

# Bio-inspired Learning and Database Expansion of Compliant Movement Primitives

Tadej Petrič<sup>1,2</sup>, Luca Colasanto<sup>1</sup>, Andrej Gams<sup>2</sup>, Aleš Ude<sup>2</sup> and Auke J. Ijspeert<sup>1</sup>

**Abstract**—The paper addresses the problem of learning torque primitives – the torques associated to a kinematic trajectory, and required in order to accurately track this kinematic trajectory. Learning torque primitives, which can be interpreted as internal dynamic models, is crucial to achieve at the same time 1) high tracking accuracy and 2) compliant behaviour. The latter improves the safety concerns of working in unstructured environments or with humans. In the proposed approach, first learning by demonstration is used to obtain the kinematic trajectories, which are encoded in the form of Dynamic Movement Primitives (DMPs). These are combined with the corresponding task-specific Torque Primitives (TPs), and together they form new task-related compliant movements, denoted as Compliant Movement Primitives (CMPs). Unlike the DMPs, the TPs cannot be directly acquired from user demonstrations. Inspired by the human sensorimotor learning ability, we propose a novel method which can autonomously learn task-specific Torque Primitives (TPs) associated to given kinematic trajectories in the form of DMPs. The proposed algorithm is completely autonomous, and can be used to rapidly generate and expand the database of CMPs motions. Since the CMPs are parameterized, statistical generalisation can be used to obtain an initial TP estimate of a new CMP motion. Thereby, the learning rate of new CMPs can be significantly improved. The evaluation of the proposed approach on a humanoid robot CoMan performing reaching task shows fast TP acquisition and accurate generalization estimates in real-world scenarios.

## I. INTRODUCTION

One of the key skills that a humanoid robot should possess is the ability to learn new motor behaviours based on human demonstration [1]. The most common way of learning new behaviours is programming by demonstration (PbD) [2], which can be done by using different sensory systems, e.g., visual [3] or kinematics guidance [4]. The key advantage of PbD in joint space is that the robot kinematics is already adapted to the task and the posture is preserved even when redundant robots are used. Different methods have been proposed for PbD. Dynamic Movement Primitives (DMPs) [5] are generally used for learning kinematic trajectories. To execute the desired DMP trajectory accurately, an underlying robot controller that guarantees accurate tracking is usually employed. For example an impedance controller with high gains in the feedback loop as in [6]. However, using high gains makes robots inherently unsafe for interaction with the

environment or humans, due to the high interaction forces that may occur during unforeseen contacts [7]. Moreover, in the case of humanoid robots this might also lead to an unsuspected fall.

Different approaches can be used to minimise the interaction forces and at the same time assure accurate trajectory tracking. For example, combining high gain impedance control for accuracy together with proximity sensors or an artificial skin for detecting contacts [8]; by using bi-articular mechanical structures based on artificial pneumatic muscles [9] or by accurate inverse dynamic models for control algorithms with active compliance [10]. However, changing mechanical structures or adding artificial skin to the system will increase its overall price. On the other hand, it is impossible to obtain an accurate generic dynamical model, even for a simple task like table wiping, due to the unknown parameters – for example the friction between the sponge and the table.

To overcome the problem of using impedance control by using dynamical models we propose a novel method which can learn missing dynamic parameters of a given task, and encode them as task-specific torque primitives. Hence, there is no need for modelling the task dynamics. Moreover, if a dynamical model is available, our method compensates for uncertainties with the learned torque primitives. By learning the task-specific torque primitives, the control of the robot allows 1) accurate trajectory tracking and 2) compliant behavior, which is the result of using a low gain feedback loop in an underlying impedance controller. While accurate tracking is required for proper task execution, compliant behaviour is essential for ensuring safe interaction with the environment or, most importantly, humans [7].

To enable both accurate trajectory tracking and compliant behaviour, the DMP [11]–[14] framework needs to be extended toward torque-controlled robots. Inspired by the human sensory motor ability [15]–[18], where hand kinematics are learned from errors in extent and direction in an extrinsic coordinate system, and dynamics are learned from proprioceptive errors in an intrinsic coordinate system, we propose an extension by augmenting the DMP framework with Torque-Primitives (TP). The combination of a DMP and a TP forms a *Compliant Movement Primitive* (CMP) and represents a new model-free control approach, while keeping modulation and parametrization properties of the position-based DMPs.

