm) encounters an obstacle on the left (orange ball). Before that, the follower robot encounters an obstacle in front of itself (pink foam). All robot experiments are presented in the video accompanying the paper.

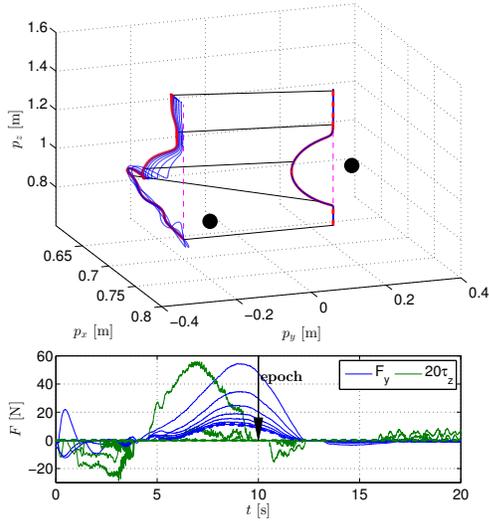


Fig. 10. Real world results of double obstacle avoidance of cooperative DMP trajectories. The left robot is the leader, which encounters an obstacle, but the follower also encounters an obstacle. The top plot shows the trajectories, the final, 7th plot marked in red. Dotted red lines show the original trajectories. The bottom plot shows the resulting force in p_y direction and the resulting torque (scaled 20 times for presentation purposes) around the world z axis.

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.

V. CONCLUSION

The proposed approach enables learning of coupling terms that establish desired contact forces with the environment and the adaptation of trajectories for cooperative task execution. We showed in the Appendix that it is important that the coupling terms are added at the appropriate level, i. e. velocity and acceleration. We demonstrated in a number of experiments that the approach can be applied to actual interaction and bimanual cooperation tasks and that it can work with real-world noisy signals such as force measurements.

The introduction of forces and torques into the well defined and studied framework of DMPs allows the execution of a vast array of tasks, which were previously inconceivable with purely kinematic variables. Remaining in the framework of the DMPs is therefore one of the key features of the proposed approach. Furthermore, the extremely low number of learning epochs makes on-line learning of the coupled/interactive trajectories a viable possibility. Unlike

reinforcement learning methods, which require tens and even hundreds of repetitions [28], we managed to achieve the learning goals in under 10 for all the demonstrated experiments.

The combination of feedback and learned coupling allows the method to adapt to changes between epochs. It should be noted, that if the method is to converge, it needs a target to converge to. The changes during epochs can be partially handled by the feedback and partially by the ILC, which can be made more robust with proper parameter setting [20]. Once the target converges, the algorithm will converge as well.

APPENDIX

Even though single DMPs are stable, the stability of coupled DMPs, given by Eqs. (16) – (21) is not guaranteed. Cooperative DMPs change the system from single-input-single-output (SISO) into a multiple-input-single-output (MISO) system. The two DMPs both have inputs (g_1, g_2), but the system only has one output since their outputs are subtracted. The coupling comes from the arising force, which depends on the positions of the two robots as given in Eq. (14), where the actual distance is $d_a = p_1 - p_2$ and p_1 and p_2 are the positions of the two robots. 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 $d_a = y_1 - y_2$. Also note that the sign of c has to be set according to Eq. (15), otherwise the system becomes unstable.

For the given, stable DMP parameters, the gain c of the coupling term determines the behavior of the MISO system. We can derive the state-space system (25) – (26) from Eqs. (16) – (21) with the applied feedback $C_{1,2} = -C_{2,1} = k(d_d - d_a)$, $l_{f,1} = l_{f,2} = 1$,

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (25)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \quad (26)$$

where the nonlinear parts $f_1(x)$ and $f_2(x)$ have been omitted. The system matrices for the controllable canonical form are given by

$$\mathbf{A} = \begin{bmatrix} -\frac{\alpha_z \tau + 2ck(c_2 + \tau)}{\tau^2} & 1 \\ -\frac{\alpha_z \beta_z \tau + 2ck}{\tau^2} & 0 \end{bmatrix}, \quad (27)$$

$$\mathbf{B} = \begin{bmatrix} \frac{2k(c_2 + \tau)}{\tau^2} & \frac{2ck(c_2 + \tau)}{\tau^2} & 0 \\ \frac{2k\alpha_z}{\tau^2} & \frac{2ck\alpha_z}{\tau^2} & \frac{k\alpha_z \beta_z}{\tau^2} \end{bmatrix}, \quad (28)$$

$$\mathbf{C} = [1 \ 0 \ 0]. \quad (29)$$

The input vector and scalar output are $\mathbf{u} = [F_c, F_d, g_1 - g_2]^T$ and $y = F$ (in Eq. (26)), respectively. The state vector is defined as

$$\mathbf{x} = \begin{bmatrix} F \\ \dot{F} - \frac{\alpha_z \tau + 2ck(c_2 + \tau)}{\tau^2} F \end{bmatrix}. \quad (30)$$

Since the nonlinear parts $f_1(x)$ and $f_2(x)$ in Eqs. (16) and (19) are bounded and tend to zero as the phase tends to 0, it is sufficient to prove the stability of the linear part of

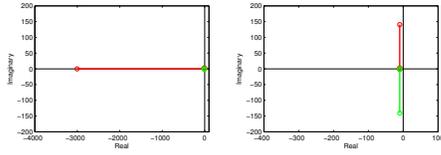


Fig. 11. Root locus plot of the coupled DMP structure with modulation at acceleration level (right) and both the velocity & acceleration levels (left). We varied the gain ck from 0 to 10000 while the gain c_2 was fixed at 1. The system has two poles denoted with red and green. The full circles denote roots at $ck = 0$ and the empty circles at $ck = 10000$.

the system (16) – (21), i. e. of system (25). We assume the environment stiffness as defined by Eq. (10).

