Autonomous Learning of Internal Dynamic Models for Reaching Tasks

Tadej Petrič, Aleš Ude and Auke J. Ijspeert

Abstract The paper addresses the problem of learning internal task-specific dynamic models for a reaching task. Using task-specific dynamic models is crucial for achieving both high tracking accuracy and compliant behaviour, which improves safety concerns while working in unstructured environment or with humans. The proposed approach uses programming by demonstration to learn new task-related movements encoded as Compliant Movement Primitives (CMPs). CMPs are a combination of position trajectories encoded in a form of Dynamic Movement Primitives (DMPs) and corresponding task-specific Torque Primitives (TPs) encoded as a linear combination of kernel functions. Unlike the DMPs, TPs cannot be directly acquired from user demonstrations. Inspired by the human sensorimotor learning ability we propose a novel method which autonomously learns task-specific TPs, based on a given kinematic trajectory in DMPs.

Keywords Compliant movement primitives · Task-specific dynamics · Learning · Dynamic movement primitives

1 Introduction

Learning of a new motor behaviour is one of the key skills that a humanoid robot should have. Programming by demonstration is a popular way to acquire new motor behaviour [1], which can be done by using different sensory systems, i.e. visual [2] or kinaesthetic guidance [3]. The main advantage is that the robot kinematics is

T. Petrič (✉) · A.J. Ijspeert
Biorobotics Laboratory, EPFL, École Polytechnique Fédérale de Lausanne,
Lausanne, Switzerland
e-mail: tadej.petri@epfl.ch

T. Petrič · A. Ude
JSI, Jozef Stefan Institute, Ljubljana, Slovenia
e-mail: ales.ude@ijs.si

T. Borangiu (ed.), *Advances in Robot Design and Intelligent Control*,
Advances in Intelligent Systems and Computing 371,
DOI 10.1007/978-3-319-21290-6_44

adapted and the posture is preserved even for redundant robots. A common way of learning kinematic trajectories is by using the Dynamic Movement Primitives (DMPs) [4]. To execute the desired DMP trajectory on a robot the underlying controller, usually based on high gain feedback loop, is employed to guarantee that the trajectory is accurately executed. However, using high gains in a feedback loop makes robots inherently unsafe for interaction with the environment or humans, due to the high interaction forces that may occur during unforeseen contacts.

Different approaches can be used to minimize the interaction forces during contacts: e.g., by actively detecting them using proximity sensors or artificial skin [5]; by using passively compliant mechanical structures like artificial pneumatic muscles [6] or by using control algorithms for active compliance [7]. The latter is achieved using the control approaches based on the inverse dynamic models. Obtaining a dynamical model is a challenging and time consuming task even for experts. But it only needs to be done once. On the other hand, it is impossible to obtain the generic dynamic model, even for simple task like table wiping, due to the unknown friction between the sponge and the table. Therefore, instead of modelling the task, we propose a novel method for learning yet unknown and undefined task-specific torque primitives. The proposed method would replace the need for modelling the task dynamics and if the dynamic model is already in use, it will compensate for additional uncertainties. By learning the exact dynamic model of the task, one can achieve both the compliant robot behaviour and the accurate tracking of the desired motion. While accurate tracking is required for proper task execution, the compliant behaviour is an essential feature for safe interaction with the environment or with a human [8].

While the DMPs and their possible modulations during contact with the environment have been thoroughly analysed [9–12], their extension towards torque controlled robots was not sufficiently explored yet. Inspired by the human sensory motor ability [13–16] we propose a novel method based on Compliant Movement Primitives (CMPs), which represent a new model-free approach, while keeping simple and smooth modulation properties of the position based DMPs. Similar to what can be observed in humans [14], the proposed approach independently learns the kinematic trajectory in Cartesian space (DMPs) and the task dynamics in the joint space (TPs). The proposed approach exploits the motion pattern encoded by the DMPs to acquire the task-specific torques and encodes them as TPs.