The first step of gaining the CMPs is to learn the desired motion trajectory in DMPs. Different approaches were established in the past, allowing to learn motions (off-line or on-

Research leading to these results was supported by Sciex-NMSCH project 14.069, European FP7 ICT project Xperience (no. 270273) and project WALK-MAN (No. 611832).

<sup>1</sup> Biorobotics Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), Station 14, CH-1015 Lausanne, Switzerland tadej.petric@epfl.ch

<sup>2</sup>Dept. for Automatics, Biocybernetics and Robotics, Jožef Stean Institute (JSI), Jamova cesta 39, 1000 Ljubljana, Slovenia

line) by using for example kinesthetic or haptic guidance [4], [19]. Once the desired motion is obtained, the corresponding TPs need to be acquired. Our contribution is achieving this goal by using a low impedance control loop combined with a recursive regression method that uses the difference between the desired and the actual movement to update the TPs. Hence the acquisition of the TPs is autonomous. In each subsequent time step, and through a few iterations, the TPs are learned. TPs are employed as feedforward terms, essentially representing new learned task-specific dynamics and allowing the robot to accurately and compliantly execute the desired motion. The proposed approach is similar to one observed in humans [17], where the kinematic trajectory is learned in Cartesian space (DMPs) and the task dynamics in the joint space (TPs).

While the proposed approach eliminates the need for dynamical modeling, the CMPs still have to learn TPs for each task variation. However, since the CMPs are structured, they can be added into a database and statistical generalization (as in [20]) can be used to generate new instances to previously unexplored regions within the database. This also significantly improves the rate of learning, because the initial TPs for a new motion is the outcome of the generalization and hence potentially already a good approximation, depending on the size of the database and the actual query. This allows rapid autonomous expansion of the database of CMPs allowing the robot to perform different variations of the same task in a compliant manner without the need of any analytic models of the task or programming experts.

## II. LEARNING OF COMPLIANT MOVEMENT PRIMITIVES

We define Compliant Movement Primitives (CMPs)  $h(t)$  as a combination of kinematic trajectories encoded in Dynamic Movement Primitives (DMPs) and corresponding task-specific dynamics encoded in Torque Primitives (TPs)

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

where  $\mathbf{p}_d$  are the desired task-space trajectories encoded in the DMPs, and  $\boldsymbol{\tau}_f$  are the corresponding task-specific feedforward joint torques encoded in TPs. In the proposed approach the kinematic motion trajectories are first obtained by human demonstration and encoded as DMPs [5], [12]. Next the corresponding torques are obtained using recursive regression based on error learning. The corresponding Torque Primitives (TP) are encoded as a linear combination of radial basis functions. Since DMPs are encoded in task-space, the error mapping of the recursive regression for the joint-space of TPs is done using the Jacobian transpose. A pair of a DMP and a TP now describes a Compliant Movement Primitive (CMP).

### A. Cartesian Motion Trajectories

The following equations for encoding Cartesian motion, i.e. the motion of the end-effector, are valid for one DOF, for multiple DOF the equations are used in parallel. For a point-to-point movement the trajectory for each DOF is described

by the following system of nonlinear differential equations that specifies the attractor landscape of a trajectory  $y$  towards the anchor point  $g$

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f(x), \quad (2)$$

$$\tau \dot{y} = z, \quad (3)$$

$$\tau \dot{x} = \frac{-\alpha_x x}{1 + h}, \quad (4)$$

where  $x$  is the phase variable and  $h$  is used for modifying the execution speed, i.e., usually to slow it down. Therefore it is usually referred to as the slow-down feedback parameter. Note that when controlling more DOF, the phase variable  $x$  is common for all DOFs, while (2) and (3) are separate for each DOF.  $\tau$  is the temporal scaling parameter,  $\alpha_z$ ,  $\beta_z$  and  $\alpha_x$  are defined such that the system converges to the unique equilibrium point. Usually  $\alpha_z = 4\beta_z$ , which makes system critically damped. The nonlinear term  $f(x)$  is given by

$$f(x) = \frac{\sum_{i=1}^N \psi_i(x) w_i}{\sum_{i=1}^N \psi_i(x)}, \quad (5)$$

where parameters  $w_i$  define the dynamics of the second-order differential equations system. They are estimated with regression such that the DMP encodes the desired trajectory. The basis functions  $\psi_i(x)$  are given by

$$\psi_i(x) = \exp(-h_i(x - c_i)^2). \quad (6)$$

$N$  radial basis functions with a width  $h_i > 0$  and centres  $c_i$  are distributed along the trajectory.

