2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
September 14-18, 2014. Chicago, IL, USA,

Online Learning of Task-Specific Dynamics for Periodic Tasks

Tadej Petrič, Andrej Gams, Leon Žlajpah and Aleš Ude

Abstract— **In this paper we address the problem of accurate trajectory tracking while ensuring compliant robotic behaviour for periodic tasks. We propose an approach for on-line learning of task-specific dynamics, i.e. task specific movement trajectories and corresponding force/torque profiles. The proposed control framework is a multi-step process, where in the first step a human tutor shows how to perform the desired periodic task. A state estimator based on an adaptive frequency oscillator combined with dynamic movement primitives is employed to extract movement trajectories. In the second step, the movement trajectory is accurately executed in the controlled environment under human supervision. In this step, the robot is accurately tracking the acquired movement trajectory, using high feedback gains to ensure accurate tracking. Thus it can learn the corresponding force/torque profiles, i. e. task-specific dynamics. Finally, in the third step, the movement is executed with the learned feedforward task-specific dynamic model, allowing for low position feedback gains, which implies compliant robot behaviour. Thus, it is safe for interaction with humans or the environment. The proposed approach was evaluated on a Kuka LRW robot performing object manipulation and crank turning.**

I. INTRODUCTION

Many commercially available robots are specialised for different tasks, from entertainment [1] to house keeping [2]. However, general purpose robots capable of performing different tasks to help people in their natural environment are still not possible. Such robots will have to posses the ability to quickly learn new skills either autonomously or with the help of a human tutor. The issue of autonomous skill learning remains one of the challenges of robotics research [3].

In the last decade many algorithms for machine learning were proposed and later adopted in robotics. One of the main reasons for their applicability in robotics is that they can be used to learn complex models. It has been shown that machine learning algorithms can successfully be used to acquire kinematics [4] and dynamic [5] models, for locomotion tasks [6] or even complex behaviors such as ball-in-a-cup game [7]. Despite significant improvement in learning algorithms, the necessity to acquire large amounts of training data and long learning times remain their main drawbacks. An extensive review for model learning in robotics was recently published by Nguyen-Tuong and Peters [8].

For improving learning and control performance, researches proposed different biologically inspired methods for robot control. An extensive review, covering methods from optimal feedback control [9] to forward models and predictive control [10], was recently published by Franklin and Wolpert [11]. One possible approach to biologically inspired robotic control is the use of central pattern generators for the control of periodic tasks [12]. Central pattern generators are neural circuits capable of producing coordinated motions while receiving only simple input signals [12]. Their applicability was shown on different robots and tasks such as, for example, swimming and walking with a robotic salamander [12] or walking with humanoid robots [13], [14].

In this paper we propose a novel control framework by combining central pattern generators with task-specific kinematic and dynamic profiles encoded with dynamic movement primitives (DMP) and dynamic torque primitives (DTP), respectively. The proposed framework is build on the two-layered motion imitation system presented in [15] and [16]. We propose to augment this system with dynamic torque primitives (DTPs), which can learn and repeat task-specific dynamics when performing a desired periodic task. The applicability of the approach is shown for two periodic tasks, i. e. manipulating objects and crank-turning as shown in Fig. 1.

The proposed algorithm is comprised of three steps. In the first step, it learns a task-specific motion trajectory with a standard DMP. In the second step, the task-specific dynamics is learned with the DTP. In the last step, the task is executed with the DMP providing the reference trajectory. Since DTP can provide feed-forward torques, the trajectory can be accurately tracked with low position feedback gains, thus ensuring compliant behaviour. Due to compliance the robot can safely interact with the environment or with humans.

The paper is organized as follows. In Section II, we give a detailed description of the proposed learning system. In Section III we evaluate the proposed approach in real-world tasks including manipulating an object and crank-turning.

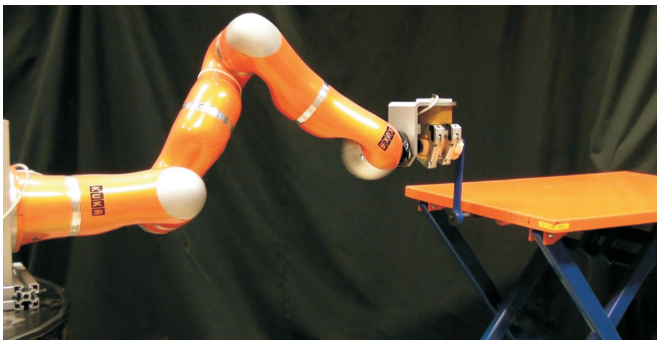

Fig. 1. Experimental setup for the crank-turning.