The first step in gaining the CMPs is learning the desired motion trajectory in DMP. Next, the DMP trajectory is executed on a robot using low-feedback gains with the proposed method for learning the task-specific TPs. In each subsequent step the TPs are then used as a feedforward term, which essentially represents new learned task-specific dynamics. The stability of the proposed controller is assured by keeping the low-gain feedback loop, which ensures that the desired task is performed safely even in unstructured environment or with humans. While the proposed approach eliminates the need for dynamical modelling, the CMPs still have to learn TPs for each task variation. Once several CMPs for different task variations are learned, they can be added to a database and statistical generalization can be used as in [17]. This allows learning and performing different variations of the same task in a compliant manner without the need of any analytical models of the task or programming experts.

2 Compliant Movement Primitives

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

$$\boldsymbol{h}(t) = [\ddot{\boldsymbol{y}}_d(t), \dot{\boldsymbol{y}}_d(t), \boldsymbol{y}_d(t), \boldsymbol{\tau}_f(t)] \tag{1}$$

Here $\ddot{\boldsymbol{y}}_d(t), \dot{\boldsymbol{y}}_d(t), \boldsymbol{y}_d(t)$ are vectors of the desired task-space accelerations, velocities and positions respectively encoded in the DMPs, and $\boldsymbol{\tau}_f(t)$ is the vector of corresponding task-specific feedforward joint torques encoded in TPs.

First the kinematic part is obtained using the learning by demonstration approach. A short overview of the periodic version of the DMPs [4, 12] with the recursive learning algorithm is given first. The following equations are valid for one DOF; for multiple DOF the equations are used in parallel. A nonlinear system of differential equations defines DMP for periodic movements

$$\dot{z} = \Omega\big(\alpha_z\big(\beta_z(g - y) - z\big) + f(\phi)\big), \quad \dot{y} = \Omega z, \quad \dot{\phi} = \Omega, \tag{2}$$

where Ω is the frequency of motion, ϕ is the phase, α_z and β_z are the positive constants set to 8 and 2 respectively. They ensure critical damping so that the system monotonically converges to the trajectory oscillating around anchor point g. The nonlinear part $f(\phi)$ is given by

$$f(\phi) = \frac{\sum_{i=1}^{N} \psi_i w_i}{\sum_{i=1}^{N} \psi_i}, \quad \psi_i = \exp(h(\cos(\phi - c_i) - 1)), \tag{3}$$

where w_i are the weights that define the shape of the trajectory, ψ_i are the Gaussian like kernel functions where parameter h defines their width, and c_i is equally spaced between 0 and 2π. To acquire the target signal for recursive learning the Eq. (2) is rewritten in the form

$$f_t(t) = \frac{\ddot{y}_d}{\Omega^2} - \alpha_z(\beta\_z(g - y_d) - \frac{\dot{y}_d}{\Omega}), \tag{4}$$

where $\ddot{y}_d$, $\dot{y}_d$, y_d are respectively the desired acceleration, velocity and position of the desired trajectory. To update the weights w_i, an incremental regression is used:

$$w_i(t+1) = w_i(t) + \psi_i P_i(t+1) e(t), \tag{5}$$

$$P_i(t+1) = \frac{1}{\lambda}\left(P_i(t) - \frac{P_i^2(t)}{\frac{\lambda}{\psi_i} P_i(t)}\right), \tag{6}$$

$$e(t) = f_t(t) - w_i(t), \tag{7}$$

where P_i is the covariance. The initial parameters are set to $P_i = 1$, $w_i = 0$ and $\lambda = 0.995$ (see [10] for details).

