

Skill Memories for Parameterized Dynamic Action Primitives on the Pneumatically Driven Humanoid Robot Child Affetto

Jeffrey Frederic Queißer¹, Hisashi Ishihara², Barbara Hammer¹, Jochen Jakob Steil³ and Minoru Asada²

Abstract—In this work, we propose an extension of parameterized skills to achieve generalization of forward control signals for action primitives that result in an enhanced control quality of complex robotic systems. We argue to shift the complexity of learning the full dynamics of the robot to a lower dimensional task related learning problem. Due to generalization over task variability, online learning for complex robots as well as complex scenarios becomes feasible. We perform an experimental evaluation of the generalization capabilities of the proposed online learning system through simulation of a compliant 2DOF arm. Scalability to a complex robotic system is demonstrated on the pneumatically driven humanoid robot Affetto including 6DOF.

I. INTRODUCTION

Modern robot applications often require skill learning that covers task variability. For this aim, Ijspeert et al. [1] proposed models for action generation based on dynamic motion primitives and perceptual coupling which display inherent generalization and robustness to disturbances. Further work extends this idea and introduces skill memories to perform a generalization of DMPs and other action primitives based on a high level task description [2]–[9].

In recent years, interactive robots incorporating robust pneumatic actuators have received more attention for real-world applications. In addition to their inherent compliance, a lower susceptibility to overheat and an easy combination with lightweight backdrivable transmission systems, such like proposed by Whitney et al. [10], is possible. This is important, because the risk analysis of head injuries on collision with robotic actuators by Zinn et al. [11] shows that one way to lower the risk of injury is the reduction of the inertia of the moving parts of the robot. A further option to enhance safety is a decrease of the stiffness of the actuator. Unfortunately, the control of pneumatically actuated robots is impeded by delays, friction and complex dynamics. The application of pneumatic robots in interactive scenarios is confronted with additional challenges, like variable configurations of the robot or unmodeled interaction forces. To deal with the aforementioned challenges, the complete dynamics of the robot and the interaction is required for classic model based control approaches. In addition to a parameterization

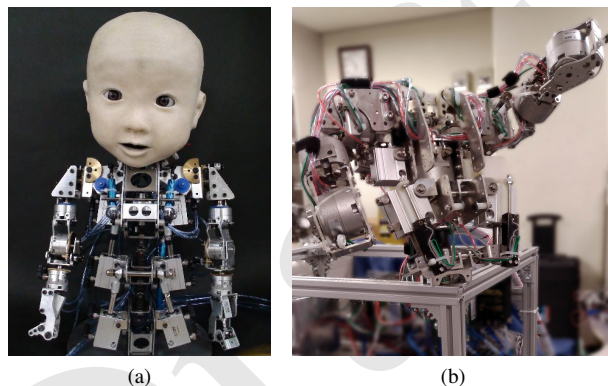


Fig. 1. Affetto robot, (a) upper body and internal structure as presented in [12], [13]. (b) Experimental setup used for online learning.

by external factors, the dynamics may evolve over time due to e.g. changing material properties caused by wear-and-tear or task demands. Modeling these properties is difficult or sometimes not possible at all and does not permit a reliable control of the actuators.

In this work, we propose to extend the concept of skill memories to generate feed-forward signals that represent complex dynamics properties of the robot and reduce the tracking error of the low-level controller. In comparison to classic approaches that estimate the complete inverse dynamics model of the robot [14], [15] or hybrid approaches [16]–[18] that incorporate learning, we focus on primitive based representations. We combine kinematic representations with the the concept of feed-forward signal generation of the servo theory of the motor cortex [19]–[21]. For a given parameterization of our task, the Parameterized Skill (PS) is supposed to estimate a solution in terms of joint angle trajectories that fulfill the task (as demonstrated in previous works) and an associated feed-forward signal that minimizes the tracking error of the joint controller. This allows to shift the complexity from learning complex robot dynamics to task related primitives. In comparison of our work with the torque primitives for impedance control, proposed in [22], a continuous generalization of forward signals based on a high-level task parameterization is performed in this work.

Our experimental platform is the Affetto robot [12], which is a pneumatically actuated humanoid with a large number of antagonistically actuated joints. The robot Affetto does not support direct torque control and does not provide dynamics models for reliable joint control. Thus, we face a high task complexity as well as delays and dynamic effects caused by the pneumatic actuation. Note that the proposed method to encode task-related feed-forward signals is not

¹Jeffrey Frederic Queißer & Barbara Hammer are with the Research Institute for Cognition and Robotics (CoR-Lab), Bielefeld University, Universitätsstr. 25, 33615 Bielefeld, Germany [jqueisse|bhammer]@cor-lab.uni-bielefeld.de

²Hisashi Ishihara & Minoru Asada are with the Graduate School of Engineering, Osaka University, Suita, Osaka 565-0871, Japan [ishihara|asada]@ams.eng.osaka-u.ac.jp

³Jochen Jakob Steil is with the Institute for Robotics and Process Control, Technische Universität Braunschweig, Mühlenpfordtstr. 23, 38106 Braunschweig, Germany jsteil@rob.cs.tu-bs.de

limited to pneumatically actuated robots. It is particularly interesting for all robots that are difficult to control by classical control schemes due to their complexity, like e.g. tendon driven actuators or soft robots.