T. Petrič, A. Gams, L. Žlajpah and A. Ude are with the Department of Automatics, Biocybernetics and Robotics and with the Humanoid and Cognitive Robotics Lab, Jožef Stefan Institute, Ljubljana, Slovenia. {tadej.petric, andrej.gams, leon.zlajpah, ales.ude} at ijs.si

Conclusions and summary are given in Section IV.

II. TASK-SPECIFIC MOTION AND DYNAMICS LEARNING

As explained above, the ability to work in an unstructured environment including possible contacts with humans implies compliant robot behaviour. Moreover, autonomous robots must be able to quickly acquire new motor skills, including skills that involve contact with the environment. To satisfy these goals, we propose a novel learning framework, which is based on an adaptive frequency oscillator [16] combined with DMPs [17] and DTPs. Incremental updates of the available models can be computed in real time to achieve rapid learning of both DMPs and DTPs. The proposed control system is a multi-layered three-step process, which in the last step essentially acts as a task-specific model-based controller.Thus in the execution step the robot remains compliant and any collision forces that might arise remain small.

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

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

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are the joint positions, velocities and accelerations, respectively, $\mathbf{H}(\boldsymbol{q})$ is the inertia matrix, $\mathbf{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ are the Coriolis and centripetal forces, $\boldsymbol{g}(\boldsymbol{q})$ are the gravity forces and $\boldsymbol{\varepsilon}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ are the nonlinearities not considered in the rigid body dynamics, e. g. friction. The inverse dynamic model of the robot (1) is denoted as $\boldsymbol{f}_{dynamic}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$. In our experiments we used Kuka LWR arm with integrated inverse dynamic model $\boldsymbol{f}_{dynamic}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ of the robot. Note that this inverse dynamic model does not include any task-specific influences on the dynamics of the robot. In general, the commanded torque for joint specific impedance control is given with

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

where $\mathbf{K}$ and $\mathbf{D}$ are matrices of the desired stiffness and damping, respectively [18] and $\boldsymbol{\tau}_f$ is the vector of additional, task-specific torques. If the diagonal elements of $\mathbf{K}$ are high, then the robot is stiff and can accurately track the desired joint trajectories $\boldsymbol{q}_d$. Vice versa, if the diagonal elements of $\mathbf{K}$ are low, then the robot is compliant but cannot accurately track the desired trajectory unless the feed-forward torques $\boldsymbol{\tau}_f$ can be provided to improve the tracking quality. $\boldsymbol{\tau}_f$ can be specified by modelling the dynamics of the task or by learning as in our approach. The integration of the learnt $\boldsymbol{\tau}_f$ into DMP framework to simultaneously ensure compliant behaviour and accurate tracking is the main contribution of this paper.

In the following we explain the three steps of the proposed learning approach: A) learning of task-specific trajectories, B) learning of task specific dynamics and C) execution of the movement with accurate trajectory tracking and compliant behaviour.

A. Learning of task-specific trajectories

In the first step, the aim is to learn task specific trajectories of motion (positions) demonstrated by a human tutor performing the desired periodic task. To learn the motion trajectory demonstrated by a human tutor in real-time, we use adaptive frequency oscillators combined with DMPs. It was shown in [15], [16] that this setup can successfully extract the frequency of motion and learn the desired waveform in real-time. The following paragraphs provide a short recap of the algorithm.

The motion learning algorithm consists of two layers. The first layer, which extracts the motion frequency is based on an adaptive frequency oscillator combined with an adaptive Fourier series [16]. It is a second order system of differential equations

$$\dot{\boldsymbol{\Omega}} = -\mathbf{PE}\sin(\boldsymbol{\phi}), \tag{3}$$

$$\dot{\boldsymbol{\phi}} = \boldsymbol{\Omega} - \mathbf{PE}\sin(\boldsymbol{\phi}), \tag{4}$$

where $\boldsymbol{\Omega}$ is the vector of extracted frequencies, ϕ is the vector of phases, $\mathbf{P}$ is the diagonal matrix of the coupling constants and $\mathbf{E}$ is the diagonal matrix with the diagonal values given by the difference between the actual joint position $\boldsymbol{q}$ and the estimated joint positions $\hat{\boldsymbol{q}}$. The vector of the estimated joint positions is defined as $\hat{\boldsymbol{q}} = [\hat{q_1},\hat{q_2},...,\hat{q_i}]^T$, where i denotes the joint index. Each $\hat{q_i}$ is given by

$$\hat{q_i} = \sum_{c=0}^{m} (\alpha_{i,c}\cos(c\phi_i) + \beta_{i,c}\sin(c\phi_i)). \tag{5}$$