The eigenvalues of \mathbf{A} determine the stability and convergence of differential eqn. system (25). These eigenvalues are given as

$$\lambda_{1,2} = \frac{1}{2} \left(-\frac{\alpha_z \tau + 2ck(c_2 + \tau)}{\tau^2} \pm \sqrt{\left(\frac{\alpha_z \tau + 2ck(c_2 + \tau)}{\tau^2} \right)^2 - 4 \frac{\alpha_z \beta_z \tau + 2ck}{\tau^2}} \right). \quad (31)$$

Since all parameters $\alpha_z, \beta_z, c, k, c_2, \tau$ 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 the system (25) converges to a unique attractor point for all positive parameter values. We obtain complex eigenvalues only for unreasonable parameter values, e. g. for large τ .

The difference of the acceleration-velocity based scheme versus only the acceleration based scheme is shown in the Root-locus plot in Fig. 11. It shows that both schemes remain stable with increasing gain ck . On the other hand, the imaginary part of conjugate-complex eigenvalues increases only in the case when just the acceleration level modulation is used, whereas it remains zero when modulating both the velocity and acceleration levels. The results clearly support the proposed velocity & acceleration levels, where the response is always damped, whereas the convergence is slower for modulation at acceleration level only.

REFERENCES

- [1] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, "Online movement adaptation based on previous sensor experiences," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 365–371.
- [2] 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 *2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2010, pp. 560–565.
- [3] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarcic, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Autonomous Robots*, vol. 27, no. 1, pp. 3–23, 2009.
- [4] A. Ude, C. Atkeson, and R. M., "Programming full-body movements for humanoid robots by observation," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 93–108, 2004.
- [5] 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 Robotics and Automation Magazine*, vol. 17, no. 2, pp. 44–54, 2010.

- [6] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, "Embodied symbol emergence based on mimesis theory," *The International Journal of Robotics Research*, vol. 23, no. 4–5, pp. 363–377, 2004.
- [7] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2002, pp. 1398–1403.
- [8] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, p. 682697, 2008.
- [9] F. Stulp, E. A. Theodorou, and S. Schaal, "Reinforcement learning with sequences of motion primitives for robust manipulation," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1360–1370, 2012.
- [10] M. Tamosiunaite, B. Nemeč, A. Ude, and F. Worgotter, "Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives," *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 910–922, 2011.
- [11] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Autonomous Robots*, vol. 33, pp. 361–379, 2012.
- [12] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-specific generalization of discrete and periodic dynamic movement primitives," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 800–815, 2010.
- [13] T. Matsubara, S.-H. Hyon, and J. Morimoto, "Learning parametric dynamic movement primitives from multiple demonstrations," *Neural Networks*, vol. 24, no. 5, pp. 493–500, 2011.
- [14] B. Nemeč and A. Ude, "Action sequencing using dynamic movement primitives," *Robotica*, vol. 30, no. 05, pp. 837–846, 2012.
- [15] T. Kulvicius, K. Ning, M. Tamosiunaite, and F. Worgotter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 145–157, 2012.
- [16] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2009, pp. 2587–2592.
- [17] T. Petrič, A. Gams, A. J. Ijspeert, and L. Žlajpah, "On-line frequency adaptation and movement imitation for rhythmic robotic tasks," *The International Journal of Robotics Research*, vol. 30, no. 14, pp. 1775–1788, 2011.
- [18] S. Schaal, P. Mohajeri, and A. Ijspeert, "Dynamics systems vs. optimal control – a unifying view," *Progress in Brain Research*, no. 165, pp. 425–445, 2007.
- [19] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [20] D. Bristow, M. Tharayil, and A. Alleyne, "A survey of iterative learning control," *IEEE Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.
- [21] G. Heinzinger, D. Fenwick, B. Paden, and F. Miyazaki, "Stability of learning control with disturbances and uncertain initial conditions," *IEEE Transactions on Automatic Control*, vol. 37, no. 1, pp. 110–114, 1992.
- [22] L. Villani and J. De Schutter, *Handbook of Robotics*. Springer, 2008, ch. Force Control.
- [23] N. Hogan, "Impedance control: An approach to manipulation I II III," *Journal of Dynamic Systems, Measurement and Control*, no. 109, pp. 1–24, 1985.
- [24] 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 *2012 IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 57–64.
- [25] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, "Towards associative skill memories," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 309–315.
- [26] 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, 1996.
- [27] T. Asfour, K. Regenstein, P. Azad, J. Schröder, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An Integrated Humanoid Platform for Sensory-Motor Control," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 169–175.
- [28] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *The International Journal of Robotics Research*, 2011.