The contribution of this work is an extension of online learning of a Parameterized Skill (PS) for trajectory representations as in [3], [5], [6], [8], [9] to incorporate the unmodeled dynamics of highly compliant pneumatic robot systems. We perform an experimental evaluation of our approach to enhance the control quality on a simulated compliant 2DOF planar arm and demonstrate the scalability to a complex real 6DOF robot system. As in our previous work for kinematic PS [9], we investigate a *bootstrapping* process that results in accelerating the optimization process as soon as enough training samples have been consolidated by the memory.

II. PARAMETERIZED SKILLS FOR DYNAMIC ACTION PRIMITIVES

Our previous work, as shown in Fig. 2, introduced parameterized skills as a mapping from task parameterizations to motion primitives. This allows for generalization of actions, i.e. joint angle trajectories encoded by DMPs, for new task configurations and goals [9]. Actions are optimized w.r.t. a reward function by black-box optimization and used for incremental training of the parameterized skill. For a given task such as reaching with a 10DOF arm, a parameterized skill is able to generalize to adequate actions for new parameterizations (i.e. via-point positions). If the parameterized skill generalizes, but is not successful, an optimizer is used to solve the task. Successfully optimized tasks are used as training data for the parameterized skill and subsequent optimizations benefit from an improved initialization. This results in a process we denote as *bootstrapping*: The more solutions have been found, the less rollouts are required for a new optimization. It was shown that this leads to a significant speed up of the exploration of the parameterized skill [9].

For the current work, we expect that the generalization of joint trajectories for task parameterizations is already available. Extending our previous work [9], we train parameterized skills to generalize for forward signals that represent the dynamics of the robot and its environment. Thus the parameterized skill generalizes for policy parameterizations that are encoded into forward signals to support the feedback controller in execution of the parameterized target trajectory. Our work also constitutes a first step towards the generation of complex dynamic motions, since action primitives can be mixed or sequenced. Training samples are gathered by iterative optimization of the initial guess of the parameterized skill. Our experiments evaluate the generalization capabilities of the parameterized skill for forward signals that reduce the tracking error of the feedback controller as well as the iterative optimization of forward signals and online learning.

Fig. 3 shows the structure of our proposed learning framework: Target trajectories in relation to the task parameterization (Fig. 3-①) are assumed to be given, as highlighted

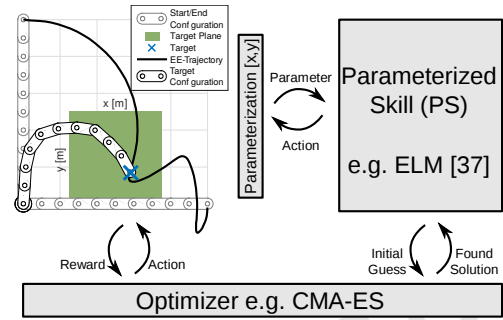


Fig. 2. Previous work, bootstrapping loop of parameterized skills as proposed in [9]. System overview including simulation of a 10DOF planar arm, the reaching target at time $\frac{T}{2}$ is variable and located on the target plane. The parameterized skill performs generalization from the reaching target to the high dimensional parameterization of the action primitive. Training samples for the parameterized skill are estimated by black-box optimization.

in red in Fig. 3-②). The estimation for feed-forward signals $u_{j=1}^{FFWD}(t)$ for the first iteration $j = 1$ is performed by the parameterized skill $PS(\tau)$ (Fig. 3-③) and its encoding (Fig. 3-④). Iterative optimization of the generalized feed-forward signal $u_{j+1}^{FFWD}(t)$ for one task instance (defined by τ) is given by Fig. 3-⑤). Optimization is performed until convergence of the tracking error has been achieved. The feed-forward signal $u^{FFWD*}(t)$ giving the lowest tracking error is used as training data for an incremental update of $PS(\tau)$. For action execution, a feedback controller (Fig. 3-⑥) estimates a control signal $u_j^{PID}(t)$ based on the current tracking error $e_j(t)$. The overall forward signal commanded to the robot system is given by $u(t) = u_j^{PID}(t) + u_j^{FFWD}(t)$. The parameterized skill does not estimate the complete inverse dynamics of the robot system and its environment, as performed in case of classic robot control applications for estimation of $u_j^{FFWD}(t)$. The generalization of optimized u_j^{FFWD} is based on the high level task parameterization and is supposed to support the feedback controller.

In the case of the Affetto robot, we are not able to directly command joint torques or accelerations. To abstract the antagonistic control signals that represent the opening of the valves of the pneumatic chambers, we refer to the PIDF controller [23] as shown in Fig. 3-⑦). This allows us to operate with $u(t)$ in the domain of desired pressure differences that correlate to torques at the end-effector (Fig. 3-⑧). The overall system incorporates three nested loops: 1) Generalization of forward signals and the respective joint angle trajectories for each new task instance; 2) Iterative optimization of generalized forward signals; 3) Execution of the joint trajectory by the low-level controller.