Here, m denotes the size of the Fourier series. The weights $\alpha_{i,c}$ and $\beta_{i,c}$ are updated according to the following learning rule

$$\dot{\alpha}_{i,c} = \eta\cos(c\phi_i)e_i, \tag{6}$$

$$\dot{\beta}_{i,c} = \eta\sin(c\phi_i)e_i, \tag{7}$$

where e_i is the diagonal element $E_{i,i}$ of matrix $\mathbf{E}$ and η is the learning constant (see [16] for details). If not stated otherwise, the size of the Fourier series was set to $m = 10$ in our experiments.

The second layer of the motion learning system ensures the proper waveform of the output signal. The output signal is encoded as a DMP. The equations of DMPs summarised from [15], [19], [20] are

$$\dot{\boldsymbol{r}} = -\mathrm{diag}(\boldsymbol{\Omega})\left(\mathbf{A}\left(\mathbf{B}\boldsymbol{p} + \boldsymbol{r}\right) + \boldsymbol{f}(\phi)\right), \tag{8}$$

$$\dot{\boldsymbol{p}} = \mathrm{diag}(\boldsymbol{\Omega})\boldsymbol{r}, \tag{9}$$

where $\mathbf{A}$ and $\mathbf{B}$ are constant diagonal matrices which guarantee that the system monotonically converges to the desired trajectory, with $A_{i,i} = 8$ and $B_{i,i} = 2$, and $\boldsymbol{f}$ is the vector forcing term that determines the actual shape of the trajectory. The elements of $\boldsymbol{f}$ are given by

$$f_i(\phi) = \frac{\sum_{j=1}^{N} \psi_j(\phi) w_{i,j}}{\sum_{i=1}^{N} \psi_j(\phi)}, \tag{10}$$

where ψ are Gaussian-like kernel functions defined by

$$\psi_j(\phi) = \exp\left(h\left(\cos\left(\phi - c_j\right) - 1\right)\right). \tag{11}$$

Here, h is the width and c_j the centres describing the distribution of basis functions over one phase period. If not stated otherwise, we use c_j, $j = 1,\ldots,25$, equally spread between 0 and 2π.

By using locally weighted regression as in [15], the system can learn the shape of the trajectory on-line. The target trajectory of a single joint i is defined by

$$f_i = \frac{\ddot{q}_i}{\Omega_i^2} - A_{ii}\left(B_{ii} - q_i - \frac{\dot{q}_i}{\Omega_i}\right). \tag{12}$$

To update the weights $w_{i,j}$ for the kernel function ψ_j, we use the recursive least-squares method with the forgetting factor λ. The forgetting factor was apriori set to 0.995. With the given target for fitting (12), the recursive algorithm is updating the weights $w_{i,j}$ using the following rule

$$w_{i,j}(t+1) = w_{i,j}(t) + \Psi_j P_{i,j}(t+1) r_i(t) e_i(t), \tag{13}$$

$$e_i(t) = f_i(t) - w_{i,j}(t) r_i(t), \tag{14}$$

$$P_{i,j}(t+1) = \frac{1}{\lambda}\left(P_{i,j}(t) - \frac{P_{i,j}(t)^2 r_i(t)^2}{\frac{\lambda}{\Psi_j} + P_{i,j}(t) r_i(t)^2}\right). \tag{15}$$

If not stated otherwise, we use $w_{i,j}(0) = 0$ and $P_{i,j}(0) = 1$, where $i = 1,\ldots,\mathrm{DOF}$ and $j = 1,\ldots,25$.

Combining DMPs with adaptive frequency oscillators by anchoring the kernel functions to the phase signal $\boldsymbol{\phi}$ makes it possible to synchronise an arbitrary trajectory to an arbitrary periodic signal congruent with the desired task. Because the phase estimation in adaptive frequency oscillators and learning of the motion in DMPs are done simultaneously, all system delays are automatically included (see [16] for details).