### B. Joint Torque Trajectories

The Torque Primitives are learned recursively by executing the encoded DMP motion while using low gain impedance control. The desired motion  $\ddot{\mathbf{p}}_d$ ,  $\dot{\mathbf{p}}_d$ ,  $\mathbf{p}_d$  encoded in DMPs is executed using the following control law

$$\boldsymbol{\tau}_u = \mathbf{J}^T(\mathbf{K}_p \mathbf{e} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_i \ddot{\mathbf{e}}) + \mathbf{N} \mathbf{K}_n \dot{\mathbf{q}} + \boldsymbol{\tau}_f(s), \quad (7)$$

where,  $\mathbf{J}^T$  is the Jacobian transpose,  $\mathbf{N}$  is the null-space matrix,  $\mathbf{e}$ ,  $\dot{\mathbf{e}}$  and  $\ddot{\mathbf{e}}$  are the differences between desired and actual position  $\mathbf{p}$ , velocity  $\dot{\mathbf{p}}$  and acceleration  $\ddot{\mathbf{p}}$ , respectively and  $\mathbf{K}_p$ ,  $\mathbf{K}_d$ ,  $\mathbf{K}_i$  and  $\mathbf{K}_n$  are the constant gain matrices selected such that the robot behaves compliantly, i.e. set to match the low impedance control requirements. For details on Cartesian DMPs including rotations see [21]. The  $\boldsymbol{\tau}_f(s)$  is vector of feedforward torque trajectories  $\boldsymbol{\tau}_f(s) = [\tau_{f,1}(s), \tau_{f,2}(s), \dots, \tau_{f,j}(s), \dots, \tau_{f,M}(s)]^T$ , where  $M$  is the number of DOF. For one DOF,  $\tau_{f,j}(s)$  it is given by

$$\tau_{f,j}(s) = \frac{\sum_{i=1}^N \psi_i(s) w_{i,j}}{\sum_{i=1}^N \psi_i(s)}, \quad (8)$$

where the phase  $s$  goes from 1 towards 0, similarly to the DMP phase from (4), resulting in

$$\tau \dot{s} = -\alpha_s s. \quad (9)$$

Note that unlike the DMPs, the TPs cannot be time invariant, and cannot be stopped, i.e., the torque does not scale linearly. Therefore an execution time must always be provided in

advance, i.e., set by parameter  $\tau$ . However, for each movement variant, e.g., the same movement but at a different speed, a new TP has to be obtained. A library of TPs for a separate motion, or a library of complete CMPs can be build. Generalization and/or graph search can be used to execute new, previously not explicitly learned movements, see details in [22].

However, in some cases, when the desired movement are not covered by the database, the method which can autonomously learn new CMPs needs to be used. To learn new CMPs we propose a new method that recursively updates the weights  $w_{i,j}$  of TPs. The recursive regression method is given by

$$w_{i,j}(t+1) = w_{i,j}(t) + \psi_i P_{i,j}(t+1) \epsilon_j(t), \quad (10)$$

$$P_{i,j}(t+1) = \frac{1}{\lambda} \left( P_{i,j}(t) - \frac{P_{i,j}^2(t)}{\psi_i P_{i,j}(t)} \right), \quad (11)$$

$$(12)$$

where  $P_{i,j}$  is the covariance. The initial parameters are set to  $P_i = 1$ ,  $w_i = 0$ ,  $\lambda = 0.995$  and the update rate is defined similar as in [23], with

$$\epsilon(t) = \mathbf{J}^T (\alpha_t (\mathbf{p}_d(t) - \mathbf{p}(t)) + \beta_t (\dot{\mathbf{p}}_d(t) - \dot{\mathbf{p}}(t))). \quad (13)$$

Here the rate of learning is defined by setting the parameters  $\alpha_t$  and  $\beta_t$ . The error vector is defined as  $\epsilon = [\epsilon_1, \epsilon_2, \dots, \epsilon_j, \dots, \epsilon_M]$ , where  $M$  is the number of DOFs. Note that each DOF is updated separately using (10)-(11). The learning, i.e., the motion execution with learning, is repeated as long as the desired error metric is not below the desired threshold. Once the required accuracy of motion is met, either only the TP or the complete CMP can be added into the database.