To obtain the dynamical part of the CMP, i.e. the corresponding TP, the motion is executed on a robot using a low gain feedback loop and a learning algorithm for acquiring the task specific torque primitive (TP). The desired motion $\ddot{y}_d$, $\dot{y}_d$, y_d encoded in DMPs is executed using the following control law

$$\boldsymbol{\tau}_u = \mathbf{J}^{\#}\left(\mathbf{K}_p(\boldsymbol{y}_d - \boldsymbol{y}) + \mathbf{K}_d(\dot{\boldsymbol{y}}_d - \dot{\boldsymbol{y}}) + \mathbf{K}_i(\ddot{\boldsymbol{y}}_d - \ddot{\boldsymbol{y}})\right) + \mathbf{N}\mathbf{K}_n\dot{\boldsymbol{q}} + \boldsymbol{\tau}_f(\phi), \tag{8}$$

where, $\mathbf{J}^{\#}$ is the pseudo inverse, $\mathbf{N}$ is the null-space matrix 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 feedback gain requirements.

Here $\boldsymbol{\tau}_f(\phi)$ is the feedforward term encoded in TPs. For one DOF it is given by

$$\tau_f(\phi) = \frac{\sum_{i=1}^{N} \psi_i w_i}{\sum_{i=1}^{N} \psi_i}, \tag{9}$$

where the kernel functions ψ_i are defined as in Eq. (3) and the weights are updated by inserting the error given by

$$\boldsymbol{e}(t) = \mathbf{J}^{\#}(\alpha_t(y_d(t) - y(t)) + \beta_t(\dot{y}_d(t) - \dot{y}(t))) \tag{10}$$

into Eqs. (5)–(6). The rate of learning is defined by setting parameters α_t and β_t.

3 Experimental Evaluation

The proposed method was evaluated on a 3-DOF planar robot simulated in Matlab using a Planar Manipulator Toolbox [18]. The simulation step was set to 0.01 s and the link lengths of the robot were set to 1 m. The initial task space position of the robot was $y_0 = [1, 1.5]$ m. The initial robot configuration (green line) is shown in Fig. 1.

The task was to reach towards certain points in space and return back to the initial position. A proposed controlled given by Eq. (8) with low feedback gains was used. Note that inverse dynamical model for compensating the robot dynamics was not used. This task is also similar to the task used in a study of human sensory

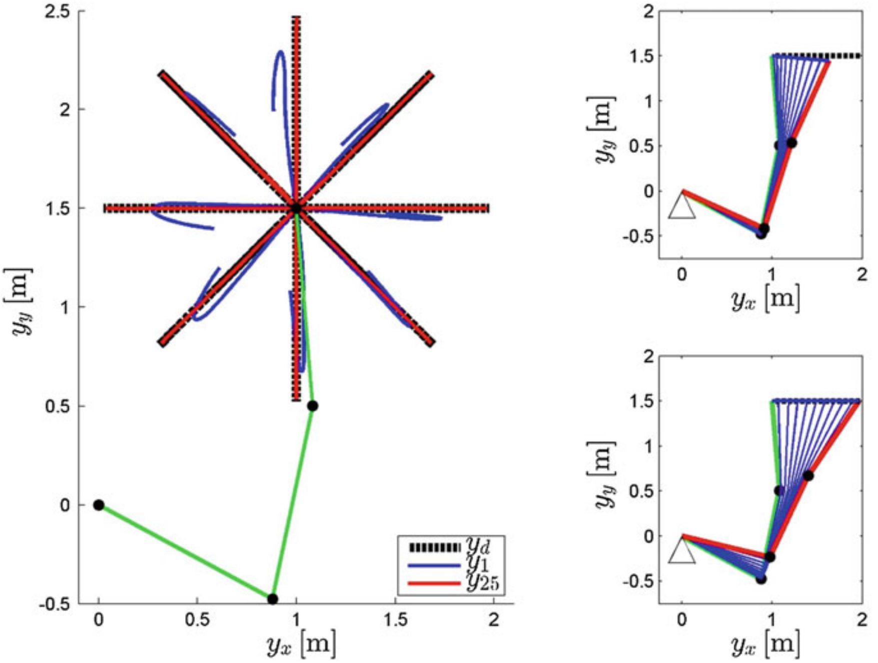

Fig. 1 Left hand side plot shows eight examples of the initial (*blue*) and the learned (*red*) robot behaviour in Cartesian space. Right hand side plots show one detailed example of the initial (*top*) and the learned (*bottom*) robot behaviour in Cartesian space