B. Learning of task-specific dynamics

In the second step, the motion imitation system must learn the task-specific dynamics. The learning of the task-specific dynamics is done by accurately tracking the trajectory $\boldsymbol{p}$ that was learned in the first step. High position feedback gains $\mathbf{K}_h$, implying also strong reactions to perturbations, are used to achieve accurate DMP tracking. To perform the motion with constant frequency $\boldsymbol{\Omega}_d$, the error $\boldsymbol{E}$ in (4) is set to 0 and (4) transforms into

$$\dot{\boldsymbol{\phi}} = \boldsymbol{\Omega}_d. \tag{16}$$

This ensures that the motion is repeated with the last learned frequency of motion. Preserving the exact frequency of motion is important because dynamic trajectories are not the same if the frequency is changing.

To learn task-specific dynamics at the desired frequency, we use a similar set of differential equations DTPs as in the case of DMPs. The main difference is that for DTPs the target function for learning is defined as

$$f_{\tau,i} = \frac{\ddot{\tau}_i}{\Omega_n^2} - A_{ii}\left(B_{ii} - \tau_i - \frac{\dot{\tau}_i}{\Omega_i}\right), \tag{17}$$

where τ_i is the commanded torque signal of the selected joint. The second order differential equations for DTPs are this given as

$$\dot{\boldsymbol{u}} = -\mathrm{diag}(\boldsymbol{\Omega})\left(\mathbf{A}\left(\mathbf{B}\boldsymbol{v} + \boldsymbol{u}\right) + \boldsymbol{f}\right), \tag{18}$$

$$\dot{\boldsymbol{v}} = \mathrm{diag}(\boldsymbol{\Omega})\boldsymbol{u}, \tag{19}$$

where $\boldsymbol{v}$ is essentially the vector of the corresponding task-specific inverse dynamics, i. e. open-loop feed-forward torques. Function $\boldsymbol{f}$ is defined as in (10) and its free parameters can be estimated using locally weighted regression, i. e. Eq. (13) – (15).

C. Executing of the desired task

In the third step, the task is executed by using control algorithm (2), but replacing $\boldsymbol{q}_d$ with $\boldsymbol{p}$ from (8) – (9) and $\boldsymbol{\tau}_f$ with $\boldsymbol{v}$ from (18) – (19)

$$\boldsymbol{\tau}_u = \mathbf{K}_l(\boldsymbol{p} - \boldsymbol{q}) + \mathbf{D}(\dot{\boldsymbol{q}}) + \boldsymbol{v} + f_{dynamic}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}), \tag{20}$$

Here, $\boldsymbol{v}$ is the task specific feedforward torque and $f_{dynamic}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ is the robot inverse dynamic torque. Because task-specific dynamics is provided in a feedforward manner, the tracking accuracy will remain in the same range even if much lower position feedback gains are used than during training, i. e. $\mathbf{K}_l << \mathbf{K}_h$. By applying low position gains $\mathbf{K}_l$ we assure compliant robot behaviour. Note, that the tracking accuracy remains the same as long as the system is not perturbed.

The main idea used in the proposed approach is that feedforward compensation is used to assure the nominal behaviour of the robot for the given task even when low position feedback gains are used. In this way, we can assure good tracking and compliant behaviour.

III. PERFORMANCE EVALUATION

In this section we evaluate the performance of the proposed approach. The results of obtained by kinesthetically teaching Kuka LWR arm to perform the desired motion and then autonomously learning the corresponding task-specific dynamics are presented in the paper and the accompanying video.

We performed two separate experiments. In the first experiment, a human tutor was teaching the Kuka LWR arm how

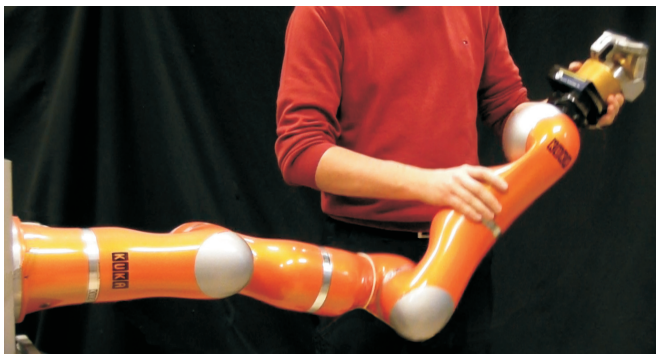

Fig. 2. Experimental setup for kinesthetically teaching the robot to perform desired periodic task.

to perform a periodic manipulation task. The experimental setup is shown in Fig. 2. For performance illustration and motion comparison, the robot was performing the same task but with different payloads: with an empty gripper (BarrettHand BH 262) and with the gripper holding a weight of 1.5 kg. Note that the payload influences the overall dynamic properties of the system.