A crucial requirement for the estimation of optimized feed-forward signals is the repeatability of the generated movements of the robot. As investigated in [23] for a humanoid robot with comparable air valves and actuation principle, resulting end effector trajectories showed proper repeatability under multiple executions of identical controller signals. We are faced with a complex representation: The parameterization of the task will affect the desired trajectory as well as the optimal feed-forward signal, e.g. caused by different loads at the end-effector, variable stiffness of the actuator

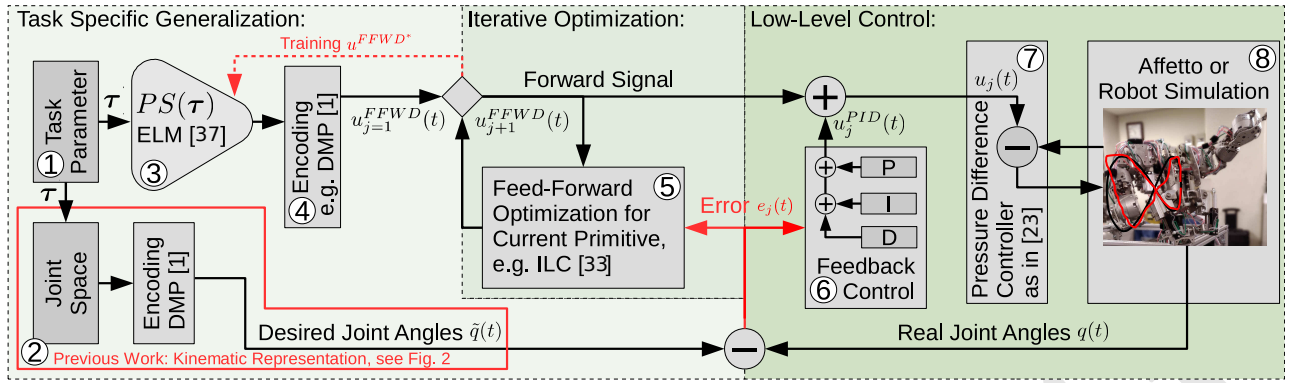


Fig. 3. System overview of the proposed action generation framework. The parameterized skill $PS(\tau)$ is the core component and mediates between high-level task parameter and feed-forward signals representing the dynamic properties of the system. Background color indicates functional grouping and the nested loop structure of task parameterization, feed-forward signal optimization and primitive execution.

or changing trajectory durations. Our evaluation metric is the generalization performance of the parameterized skill for feed-forward signals of unseen task parameterizations. We expect that the more training samples have been presented to the parameterized skill, the better is the generalized feed-forward signal. We therefore expect a gradually increasing tracking performance as well as a reduced number of required optimization steps to achieve convergence of minimizing the tracking error of the system. In the following, the chosen signal representation, the algorithm for feed-forward signal optimization, the selected learning method and the task variability are introduced.

A. Feed-Forward Signal Representation

The proposed method does not rely on a specific type of policy representation, i.e. compact representation and encoding of forward signals to support the execution of motion primitives. Many methods for compact temporal signal representation have been proposed, e.g. based on Gaussian Mixture Models (GMM) [24] or Neural Imprinted Vector Fields [25]. We decided for a dynamical system representation based on Dynamic Motion Primitives (DMP, [1]), because they are widely used in the field of motion generation and show good task related generalization capabilities. DMPs for point-to-point motions are based on a dynamical point attractor system. Feed-forward signal $u_{j=1}^{FFWD}(t)$ as well as its velocity and acceleration profiles, as in Fig. 3-④, are defined as:

$$\ddot{u}_{j=1}^{FFWD} = k_S(g - u) - k_D \dot{u}_{j=1}^{FFWD} + f(x, \theta) \quad (1)$$

The canonical system is typically defined as $\dot{x} = -\alpha x$ or in our case as a linear decay $\dot{x} = -\alpha x$ as in [26]. The shape of the primitive is defined by disturbance

$$f(x, \theta) = \frac{\sum_{k=1}^K \exp(-V_k(x - C_k)) \theta_k}{\sum_{k=1}^K \exp(-V_k(x - C_k))}, \quad (2)$$

with the number of Gaussians K set to 20 per DOF throughout this work. C_k are the Gaussian centers and V_k define the variance of the Gaussians. The DMP is parameterized by the coefficients θ_k , generalized by the parameterized skill. We assume fixed variances V_k and a fixed distribution of centers C_k as in [1], [27].

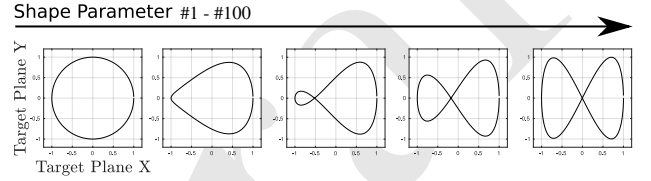


Fig. 4. Discretized shape variation that was used for evaluation.

B. Selection of Feed-Forward Signal Optimization Algorithm

For optimization of feed-forward signals encoded by policy parameters θ given a task parameterization τ , we apply Iterative Learning Control (ILC, [28]–[30]). Integration into our framework is shown in Fig. 3-⑤. ILC is a method for optimizing control signals and was initially proposed as a solely feed-forward approach. Application in combination with feedback control was demonstrated as well in [31], [32]. A successive observation and update of the feed-forward signal leads to a reduction of the tracking error and thereby to a smaller feedback controller response. ILC is widely used in industrial application areas, e.g. for enhancing positioning precision of machines [33], [34]. We utilize the PD-Type learning function for our experiments [32]: The feed-forward signal is updated based on a proportional (P) and derivative (D) gain of the current error. ILC is based on a Q-Filter and learning function L . A low-pass filter Q suppresses high frequency learning and contributes to the stability of ILC. In our case, the Q-filter is given by the representation of the feed-forward signal as DMP parameterization (inherent smoothing), additionally we use a Gaussian filter for the error signal. The function L for an update of the signal refers to

$$u_{j+1}^{FFWD}(t) = u_j^{FFWD}(t) + k_P e_j(t + d) + k_D [e_j(t + d + 1) - e_j(t + d)], \quad (3)$$

for iteration j , proportional factor k_P , derivative factor k_D and system delay d . The error $e_j(t)$ over time is defined by the difference between desired joint angle \tilde{q} and joint angles of the current iteration q_j : $e_j(t) = \tilde{q}(t) - q_j(t)$. Due to the high compliance in our application and the pneumatic actuation principle, we expect long and varying temporal delays between the control signal and a response of the actuator. Therefore we estimate the current temporal delay