motor learning ability reported in [14]. To compare the results we chose eight different positions equally spaced on a circle with a radius of 1 m from the initial position. The desired Cartesian kinematic trajectories $\boldsymbol{y}_d$ were encoded in DMPs as a part of the CMPs. They are shown with black dotted lines on the left plot in Fig. 1.

Note that the desired kinematic trajectory is encoded in the task-space, i.e. Cartesian space, and the corresponding task-specific torques are encoded in the joint space, see Eq. (8). The mapping of the task-space error $\boldsymbol{e}$(t) to the joint space is done by using a pseudo-inverse of the Jacobean. The learning of the TPs was performed after each iteration of the executed motion. Note that each executed motion started from the same initial position. The left hand side plot in Fig. 1 shows comparatively eight different examples between initial movement execution (blue lines) and the last iteration of learning (red line). One can see in all eight examples that the proposed approach was able to learn proper TPs in order to execute the task accurately. Note that in this example the proposed system was learning the unknown robot dynamics.

The successful learning of the torque primitives can be seen on the right hand side plots in Fig. 1 where we compare the behaviour of one representative example, i.e. moving to the right. For clarity we only show the first half of the movement. The top right plot shows the robot's behaviour for initial movement execution, i.e.

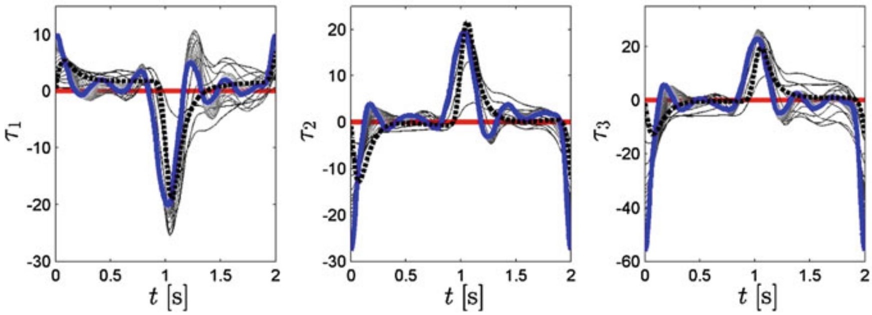

Fig. 2 Comparison with an ideal inverse dynamic model and evolution of the TPs during 25 learning iterations for all three joints. The initial TPs are in *red*, the final TPs are in *blue* and the ideal joint torques using exact dynamic models are shown with *black dotted line*

feedforward torques were zero. Here we can clearly see that the low gain feedback loop is not able to track the desired trajectory accurately. However, by applying the proposed learning algorithm to obtain the TPs, the tracking performance was improved significantly. The robot's behaviour after learning, i.e. after 25th iterations, is shown in the bottom plot, where perfect matching between the desired (black dotted line) and actual (blue line) Cartesian position can be observed.

The evolution of the TPs for all 25 iterations and for all three joints is shown in Fig. 2, were the red line shows the initial TPs, the thick blue line shows the final TPs and the black dotted line shows the ideal task-specific torques computed by using the exact inverse dynamic model.

Note that the initial torque primitives are zero. By comparing the TPs with the ideal dynamical model, we can see that they are similar in shape and amplitude. This shows that the proposed algorithm can successfully learn the required task-specific dynamics autonomously.

The proposed learning algorithm essentially minimises the error function given by Eq. (10). Thus, when $\boldsymbol{e}(t)$ in Eq. (10) is converging towards 0, $\boldsymbol{\tau}_f(\phi)$ in Eq. (9) is converging toward the exact inverse dynamic model. Ideally, if $\boldsymbol{e}(t)$ in Eq. (10) is zero then $\boldsymbol{\tau}_f(\phi)$ in Eq. (9) is the perfect inverse dynamic model.