Fig. 3 shows the results of the first step, where the tutor was kinesthetically guiding the robot to perform the desired task. We can see from the top left plot, where the extracted frequency Ω is shown, that the adaptive oscillator was able to extract the proper movement frequency which corresponds to the actual frequency of movement produced by the human through kinesthetic guidance. Extracting the proper frequency Ω of the task is crucial for the learning process because the adaptive oscillator also provides the phase signal ϕ, used to anchor the movement trajectory. The learning process for the kinematic trajectory will be effective only if a proper phase can be extracted and a stable limit cycle achieved. This is due to the fact that DMPs are driven by the frequency Ω and phase ϕ.

The bottom left plot shows the sum of square differences e_q between actual join positions $\boldsymbol{q}$ and the learned DMP trajectories $\boldsymbol{p}$. We can see that the error e_q approaches zero immediately after the estimated frequency converges towards the actual basic frequency of the motion, which was approximately 0.42 Hz. The performance of the learning process is also illustrated in the right plot where the actual joint trajectories $\boldsymbol{q}$ (solid lines) and DMP trajectories $\boldsymbol{p}$ (doted lines) are shown for one period for all seven joints. From the plots we can conclude that the motion trajectory learning was successful.

Once the system learns the motion trajectories using DMPs, it can proceed to the second step of learning the task-specific dynamics, where these motion trajectories are executed on the robot using a stiff tracking controller with high diagonal elements of $\mathbf{K}_h$. Since the robot is not compliant in this phase, the teacher must ensure that the robot does not encounter any obstacles, thereby preventing a possible damage to the robot. Examples of learning the task-specific dynamics for the same motion pattern and the two different payloads are shown in Fig. 4.

We can see in the top plot of Fig. 4 that DTPs successfully learned the task-specific dynamic trajectories in less than three motion periods. The learning of DTPs is faster than the learning of DMPs because the right frequency has already been extracted. The effectiveness of the learning process can be seen in the bottom plot, where both the actual commanded torque and the learned DTP trajectories are shown. Here we can see that they are, apart from the measurement noise, perfectly matched. By comparing bottom plots, we can also see that the dynamic trajectories are different for the same motion trajectories. This is as expected since the payload at the end effector was different in both cases.

Note that for the learning process, the torque trajectory was not filtered or processed in any other way. We can see that DTP trajectories are smooth. Smoothness of DTPs is ensured because they are encoded as a critically damped second order differential equation system (18) – (19), which

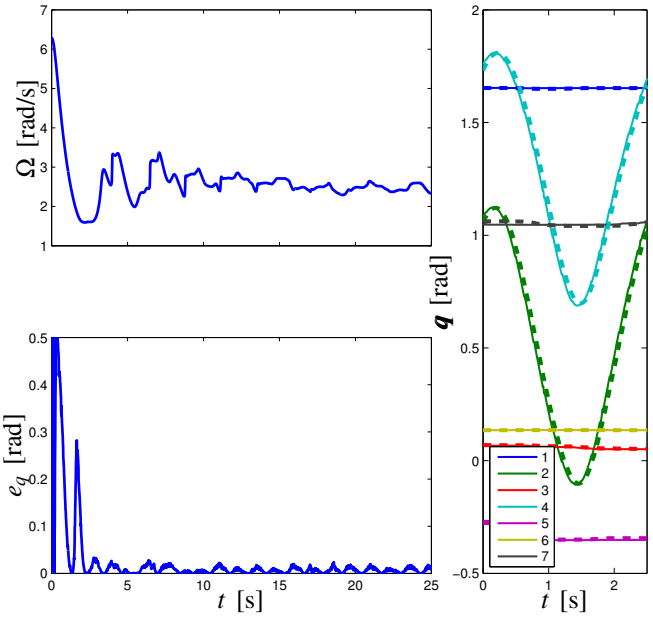

Fig. 3. On-line learning of the desired movement from kinestetic guiding. The top-left plot shows the extracted frequency Ω and the bottom-left plot the sum difference between the learned $\boldsymbol{p}$ and the actual $\boldsymbol{q}$ joint position. The right plot shows the actual $\boldsymbol{q}$ (solid lines) and the learned $\boldsymbol{p}$ (doted lines) positions in one period for all seven DOF.