### III. AUTONOMOUS DATABASE EXPANSION

Autonomous learning of CMPs simplifies the execution of dynamically versatile tasks while ensuring accurate and compliant execution of the motion. However, since torques are not linearly scalable, TPs have to be learned for every variation of the task. These include, for example, different speeds, payloads, goals, etc. This new learning can be, however, avoided or at least significantly accelerated by using statistical generalization techniques, which can generate first approximations of the TPs based on a given query point. In case the generalized TPs satisfy the given error criteria, they can be immediately added to the database of motion. If not, the recursive regression method (Section II) can be applied using the generalized TP for the initial approximation, vastly reducing the number of needed iterations of motion.

Assuming that the robot should accurately track the desired trajectory encoded in DMPs, the sum of task space error throughout the iterations is used to determine if the new TPs should be added to the database, i.e.  $\mathbf{H}_x^{\text{TP}}$ . The error metric is defined by

$$e_p = \sum_{j=1}^L \|e(j)\|, \quad (14)$$

where  $e(j)$  is the vector of absolute difference between the actual task space position  $\mathbf{p}$ , and the desired position (DMP)  $\mathbf{p}_d$ .  $L$  is the number of steps inside one movement execution (iteration).

By encoding the torque signals as TPs, we obtain a set of  $M$  examples

$$\mathbf{H}_x^{\text{TP}} = \{\mathbf{w}_{\tau k}, \mathbf{c}_k\}, \quad k = 1, 2, \dots, M, \quad (15)$$

where a CMP, defined by weights  $\mathbf{w}_q$  in DMP and weights  $\mathbf{w}_\tau$  in TP is used to execute a task, defined by the query  $\mathbf{c}$ , with a low feedback gain and thus in a compliant manner. By using Gaussian process regression (GPR) for statistical regression

$$\mathbf{F}_{\mathbf{H}_x^{\text{TP}}} : \mathbf{c} \mapsto [\mathbf{w}_q, \mathbf{w}_\tau]. \quad (16)$$

we can compute the appropriate TP parameters for the given query  $\mathbf{c}$  i.e., for the task variation. For the details on generalization for DMPs see [20] and for CMPs see [22];

The process of learning TPs repeats as long as the  $e_p > e_c$ , where  $e_c$  is a predefined constant. Once the following criteria is met, the TPs are added into the database of motion  $\mathbf{H}_x^{\text{TP}}$ . With the proposed approach, the database of CMPs can be autonomously expanded.

## IV. EXPERIMENTAL EVALUATION

The proposed method was evaluated in simulations and on a real humanoid robot CoMaN developed by at the Italian Institute of Technology [24].

The method was evaluated in three different scenarios. First, demonstrating the ability to learning new torque primitives while moving an arm to reach towards a certain point in space in a well structured and unconstrained space. Second, learning new torque primitives while using the previous experience by applying statistical generalisation. Third, learning of torque primitives while transferring a skill from human tutor in an unstructured environment to perform a hammering task.

The experimental setup and the initial robot pose used for the first task can be seen in Fig. 1. We denote the initial robot Cartesian position as  $\mathbf{p}_0 = [0, 0, 0]$ . The robot was commanded to reach a desired point in space. This task was chosen because it is similar as the one used in studying human learning of sensory motor ability reported in [17]. To compare the behaviour of the proposed controller to human sensory motor ability, we have choose 8 different end points on a  $\mathbf{p}_x - \mathbf{p}_y$  plane equally spaced on a circle with radius of 14 cm. The Cartesian reaching trajectories were then encoded in the DMPs as parts of CMPs. The desired trajectories are shown as red-dotted lines in Fig. 2. To learn the proper torque primitives an algorithm, i.e. Section II, was employed. Note that the desired trajectories are encoded in the Cartesian space, i.e. task-space, and the corresponding torque primitives are encoded in the joint space, see Eq. (7). The mapping of the task-space error  $e$  to the joint-space was done by using the task-space Jacobian transpose. The learning of the TPs was done recursively using Eq. (10)-(13)