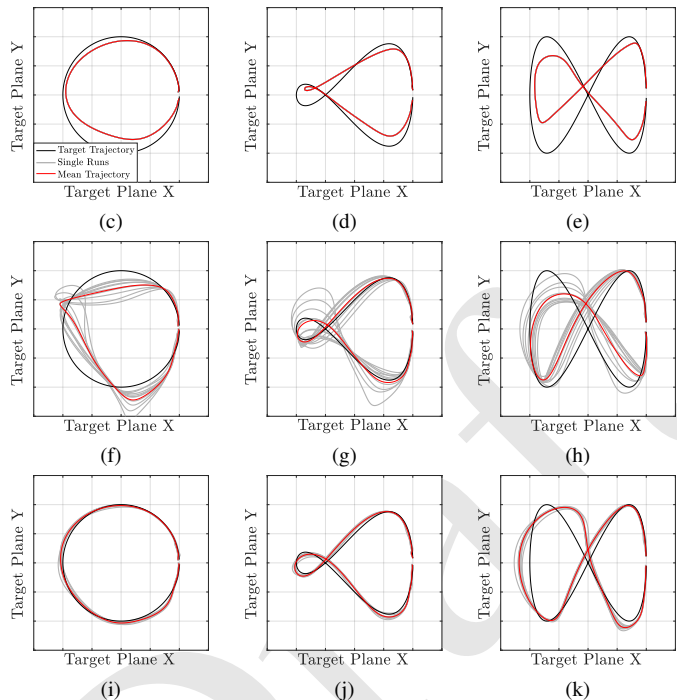
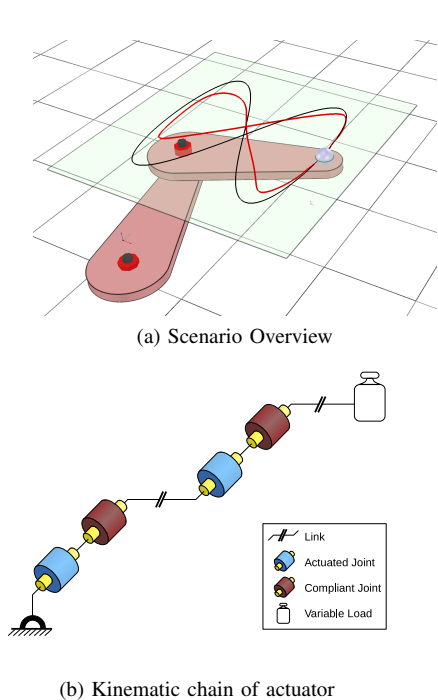


Fig. 5. (a) Experimental setup of the compliant 2DOF arm experiment. Due to the high compliance of the robot, tracking tasks on the 2D target plane (black line) result in disturbed trajectories (red line). (b) Kinematic chain of the simulated actuator. (c-k) Examples of the generalization of $\text{PS}(\tau)$ to unseen tasks. Results for generalized forward signals: for three shape parameterizations and a fixed load resulting target trajectories for a zero forward signal (c-e), for a parameterized skill trained with two samples (f-h) and for 10 presented training samples (i-k) is shown.

d of the system by estimation of the time shift with the minimum error between the target and the actuator response: $\text{argmin}_d \frac{1}{T} \sum_t \|\tilde{q}(t) - q_j(t+d)\|$.

C. Selection of Learning Algorithm

Fig. 3-③ shows the parameterized skill $\text{PS}(\tau)$. For learning of optimized feed-forward signals $u^{FFWD*} = \text{PS}(\tau)$, we apply an incremental variant of the Extreme Learning Machine (ELM, [35]). ELMs are feed-forward neural networks with a single hidden layer:

$$\text{PS}_i(\tau) = \sum_{j=1}^H \mathbf{W}_{ij}^{\text{out}} \sigma \left(\sum_{k=1}^M \mathbf{W}_{jk}^{\text{inp}} \tau_k + \mathbf{b}_j \right) \quad \forall i = 1, \dots, N \quad (4)$$

with input dimensionality M , hidden layer size H and output dimensionality N . Hidden layer size was set to $H = 50$ for the experiments conducted in this work. Regression is based on a random projection of the input $\mathbf{W}^{\text{inp}} \in \mathbb{R}^{H \times M}$, a non-linear transformation $\sigma(x) = (1 + e^{-x})^{-1}$ and a linear output transformation $\mathbf{W}^{\text{out}} \in \mathbb{R}^{N \times H}$. The incremental update scheme of the ELM was introduced as Online Sequential ELM (OSELM) [36] that allows for additional regularization on the weights [37] or exponential forgetting of previous samples [38]. Since we expect to deal with a small number of training data, regularization of the network can help to prevent over-fitting and foster reasonable extrapolation.