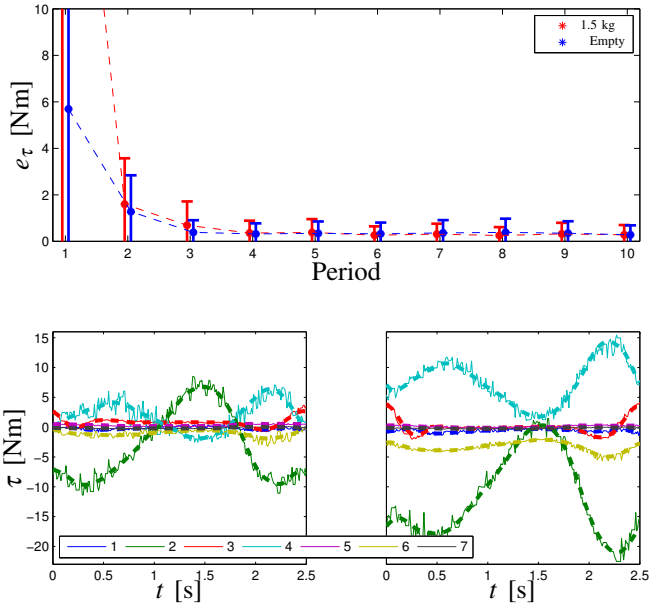

Fig. 4. Performance evaluation when learning task-specific dynamics. The top plot shows the mean and standard deviation of the sum of differences between the commanded torques $\boldsymbol{\tau}_u$ and the learned DTP trajectories $\boldsymbol{v}$ in one period. The blue colour shows the learning without any payload at the end effector and the red color shows the learning for the same motion but with a payload of 1.5kg. The bottom plots shows the commanded torques $\boldsymbol{\tau}_u$ (solid line) and the learned DTP trajectories $\boldsymbol{v}$ (doted line) in the 6th period. The bottom left plot shows the learned torques without a payload at the end-effector and the bottom right plot the learned torques with the payload at the end effector.

also assures smooth derivatives.

The comparison between different control states for the task of lifting an object held by a robot is shown in Fig. 5, where the sum of square tracking errors $e_q = \sum \|\boldsymbol{q}_d - \boldsymbol{q}\|^2$ and the exerted external force F_e is shown. There are four different control states in Fig. 5. For the first 15 periods we can see the task-specific dynamic learning, where the controller was set to accurately track the desired motion. As expected the tracking error was marginal. We can also see that in this step there were no external forces acting on the robot. This is necessary because this step can only be accomplished if there are no unexpected collisions. The second step between period 15 and 25 shows the performance with high gain position feedback control augmented with the feedforward task dynamics learned in the previous step. Here we can see that the tracking error remains small even if there are external forces acting on the robot. This may be a useful behavior under some circumstances, but dangerous for robots working together with humans.

If a robot is working in an unstructured environment or in contact with humans, accurate trajectory tracking using only feedback control may not be an optimal solution. In such cases the robot is stiff and any collisions with objects cause high impact forces that may damage the robot or even worse, the robot could harm people working with it. In such situations, the control system should on the one hand enable natural compliant behaviour and on the other hand a good tracking precision. The performance of the control system with compliant behaviour and feedforward task dynamics is shown form 25th to 45th period in Fig. 5. Here we can see that the impact forces when the robot collides with the environment are smaller because the robot moves away from the desired motion pattern when collisions occur. Nevertheless, we can see that the tracking accuracy remains excellent if no external forces present. The last part, from 45th to 50th period, shows the behaviour if feedforward dynamic model is not used, but the robot is compliant with low stiffness gain matrix $\mathbf{K}_l$. It is clear that with the low gain position feedback control and without the feedforward dynamic model, the robot can not perform the desired task at all, even when no external forces are present. The results are also supported by the behaviour exhibited by a robot in the first part of the accompanying video.

To further demonstrate the applicability of the proposed approach, we applied it to a crank turning task (see also Fig. 1). By turning the crank, the robot changed the height of the table, but at the same time also the center of crank rotation. To successfully perform this task, the robot must keep contact with the handle of the crank. To teach the robot how to perform this task, we first put the table upside down so that the center of the handle position was kept constant. The robot was then kinesthetically guided to turn the handle, and the pattern was learned with DMPs. The learned motion was then executed with accurate tracking controller for the purpose of learning task-specific dynamics with DTPs. Both, the actual joint values $\boldsymbol{q}$ and the DMPs $\boldsymbol{p}$ for motion learning and commanded torqes τ and DTPs $\boldsymbol{v}$ are shown in Fig. 6.