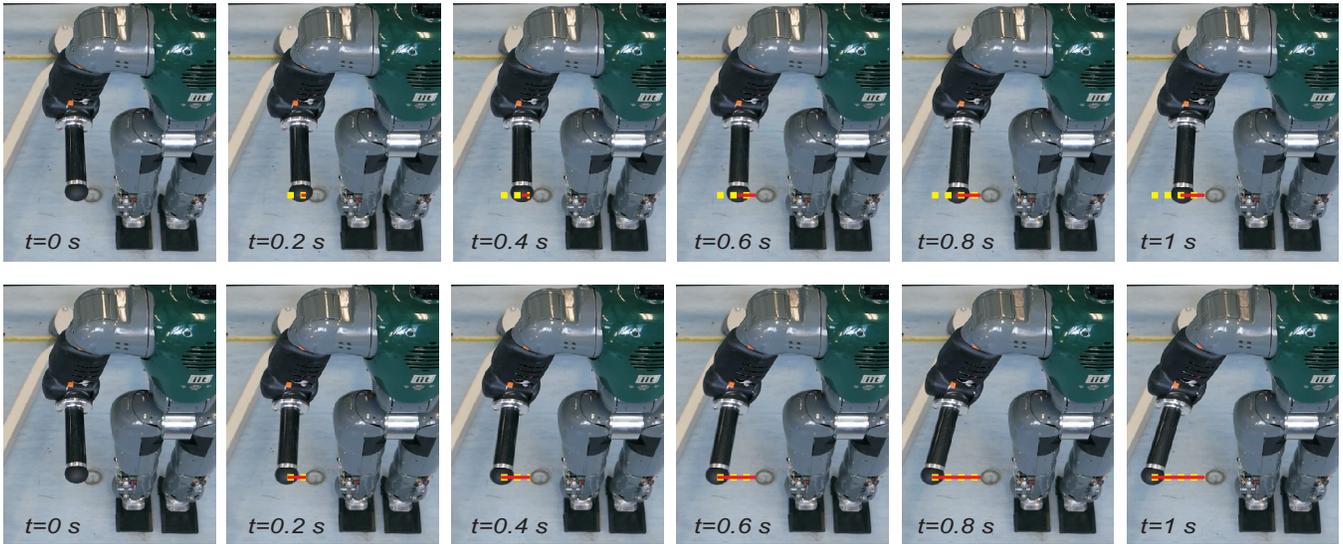


Fig. 1. Sequence of motion for one representative example, i.e. moving to the left. The top row shows the sequence of motion for the first iteration where the TPs are zero, and the bottom plot shows the sequence of motion after learning of TPs.

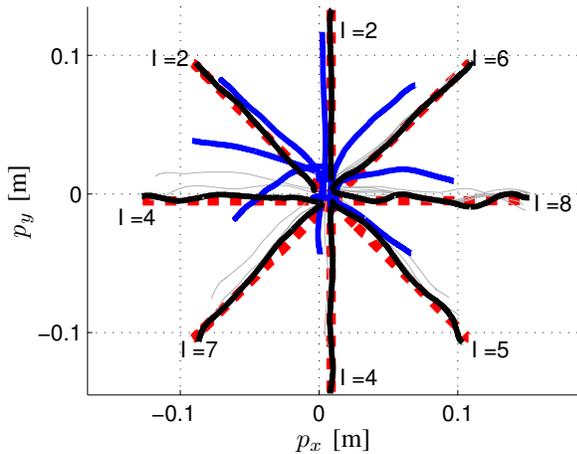


Fig. 2. Results from the real CoMan robot shows eight different reaching examples. The desired trajectory is shown with dotted red line, initial trial is shown with blue line, grey thin lines show the intermediate learning iterations and the black thick line shows the learned behaviour. The  $I$  indicates the number of iterations needed for successful learning of TPs, i.e. once the error  $e$  was smaller than the desired threshold, see Eq. (14).

in each time step. Note that each executed motion started from the same initial position.

The learning results for all 8 examples are shown in Fig. 2, where we can see the difference between the initial movement execution (blue lines) and the last iteration of learning (black lines). We can see that in all eight examples the proposed approach was able to significantly improve the tracking accuracy by updating the TPs. Note that in this example no inverse dynamics or friction compensation of the robot arm was used.

The successful learning is also shown in the Fig. 1, where we compare the behaviour on one representative example, i.e. moving to the left. The top row of Fig. 1 shows the sequence of motion for the initial movement execution where

the initial torque primitives were zero. It can clearly be seen that without TPs the impedance control with low gains is not able to track the desired motion. However, if the TPs are learned using the proposed human inspired controller, we can see in the bottom sequence of Fig. 1, that the tracking is significantly improved. In fact, we can see perfect matching between the desired (yellow dotted line) and actual position (red line).

Although learning of the TPs simplify the execution of dynamically versatile task, TPs still need to be learned for every variation of task execution. If no prior knowledge is used during the learning, this might be a time consuming task. However, by storing the known examples into the database, and applying statistical generalization for estimating the initial TPs for the first trial, we can significantly improve the rate of learning. In fact if the initial approximation already satisfied given criteria it can immediately be added into the database of CMPs motions. Otherwise, the proposed learning of TPs can be applied to update them accordingly.