The convergence of the proposed algorithm and the ability to re-learn the task-specific torques is also illustrated on the left hand side plot in Fig. 3, where the mean value and the standard deviation of the tracking error for all eight examples are shown. In studies about humans it has been demonstrated that they can quickly adapt to the changes in the task dynamics [14]. To show that the proposed approach can also cope with changes of task dynamics, the parameters of robot dynamics were changed after the 25th iteration. In the 26th iteration the inertia parameter of the robot was increased by a factor five and remained afterwards at that new value.

In general the learning rate of the proposed algorithm is defined with the gains α_t and β_t. If the gains are set too high, the system may potentially be destabilized, or vice versa, if the gains are too low, the learning rate would be slow and impractical. We set them empirically to $\alpha_t = 20$ and $\beta_t = 10$, which results in similar learning rate

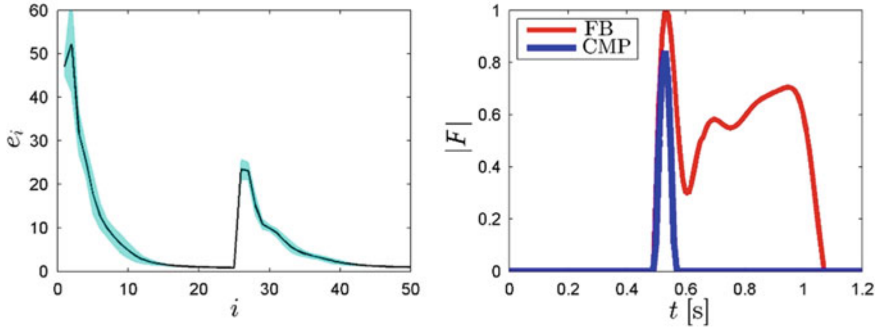

Fig. 3 Left plot shows the average error and the standard deviation (*shaded area*) for all eight examples. Right plot shows the interaction force in case of collision using the proposed control approach (*CMP*) and standard feedback (*FB*) control approach with high gains

as observed in human sensory motor learning ability reported in [14]. The results are shown on the right hand side plot in Fig. 3, where we can see that only 10 iterations were needed in order to obtain perfect tracking performance even after the dynamics was significantly altered, i.e. inertia parameters were increased by factor five. This also shows that the proposed approach can cope with sudden changes in the task dynamics and re-adapt online if necessary.

Since low feedback gains are used in combination with the CMPs we also assume that the interaction force in case of a collision with an unforeseen object will be significantly smaller compared to the general feedback approach with similar tracking performance. To investigate and compare the performance in case of unforeseen collisions, a solid wall was placed on the desired end-effector path. The results are shown on the right hand side plot in Fig. 3, where the normalized interaction force at the end effector is given. The blue line shows the results of the proposed approach and the red line shows the results of the high gain feedback control.

On the right hand side plot in Fig. 3 we can see that, at the beginning of the impact, resulting forces are similar in both cases. This was expected since the initial force is mainly a result of the robot inertia, i.e. the robot is suddenly forced to stop in both cases. This also explains why the forces are almost identical at the beginning of the impact. However, one can see that by using the proposed approach (CMPs), the impact force only appears initially and then goes towards zero. In this particulate example the normalized force impulse was only 0.035 Ns. On the contrary, in the case of the high gain feedback control, the force after impact remains rather large, i.e. it keeps almost the same value as it was at the beginning of the impact, resulting in a large force impulse of 0.33 Ns. In the latter case the force impulse was almost 10 times larger compared to the CMPs approach.