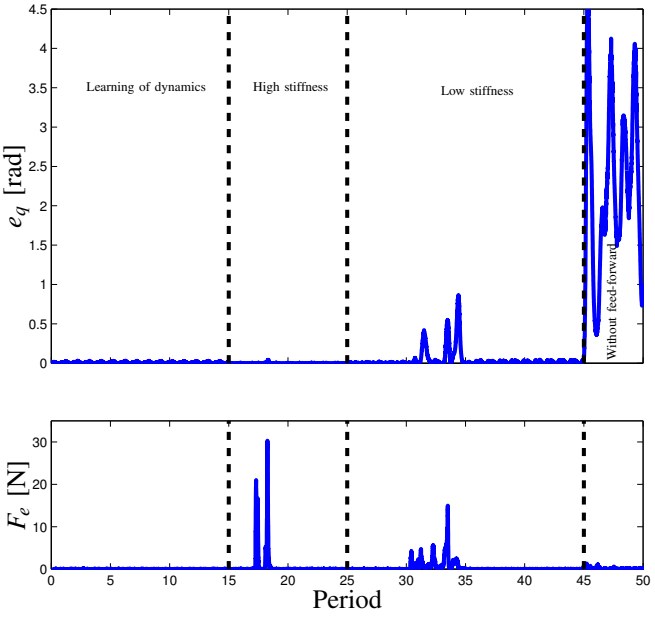

Fig. 5. Comparison between different control modes: learning task-specific dynamics with DTPs, accurate tracking with high position feedback loop, accurate tracking with low position feedback loop and feedforward DTPs, and low position feedback without using feedforward DTPs. The top plot shows the sum square tracking errors e_q and the bottom plot shows the exerted external force F_e.

Here we can see that the difference between the actual and learned values is minimal for both DMPs and DTPs.

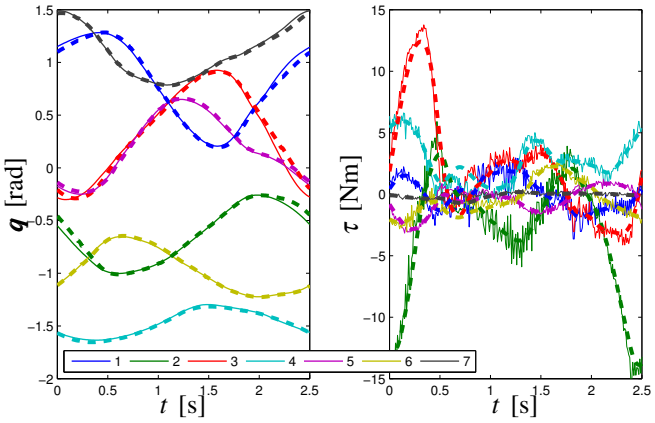

Fig. 6. Learning of the motion pattern with DMPs in the left plot and learning of the task-specific dynamics in the right plot. Solid lines show the actual values and dashed lines the DMPs and DTPs, respectively, for both plots.

Once the task-specific motion pattern and the dynamics are learned, we can use compliant robot control with low position gains $\mathbf{K}_l$ and feedforward dynamics. The advantage of compliant control in this specific task is the ability to automatically adapt to the vertical translation of the rotation center during crank-turning. If accurate tracking with high gain position control is used, high forces will appear very quickly. A comparison between accurate tracking controller with high position gains and compliant position control with the learned feedforward task-specific dynamics is shown in

Fig. 7. We can see in the top plot that the exerted forces were above safe working limits after 17th periods when a stiff controller was used. On the other hand, when using compliant control with feedforward DTPs, the robot was able to perform many more turns, altogether 40. We can also see that the force acting on the robot was lower and did not exceed safe working limits. The bottom plot of Fig. 7, where top positions in each period are displayed, shows that when compliant robot control with feedforward DTPs was used, the robot was able to adapt to the motion of the handle. On the other hand, if stiff control behaviour was used, adaptation was not possible. As expected, due to a certain degree of environment compliance, the top positions were in this case the same in all periods, but the forces increased significantly. This is also shown in the last part of the accompanying video.