The evolution of torques for one representative example, i.e. moving sideways towards  $g = [0.1, -0.1]$ , for all four joints of the right hand, of TPs for both learning without or with a prior knowledge with generalization, is shown in Fig. 3. The left hand side plots shows the evolution of torques of learning without generalisation, and the right hand side plots shows the evolution with the use of prior knowledge and statistical generalization.

The red dotted line shows the initial state of TPs, the blue lines shows the TPs during learning iterations and the black line shows the TPs once the learning criteria was met. We can see on the left side that 5 iterations were needed for learning reaching the desired criteria, i.e. desired accuracy of motion. On the other hand on the right side we can see that with the use of statistical generalization the initial TPs were already close to the final solution. Therefore, fewer

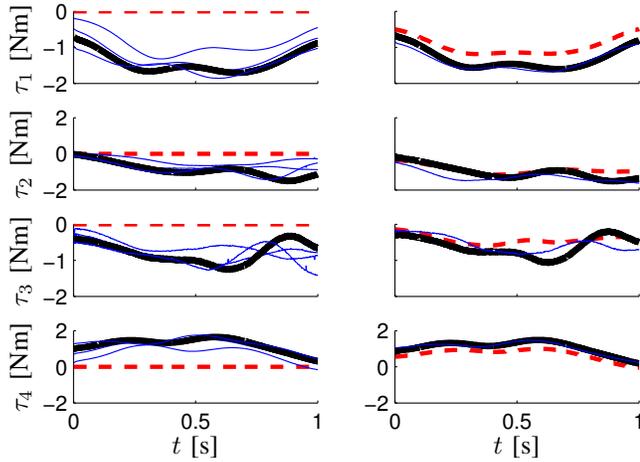


Fig. 3. The comparison between learning of TPs without (left hand side) and with (right hand side) prior knowledge, using statistical generalization for initial TPs estimation. The evolution of torques is shown for one representative example, i.e. moving to the side  $g = [0.1 - 0.1]$ . The red dotted line shows the initial TPs, blue line shows TPs during learning iterations, and black line shows learned TPs.

learning iterations were needed for satisfying the criteria. By using statistical generalization to define the initial torques and further applying recursive regression for updating TPs to minimise the error shows also that the proposed algorithm has the ability to re-learn the torques if needed.

By combining the statistical generalization based on prior knowledge in the database and the recursive regression algorithm for updating the TPs, we can rapidly extend the database with exploration. An example showing rapid database expansion with learning of TPs is shown in Fig. 4, where the database of CMPs was gradually expanded. The learning sequence is indicated with Roman numbers. Note that for the first trajectory, i.e. I, the database was empty, therefore the behaviour is similar as in the Fig. 2. For the second case, i.e., II, only the knowledge from case I was in the database, therefore the initial TPs were completely wrong. Nonetheless, the proposed approach was able to update the TPs to meet required criteria, which show that the proposed approach has the ability to re-learn the TPs if needed. This also shows that the proposed approach can cope with sudden changes in the task dynamics and re-adapt if needed.

The ability to re-learn is also one of the key features that a human possess. It was shown in [17] on a similar reaching experiment as presented here but with humans, that they also have to adapt to changes in the task dynamics, i.e. if the task dynamic is different than before even humans have to re-learn. From Fig. 4, we can also see that once more knowledge is present in the database, the initial TPs estimations are better, from which follows that fewer iterations for updating TPs are needed to meet the required criteria.

In general the learning rate of the proposed algorithm not only depends on the initial TPs state but also on gains  $\alpha_t$  and  $\beta_t$ . If these gains are too high, the system might potentially be destabilised, or vice versa, if they are too low,