D. Selection of Parameterized Task

For our experiments we evaluate parameterized 2D end-effector tracking tasks as shown in Fig. 4. In addition we

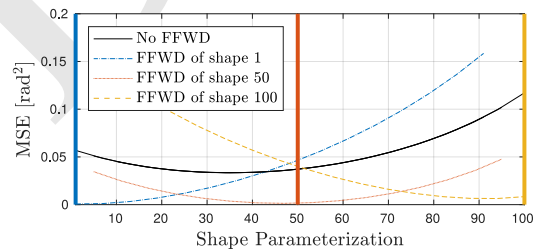


Fig. 6. Evaluation of generalization of forward signals with respect to the task parameterization. Tracking error of the 2DOF arm with zero forward signal (black) is compared to situations when the optimized forward signal (FFWD) for a specific shape parameterization is used (#1, #50 and #100).

vary end-effector loads in simulation as well as the overall duration for the real robot of the action primitives. As mentioned before we evaluate the learning of the feed-forward signals and assume the joint angle trajectories are given.

III. EXPERIMENTS

In the following, we demonstrate the feasibility of our proposed bootstrapping algorithm. Therefore we designed two scenarios to test the bootstrapping of parameterized skills according to the method presented in sec. II.

A. 2 DOF Planar Arm Task

The first experiment was performed in simulation. We modeled a compliant 2DOF planar arm in the simulation environment VREP [39]. To be able to simulate highly compliant joints, we utilize two simulated joints for each DOF of the robot. The resulting kinematic chain and the simulation setup is shown in Fig. 5(a-b). For simulation

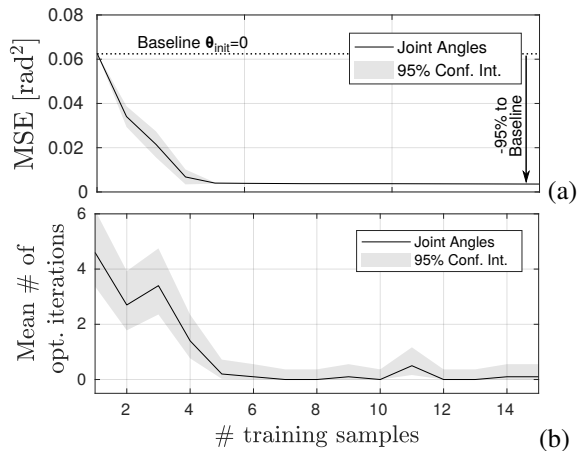


Fig. 7. Decreasing tracking error caused by the forward signal that is encoded as $\theta_{start} = \text{PS}(\tau)$ in relation to the number of presented training samples (a) and the mean number of rollouts that are necessary for optimization by ILC until convergence (b). Results and confidence interval are based on ten repeated experiments.

of the dynamics we select the *Newton Dynamics* engine with a temporal resolution of 20ms. Each joint is driven by a feedback controller that calculates the error between the target joint angle and the real joint angle given by an actuated and the compliant joint. Based on this error the PID controller results in a control signal for the actuated joint. In addition, we provide a forward signal so that the final control of the actuated joint is based on the sum of the PID controller and the forward signal. As presented in sec. II, we parameterize the task by the shape of the end-effector trajectory and estimate appropriate joint angle signals by the inverse kinematic solver of VREP. As a second dimension of the parameterization of the task, we vary the weight of a load attached to the end-effector of the robot.

The evaluation of the generalization properties of optimized forward signals for single instances is analysed in Fig. 6. We compare the tracking performance of the PID controller with zero forward signal (baseline) to three situations in which we utilize forward signals optimized by ILC for a specific shape parameterization (#1,#50 and #100, see Fig. 4). By manual tuning we estimated ILC update parameters $K = [k_P, k_D] = [0.005, 0.04]$ and a Gaussian window filter size of 100 timesteps. As we can see in Fig. 6, the tracking error is much lower for the shape parameterizations if we optimize the forward signal for this specific shape (colored vertical bars). The more the shape deviates from the shape for which we optimized the forward signal the higher the tracking error, since we use a feed-forward signal that was not optimized for the current shape. If the forward signal was optimized for a shape that strongly deviates from the evaluated shape, the tracking error of the controller that utilizes the forward signal can be higher compared to no forward signal. In this case, the forward signal disturbs the trajectory tracking and is not beneficial for the feedback controller. This experiments shows that we can benefit in a local neighborhood of our task parameterization of an optimized feed-forward signal.

Based on the previous observations, we perform the

evaluation of the generalization capabilities of the parameterized skill in the second experiment. We generate a fixed set of test parameterizations over shape and load (0-2kg) to evaluate the system performance during the presentation of random tasks used for training. For each new training task instance, we query the parameterized skill for a generalization of feed-forward signals. Given this initial feed-forward signal we perform ILC iteratively for optimization. Iterations are performed until convergence criterion of the joint tracking error is fulfilled. The optimized solution for the forward signal for the given task is used as training sample and iterative update of the parameterized skill.