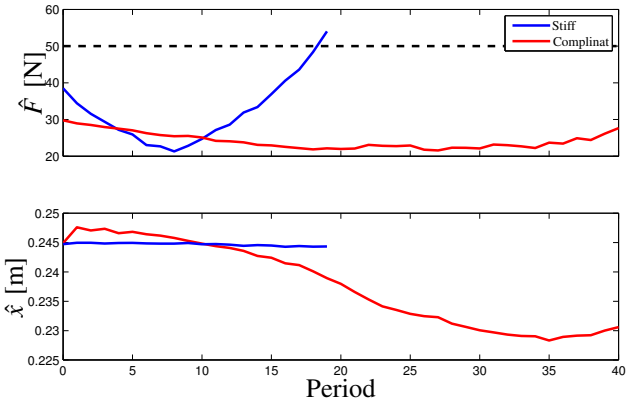

Fig. 7. Comparison between stiff and compliant control with feedforward task specific dynamics using DTPs. Top plot shows the peak force and the bottom plot the top handle position in each period.

IV. CONCLUSIONS

We proposed a new three-stage learning framework which can autonomously learn proper motion trajectories and the corresponding task dynamics to perform a desired task. The learning of motion patterns with DMPs and of task-specific dynamics with DTPs can be done in real time and without any additional signal processing methods. As such, the proposed learning framework enables simple and computationally inexpensive control in the case of dynamically challenging periodic tasks. The main contribution of our approach is that by learning the task specific dynamics we can ensure accurate task execution and at the same time compliant robot behaviour. In the future, we would like to generalise the approach for the aperiodic tasks and incorporate statistical learning methods to build a library of motions including task-specific dynamics.

ACKNOWLEDGMENT

This research has received funding from the European Communitys Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 600578, ACAT.

REFERENCES

[1] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of NAO humanoid," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, May 2009, pp. 769–774.

[2] J. Jones, "Robots at the tipping point: the road to iRobot Roomba," *IEEE Robotics & Automation Magazine*, vol. 13, no. 1, pp. 76–78, Mar. 2006.

[3] S. Schaal and C. Atkeson, "Learning Control in Robotics," *IEEE Robotics & Automation Magazine*, vol. 17, no. 2, pp. 20–29, June 2010.

[4] a. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, vol. 1, no. Iros. IEEE, 2001, pp. 298–303.

[5] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model Learning with Local Gaussian Process Regression," *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, Jan. 2009.

[6] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal, "Learning locomotion over rough terrain using terrain templates," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2009, pp. 167–172.

[7] B. Nemec, R. Vuga, and A. Ude, "Exploiting previous experience to constrain robot sensorimotor learning," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, Oct. 2011, pp. 727–732.

[8] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey." *Cognitive processing*, vol. 12, no. 4, pp. 319–40, Nov. 2011.

[9] C. G. Atkeson and J. M. Hollerbach, "Kinematic features of unrestrained vertical arm movements." *The Journal of neuroscience : the official journal of the Society for Neuroscience*, vol. 5, no. 9, pp. 2318–30, Sept. 1985.

[10] 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.

[11] D. W. Franklin and D. M. Wolpert, "Computational mechanisms of sensorimotor control." *Neuron*, vol. 72, no. 3, pp. 425–42, Nov. 2011.

[12] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review." *Neural networks : the official journal of the International Neural Network Society*, vol. 21, no. 4, pp. 642–53, May 2008.

[13] L. Righetti, "Programmable central pattern generators: an application to biped locomotion control," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, no. May. IEEE, 2006, pp. 1585–1590.

[14] 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.

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

[16] T. Petric, A. Gams, A. J. Ijspeert, and L. Zlajpah, "On-line frequency adaptation and movement imitation for rhythmic robotic tasks," *The International Journal of Robotics Research*, vol. 30, no. 14, pp. 1775–1788, Sept. 2011.

[17] A. Gams, T. Petric, A. Ude, and L. Žlajpah, "Performing Periodic Tasks: On-Line Learning, Adaptation and Synchronization with External Signals," in *The Future of Humanoid Robots - Research and Applications*, R. Zaier, Ed. Intech, 2012, ch. 1, pp. 1–28.

[18] A. Albu-Schaffer, O. Eiberger, M. Grebenstein, S. Haddadin, C. Ott, T. Wimbock, S. Wolf, and G. Hirzinger, "Soft robotics," *IEEE Robotics & Automation Magazine*, vol. 15, no. 3, pp. 20–30, Sept. 2008.

[19] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, no. May. IEEE, 2002, pp. 1398–1403.

[20] 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.