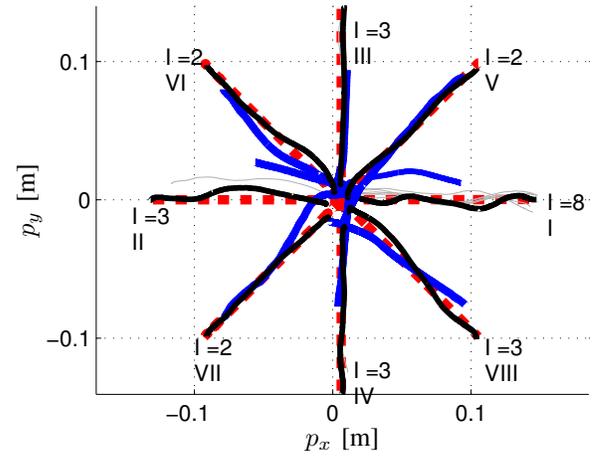


Fig. 4. Results from the real CoMan robot shows eight different reaching examples using prior knowledge for initial TPs approximation. The sequence of learning is determined with the Roman number. The desired trajectory is shown with dotted red line, initial trial is shown with blue line, grey thin lines shows the intermediate learning iterations and the black thick line shows the learned behaviour. The I indicates the number of iterations needed for successful learning of TPs, i.e. once the error  $e$  was smaller than desired threshold, see Eq. (14)

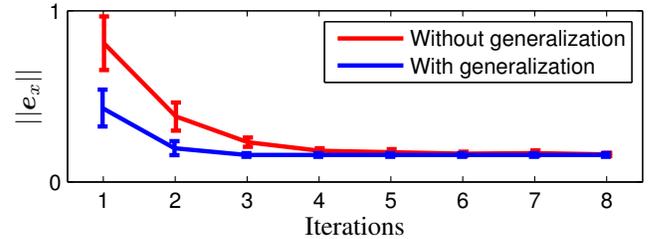


Fig. 5. The average error and the standard deviation for all eight examples for both cases, i.e. with and without prior knowledge and statistical generalization.

the learning rate would not be efficient. In our experiments we set them empirically to  $\alpha_t = 1$  and  $\beta_t = 0.5$ , which resulted in slightly faster learning rate than reported for the human experiment [17]. In Fig. 5 we can see the learning rate for both cases, i.e. with and without using the prior knowledge for initial TPs estimation. We can see on the plot that learning is almost twice as fast if the initial TPs are defined using statistical generalization.

Since CMP is combined with low gain impedance control, we can also assume that interaction forces in case of collision with an unforeseen object will be significantly smaller compared to the impedance control with similar tracking performance. Note that to achieve similar tracking performance with only impedance control the robot would be stiff. To investigate the performance in case of a collision, we chose a task of hammering. Here the robot needs to learn the tool dynamic, i.e. the dynamics of the hammer, and then interact with the environment. In case of hammering the impact with the environment is instantaneous.

The results of learning the skill of hammering are shown in Fig. 6. The top plot shows the adaptation of kinematic

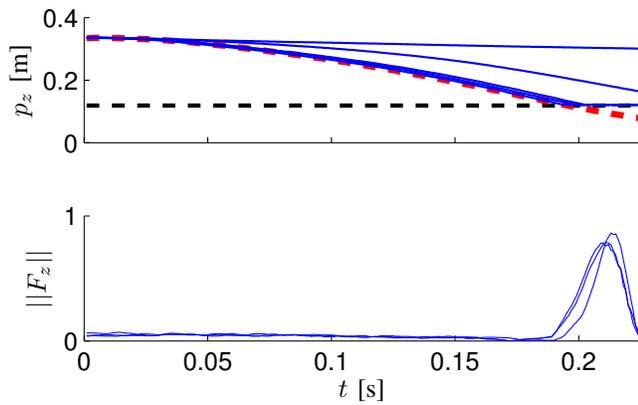


Fig. 6. Learning if dynamics while learning skill of hammering. Top plot shows the  $z$  motion in Cartesian space. The red dotted line is the desired trajectory, the blue lines shows the learning iterations and the black dotted line is the table height. Bottom plot shows the estimated impact force.

motion and the bottom plot shows the estimated interaction forces. Since no dynamical model was used, the robot had to learn the complete dynamical model of the hand and the tool, i. e. the hammer. This also explains why the robot was not able to track the desired trajectory in the first two iterations. Once the TPs were properly adapted to the task dynamic, the tracking error was small up to the point of contact with environment. Since low impedance control with CMPs was used, the interaction force is mainly a result of inertia of the robot and the hammer.

## V. CONCLUSIONS

We have showed that the proposed approach (CMPs) can successfully learn both the kinematic trajectory in Cartesian space encoded as a DMP and the corresponding dynamics encoded as a TP. The main contribution of this paper is a new approach for autonomously learning previously unknown TPs based on a given DMP and storing them in a database. Moreover, we propose to exploit previous experiences and statistical learning to accelerate the learning of TPs. We demonstrated that this way we can significantly improve the rate of learning and rapidly expand the database of TPs/CMPs. The proposed approach significantly improves the tracking accuracy of the low gain impedance control. Low gain impedance control implies low impact forces in case of unforeseen collisions, which makes robot safer for working in unstructured environment or with humans. As such, the proposed approach enables simple and computational inexpensive control of dynamically challenging tasks. The obtained results can be related to the findings of studies on human sensorimotor learning abilities.