We evaluate the current generalization performance by estimation of the tracking error for the test set. The results of this procedure can be seen in Fig. 7, with an increasing number of presented training tasks and updates of the parameterized skill the MSE of the trajectory tracking task decreases. Additionally, we observe that the number of iterations necessary to achieve convergence of the ILC for new training tasks decreases as more solutions for tasks have been consolidated by the parameterized skill. This allows for a bootstrapping of the learning process: The more experience the system has in solving task instances the faster it can find solutions for unseen instances. Fig. 5(c-k) shows the tracking performance of the end-effector for three shape parameterizations as more samples have been presented to the parameterized skill. It can be seen that the system is able to execute the desired task with a higher precision after presentation of training samples. After the presentation of only two samples we can see a higher variance in the generated samples which is caused by the high shape variance in the randomly selected tasks.

B. Upper Body Control of the Affetto Robot

The second part of the experiments targets the Affetto robot platform, as shown in Fig. 1(a). The Affetto is a humanoid robot child driven by pneumatic actuators, as introduced in [12], [13]. For our experiments we utilize 6DOF of 8DOF of one side of the upper body of the Affetto robot. Experiments are performed on the real robot platform (shown in Fig. 1(b)) and we refer to the kinematics simulation (shown in Fig. 8(a)) only for visualization and generation of joint angle trajectories. We generate joint trajectories in relation to a task parameterization that defines the shape of the target end-effector trajectory of the right arm. The remaining 2DOF are assumed to be optional joints and neglected in the further evaluation. As before we execute end-effector trajectories as described in sec. II, but we vary the duration of the actions (1.6-26.6 seconds) as second parameter.

As for the 2DOF experiment we utilize a kinematics model and the inverse kinematics solver of the VREP simulator. We ensure that the generated joint angle trajectories do not contain multiple solutions of the redundancy resolution and can be seen as parameterized functions. The simulation of the kinematics is shown in Fig. 8(a). We use the PIDF controller [23] for the pneumatically driven joints of the robot and optimize the controller parameters by automatic

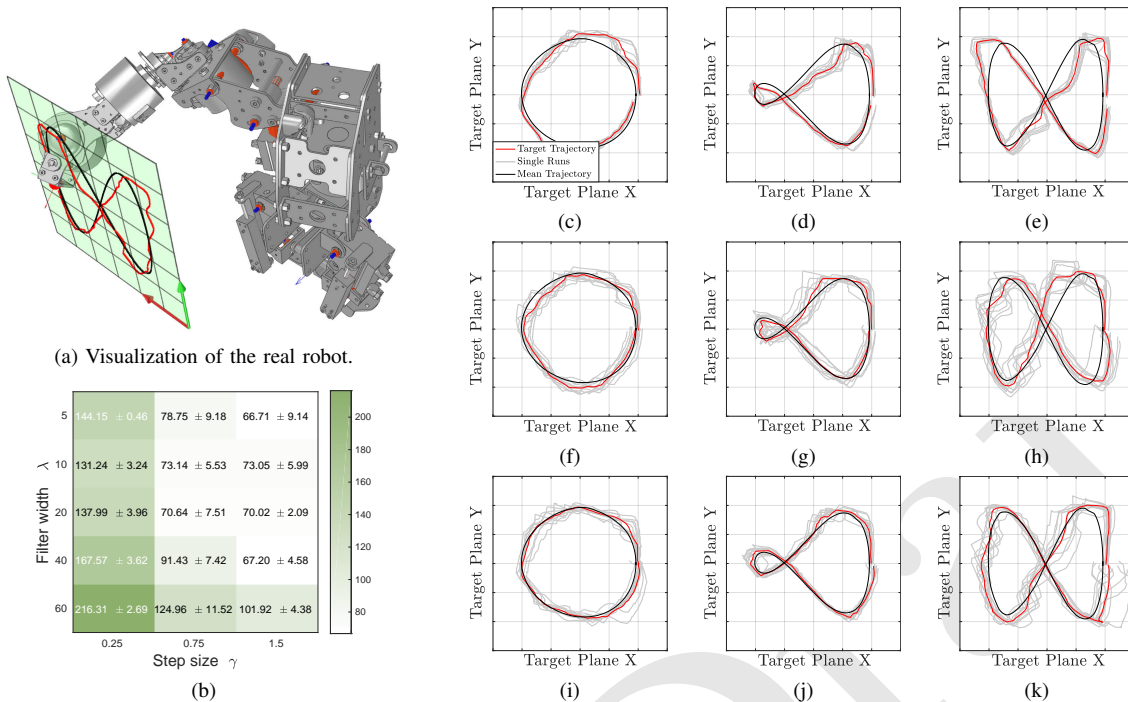


Fig. 8. (a) Experimental setup of the Affetto experiment. Kinematics simulation is used for generation of target joint angle trajectories and visualisation only, experiments are performed on the real robot platform. Due to the high compliance of the robot, tracking tasks on the 2D target plane (black line) result in disturbed trajectories (red line). (b) Results of parameter grid search of ILC filter width and step size. (c-k) Examples of the generalization of $PS(\tau)$ to unseen tasks. Results for generalized forward signals: for three shape parameterizations and a fixed load resulting target trajectories for a zero forward signal (c-e), for a parameterized skill trained with two samples (f-h) and for 20 presented training samples (i-k) is shown.

optimization and hand tuning on a test trajectory that includes sine waves and steps.