4 Conclusion

It was shown that the proposed approach (CMPs) can successfully learn both the kinematic trajectories in Cartesian space encoded with the DMPs and the corresponding torque primitives in the joint space encoded in TPs. The main contribution of the proposed approach is learning of the previously unknown task specific dynamics. With the proposed learning method, the control system was able to significantly improve the tracking accuracy in each of the subsequent iterations. Once the TPs are fully learned, the CMPs ensure accurate task execution and, at the same time, compliant robot behaviour. As such, the proposed learning framework enables simple and computationally inexpensive control of dynamically challenging tasks. Moreover, the results were also similar to human studies of sensorimotor learning abilities. Since learning of the task-specific dynamic is done autonomously by using a low gain feedback loop, the compliant behaviour is maintained during learning of task-specific torque primitives. This makes the robot safe for working in unstructured environments or with humans even during learning.

Acknowledgments The research activities leading to the results presented in this paper were supported by the Sciex-NMSCH project no. 14.069.

References

1. Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., Kawato, M.: Learning from demonstration and adaptation of biped locomotion. Robot. Auton. Syst. **47**(2–3), 79–91 (2004)
2. Ude, A., Atkeson, C.G., Riley, M.: Programming full-body movements for humanoid robots by observation. Robot. Auton. Syst. **47**, 93–108 (2004)
3. Hersch, M., Guenter, F., Calinon, S., Billard, A.: Dynamical system modulation for robot learning via kinaesthetic demonstrations. IEEE Trans. Rob. **24**(6), 1463–1467 (2008)
4. Schaal, S., Mohajerian, P., Ijspeert, A.: Dynamics systems vs. optimal control—a unifying view. Prog. Brain Res. **165**(1), 425–445 (2007)
5. Ulmen, J., Cutkosky, M.: A robust, low-cost and low-noise artificial skin for human-friendly robots. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 4836–4841 (2010)
6. Shin, D., Sardellitti, I., Park, Y.L., Khatib, O., Cutkosky, M.: Design and control of a bio-inspired human-friendly robot. Int. J. Robot. Res. **29**(5), 571–584 (2009)
7. Buchli, J., Stulp, F., Theodorou, E., Schaal, S.: Learning variable impedance control. Int. J. Robot. Res. **30**(7), 820–833 (2011)
8. Haddadin, S., Albu-Schaffer, A., Hirzinger, G.: Requirements for safe robots: measurements, analysis and new insights. Int. J. Robot. Res. **28**(11–12), 1507–1527 (2009)
9. Basa, D., Schneider, A.: Movement primitives learning point-to-point movements on an elastic limb using dynamic movement primitives. Robot. Auton. Syst. **66**, 55 (2015)
10. Gams, A., Petrič, T.: Adapting periodic motion primitives to external feedback: modulating and changing the motion. In: 23rd International Conference on Robotics in Alpe-Adria-Danube Region (RAAD), pp. 1–6 (2014)

11. Gopalan, N., Deisenroth, M.P., Peters, J.: Feedback error learning for rhythmic motor primitives. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1317–1322 (May 2013)
12. Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S.: Dynamical movement primitives: learning attractor models for motor behaviors. Neural Comput. **25**, 328–373 (2013)
13. Gomi, H., Kawato, M.: Neural network control for a closed-loop system using feedback-error-learning. Neural Networks **6**, 933 (1993)
14. Krakauer, J.W., Ghilardi, M.F., Ghez, C.: Independent learning of internal models for kinematic and dynamic control of reaching. Nat. Neurosci. **2**(11), 1026–1031 (1999)
15. Oztop, E., Kawato, M., Arbib, M.: Mirror neurons and imitation: a computationally guided review. Neural Networks **19**, 254–271 (2006)
16. Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control. Neural Networks J. Int. Neural Network Soc. **11**(7–8), 1317–1329 (1998)
17. Ude, A., Gams, A., Asfour, T., Morimoto, J.: Task-specific generalization of discrete and periodic dynamic movement primitives. IEEE Trans. Rob. **26**(5), 800–815 (2010)
18. Žlajpah, L.: Simulation in robotics. Math. Comput. Simul. **79**, 879–897 (2008)