## REFERENCES

[1] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, no. 4, pp. 319–40, Nov. 2011.  
 [2] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 79–91, June 2004.

[3] A. Ude, C. G. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robotics and Autonomous Systems*, vol. 47, pp. 93–108, 2004.  
 [4] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1463–1467, 2008.  
 [5] S. Schaal, P. Mohajerian, and A. Ijspeert, "Dynamics systems vs. optimal control—a unifying view," *Progress in brain research*, vol. 165, no. 1, pp. 425–45, Jan. 2007.  
 [6] A. Albu-Schaffer and G. Hirzinger, "Cartesian impedance control techniques for torque controlled light-weight robots," *Proceedings 2002 IEEE International Conference on Robotics and Automation*, pp. 657–663, 2002.  
 [7] S. Haddadin, A. Albu-Schaffer, and G. Hirzinger, "Requirements for Safe Robots: Measurements, Analysis and New Insights," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, Aug. 2009.  
 [8] J. Ulmen and M. Cutkosky, "A robust, low-cost and low-noise artificial skin for human-friendly robots," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4836–4841, 2010.  
 [9] D. Shin, I. Sardellitti, Y. L. Park, O. Khatib, and M. Cutkosky, "Design and Control of a Bio-inspired Human-friendly Robot," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 571–584, Nov. 2009.  
 [10] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 820–833, Apr. 2011.  
 [11] A. Gams and T. Petrič, "Adapting periodic motion primitives to external feedback: Modulating and changing the motion," in *Robotics in Alpe-Adria-Danube Region (RAAD), 2014 23rd International Conference on*, 2014, pp. 1–6.  
 [12] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors," *Neural computation*, vol. 25, pp. 328–73, 2013.  
 [13] D. Basa and A. Schneider, "Movement Primitives Learning Point-to-Point Movements on an Elastic Limb Using Dynamic Movement Primitives," *Robotics and Autonomous Systems*, 2015.  
 [14] N. Gopalan, M. P. Deisenroth, and J. Peters, "Feedback error learning for rhythmic motor primitives," *2013 IEEE International Conference on Robotics and Automation*, pp. 1317–1322, May 2013.  
 [15] D. M. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural networks : the official journal of the International Neural Network Society*, vol. 11, no. 7-8, pp. 1317–29, Oct. 1998.  
 [16] H. Gomi and M. Kawato, "Neural network control for a closed-loop System using Feedback-error-learning," pp. 933–946, Jan. 1993.  
 [17] J. W. Krakauer, M. F. Ghilardi, and C. Ghez, "Independent learning of internal models for kinematic and dynamic control of reaching," *Nature neuroscience*, vol. 2, no. 11, pp. 1026–31, Nov. 1999.  
 [18] E. Oztop, M. Kawato, and M. Arbib, "Mirror neurons and imitation: A computationally guided review," *Neural Networks*, vol. 19, pp. 254–271, 2006.  
 [19] D. Bruno, S. Calinon, and D. G. Caldwell, "Learning adaptive movements from demonstration and self-guided exploration," in *4th International Conference on Development and Learning and on Epigenetic Robotics*. IEEE, Oct. 2014, pp. 101–106.  
 [20] 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, Oct. 2010.  
 [21] A. Ude, B. Nemeč, T. Petrič, and J. Morimoto, "Orientation in Cartesian space dynamic movement primitives," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, no. 3, pp. 2997–3004, 2014.  
 [22] M. Denisa, T. Petrič, T. Asfour, and A. Ude, "Synthesizing Compliant Reaching Movements by Searching a Database of Example Trajectories," *International Conference on Humanoid Robots*, pp. 540–546, 2013.  
 [23] M. Kawato, "Feedback-Error-Learning Neural Network for Supervised Motor Learning," in *Advanced Neural Computers*, R. Eckmiller, Ed. North-Holland: Elsevier, 1990, pp. 365–372.  
 [24] L. Colasanto, N. G. Tsagarakis, and D. G. Caldwell, "A compact model for the compliant humanoid robot COMAN," in *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechanics*, 2012, pp. 688–694.