We perform a grid search to estimate appropriate parameters for the iterative PD update step of ILC and the filter width, as introduced in sec. II. The result of the grid search are shown in Fig. 8(b), where we evaluated the achieved tracking performance for shape parameterization #50. Based on this evaluation, we decided for a Gaussian window filter with a width of 20 time steps and update rate factor $0.75K$, for a compromise between a low update gain and a suppression of high frequency signals. As shown in 8(b), smaller filter widths or larger step sizes do not result in significantly lower tracking errors but enhance the risk for instabilities during ILC optimization. We perform the same evaluation as in the 2DOF experiment of sec. III-A. As Fig. 9 shows, we were able to achieve similar results as in our previous simulation. The parameterized skill is able to incrementally improve the generalizations for new task parameterizations. The more samples have been used for training, the lower the tracking error for unseen tasks. In addition, we can see the same bootstrapping effect as in the previous experiment: we observe a significant reduction of the required ILC iterations with the gradually improved parameterized skill. The results indicate good scaling properties of the proposed system, as only 30 training samples are required for an application on 6DOF and a real robotic system. The kinematics model is used to visualize the tracking performance of the end-effector for three shape parameterizations as more samples have been presented to the parameterized skill, as shown in Fig. 8(c-k).

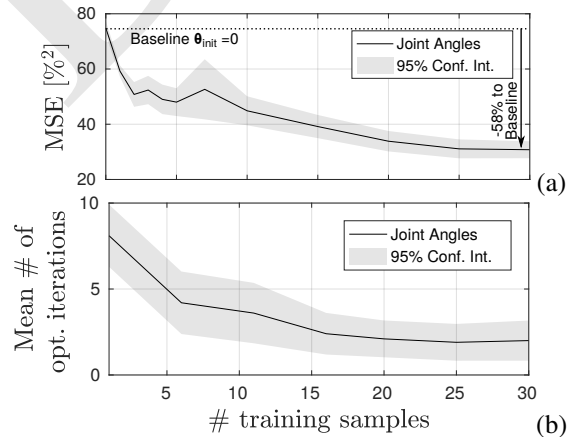


Fig. 9. Decreasing tracking error caused by the forward signal that is encoded as $\theta_{\text{start}} = PS(\tau)$ in relation to the number of presented training samples (a) and the mean number of rollouts, necessary for optimization by ILC until convergence (b). Confidence interval is based on ten repetitions. MSE is based on deviation in relation to the actuator range.

IV. DISCUSSION & CONCLUSION

In this work, we introduce parameterized skills for generalization of feed-forward signals that support feedback controller in the control of highly compliant robots. Incremental learning can significantly reduce the tracking error of the humanoid robot Affetto as well as the number of required optimization iterations for unseen task instances. One of the most fundamental arguments throughout this work is that learning of dynamics properties is not bound to the complexity of the robot and its environment since we perform an action/task related generalization. We demonstrate

the working principle on a chain of six highly compliant pneumatically actuators without to refer to complex (model based) control strategies that deal e.g. with friction nor time delays. Even under this difficult conditions the system was able to optimize for a complex task with a low number of rollouts. The low number of required training samples for the presented 2D task parameterization motivates further work scaled to higher dimensional tasks as well as the integration into a more complex experimental setup that combines the learning of trajectory representations and forward signals. Additionally, the extension of the system by a representation of the stiffness of the actuator would allow an enhanced interaction for real world tasks.

ACKNOWLEDGMENT

J. Queißer received funding from the Cluster of Excellence 277 Cognitive Interaction Technology and has been supported by the CODEFROR project (FP7-PIRSES-2013-612555) - <https://www.codefror.eu/>. In addition work was partly supported by PRESTO, JST Grant Number JPMJPR1652.

REFERENCES

- [1] 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, no. 2, pp. 328–373, 2013.
- [2] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Machine Learning*, vol. 84, no. 1, pp. 171–203, 2010.
- [3] B. D. Silva, G. Konidaris, A. G. Barto, and B. Castro, "Learning Parameterized Skills," in *Intern. Conf. on Machine Learning*, 2012, pp. 1679–1686.
- [4] 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.
- [5] F. Reinhart and J. J. Steil, "Efficient Policy Search with a Parameterized Skill Memory," in *IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems*. IEEE, 2014, pp. 1400–1407.
- [6] A. Baranes and P. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 49–73, 2013.
- [7] K. Mülling, J. Kober, O. Kroemer, and J. Peters, "Learning to select and generalize striking movements in robot table tennis," *Intern. Journal of Robotics Research*, vol. 32, no. 3, pp. 263–279, 2013.
- [8] B. C. da Silva, G. Baldassarre, G. Konidaris, and A. Barto, "Learning parameterized motor skills on a humanoid robot," in *IEEE Intern. Conf. Robotics and Automation*, 2014, pp. 5239–5244.
- [9] J. F. Queißer, R. F. Reinhart, and J. J. Steil, "Incremental Bootstrapping of Parameterized Motor Skills," in *Proceedings IEEE Humanoids*. IEEE, 2016, pp. 223–229.
- [10] J. P. Whitney, M. F. Glisson, E. L. Brockmeyer, and J. K. Hodgins, "A low-friction passive fluid transmission and fluid-tendon soft actuator," in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2014, pp. 2801–2808.
- [11] M. Zinn, O. Khatib, B. Roth, and J. K. Salisbury, "A new actuation approach for human friendly robot design," *Intern. Journal of Robotics Research*, vol. 23, no. 4–5, pp. 379–398, 2004.
- [12] H. Ishihara, Y. Yoshikawa, and M. Asada, "Realistic child robot *Affetto* for understanding the caregiver-child attachment relationship that guides the child development," in *2011 IEEE Intern. Conf. on Development and Learning (ICDL)*, vol. 2, 2011, pp. 1–5.
- [13] H. Ishihara and M. Asada, "Design of 22-dof pneumatically actuated upper body for child android *affetto*," *Advanced Robotics*, vol. 29, no. 18, pp. 1151–1163, 2015.
- [14] M. Kawato, Y. Uno, M. Isobe, and R. Suzuki, "Hierarchical neural network model for voluntary movement with application to robotics," *IEEE Control Systems Magazine*, vol. 8, no. 2, pp. 8–15, 1988.
- [15] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive Processing*, vol. 12, no. 4, pp. 319–340, 2011.
- [16] D. Nguyen-Tuong and J. Peters, "Using model knowledge for learning inverse dynamics," in *Intern. Conf. on Robotics and Automation*. IEEE, 2010, pp. 2677–2682.
- [17] D. Romeres, M. Zorzi, R. Camoriano, and A. Chiuso, "Online semi-parametric learning for inverse dynamics modeling," in *55th Conf. on Decision and Control*, Las Vegas, US, 2016, pp. 2945–2950.
- [18] R. F. Reinhart, Z. Shareef, and J. J. Steil, "Hybrid analytical and data-driven modeling for feed-forward robot control," *Sensors*, vol. 17, no. 2, p. 311, 2017.
- [19] N. Schweighofer, M. A. Arbib, and M. Kawato, "Role of the cerebellum in reaching movements in humans. i. distributed inverse dynamics control," *The European journal of neuroscience*, vol. 10 1, pp. 86–94, 1998.
- [20] M. Kawato, K. Furukawa, and R. Suzuki, "A hierarchical neural-network model for control and learning of voluntary movement," *Biological Cybernetics*, vol. 57, no. 3, pp. 169–185, 1987.
- [21] M. S. A. Graziano, *Shared Representations: Sensorimotor Foundations of Social Life*. UK: Cambridge University Press, 2015, ch. A new view of the motor cortex.
- [22] T. Petrič, L. Colasanto, A. Gams, A. Ude, and A. J. Ijspeert, "Bio-inspired learning and database expansion of compliant movement primitives," in *15th Intern. Conf. on Humanoid Robots*, 2015, pp. 346–351.
- [23] E. Todorov, C. Hu, A. Simpkins, and J. Movellan, "Identification and control of a pneumatic robot," in *3rd IEEE Intern. Conf. on Biomedical Robotics and Biomechanics*, 2010, pp. 373–380.
- [24] F. Guenter, M. Hersch, S. Calinon, and A. Billard, "Reinforcement learning for imitating constrained reaching movements," *Advanced Robotics, Special Issue on Imitative Robots*, vol. 21, no. 13, pp. 1521–1544, 2007.
- [25] A. Lemme, K. Neumann, R. Reinhart, and J. J. Steil, "Neural learning of vector fields for encoding stable dynamical systems," *Neurocomp.*, vol. 141, pp. 3–14, 2014.
- [26] T. Kulvicius, K. Ning, M. Tamosiunaite, and F. Wörgötter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting," *IEEE Trans. Robotics*, vol. 28, no. 1, pp. 145–157, 2012.
- [27] R. F. Reinhart and J. J. Steil, "Efficient policy search in low-dimensional embedding spaces by generalizing motion primitives with a parameterized skill memory," *Autonomous Robots*, vol. 38, no. 4, pp. 331–348, 2015.
- [28] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of Robots by learning," *Journal of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.
- [29] R. W. Longman, *Designing Iterative Learning and Repetitive Controllers*. Boston, MA: Springer US, 1998, pp. 107–146.
- [30] M. Norrlöf and S. Gunnarsson, "Experimental Comparison of some Classical Iterative Learning Control Algorithms," *IEEE Trans. on robotics and automation*, vol. 18, no. 4, pp. 636–641, 2002.
- [31] D. D. Roover and O. H. Bosgra, "Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system," *Intern. Journal of Control*, vol. 73, no. 10, pp. 968–979, 2000.
- [32] D. Bristow, M. Tharayil, and A. Alleyne, "Survey of iterative learning control: A learning-based method for high-performance tracking control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, 2006.
- [33] C.-K. Chen and J. Hwang, "Iterative learning control for position tracking of a pneumatic actuated X-Y table," *Control Engineering Practice*, vol. 13, no. 12, pp. 1455–1461, 2005.
- [34] D.-I. Kim and S. Kim, "An iterative learning control method with application for cnc machine tools," *Trans. on Industry Applications*, vol. 32, no. 1, pp. 66–72, 1996.
- [35] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomp.*, vol. 70, no. 1-3, pp. 489–501, 2006.
- [36] N.-Y. Liang, G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Trans. on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, 2006.
- [37] H. T. Huynh and Y. Won, "Online training for single hidden-layer feedforward neural networks using RLS-ELM," in *Intern. Symp. on Comp. Intelligence in Robotics and Automation*, 2009, pp. 469–473.
- [38] J. Zhao, Z. Wang, and D. S. Park, "Online sequential extreme learning machine with forgetting mechanism," *Neurocomp.*, vol. 87, pp. 79–89, 2012.
- [39] M. F. E. Rohmer, S. P. N. Singh, "V-rep: a versatile and scalable robot simulation framework," in *IEEE Intern. Conf. on Intelligent Robots and Systems*, 2013, pp. 1321–1326.