Künstliche Intelligenz manuscript No.
(will be inserted by the editor)

Towards Learning of Generic Skills for Robotic Manipulation

Jan Hendrik Metzen · Alexander Fabisch · Lisa Senger · José de Gea Fernández · Elsa Andrea Kirchner

Received: date / Accepted: date

Abstract Learning versatile, reusable skills is one of the key prerequisites for autonomous robots. Imitation and reinforcement learning are among the most prominent approaches for learning basic robotic skills. However, the learned skills are often very specific and cannot be reused in different but related tasks. In the project BesMan, we develop hierarchical and transfer learning methods which allow a robot to learn a repertoire of versatile skills that can be reused in different situations. The development of new methods is closely integrated with the analysis of complex human behavior.

Keywords Multi-Task Learning · Skill Learning · Movement Primitives · Transfer Learning · Reinforcement Learning

1 Introduction

BesMan ("Behaviors for Mobile Manipulation")[1] is a joint project of the Robotics Research Group of the University of Bremen (UoB) and the Robotics Innovation Center of the German Research Center for Artificial Intelligence (DFKI). The project started in May 2012 and has a duration of four years. The goal of the project is the development of generic manipulation strategies which do not depend on a specific robot morphology and are suited for both one and two-arm systems such as AILA (see Fig. 1). Novel, situation-specific behavior shall be learned by means of a learning platform. The development of manipulation procedures is mainly the responsibility of the DFKI while the UoB develops the learning platform. In this paper, we focus on the learning platform.

The main idea of the learning platform is to learn skills based on human demonstrations of complex, task-specific behavior like grasping and manipulating an object. This complex behavior is split into simpler behavioral blocks (such as object grasping) using behavior segmentation methods. The robot learns movement primitives which correspond to these behavioral building blocks based on imitation learning [2,23] and reinforcement learning (RL) [12]. These movement primitives are specific for the demonstrated tasks: for instance, a movement primitive might encode how a specific object in a specific orientation can be grasped. However, the learning platform will contain means for

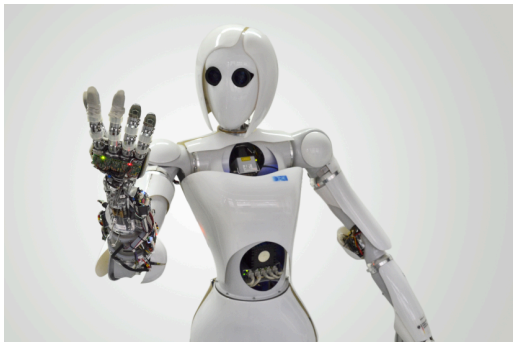

Fig. 1 Humanoid robot AILA, see `http://robotik.dfki-bremen.de/en/research/robot-systems/aila.html`.

This work was supported through two grants of the German Federal Ministry of Economics and Technology (BMWi, FKZ 50 RA 1216 and FKZ 50 RA 1217).

Jan Hendrik Metzen†, Alexander Fabisch†, Lisa Senger†, José de Gea Fernández⋆, Elsa Andrea Kirchner†,⋆
† Robotics Group, Universität Bremen, Bremen, Germany
⋆ Robotics Innovation Center, German Research Center for Artificial Intelligence (DFKI), Bremen, Germany
E-mail: {jhm,afabisch,senger}@informatik.uni-bremen.de, {jose.de_gea_fernandez,elsa.kirchner}@dfki.de

[1] See `http://robotik.dfki-bremen.de/en/research/projects/besman.html` for more details on the project.

transfer learning [26], which allow to learn more generic templates. Such a generic template may encode, e.g., how to grasp any object from a larger class of objects from a multitude of orientations. These templates can be instantiated to a movement primitive for a novel, previously unseen task (e.g., an object with a certain orientation). Furthermore, hierarchical RL [3] will allow to sequence movement primitives into more complex behaviors like grasping an object and placing it at a specific position (see, e.g., Stulp and Schaal [25]). This can be considered to be the contrary of behavior segmentation. The development of the learning platform will be accompanied by behavioral studies with human subjects that will give further insights into the learning, adaptation, and combination of movement primitives as well as their transfer between situations and tasks.

We present a summary of learning approaches for robotic movement primitives, present the learning platform used in BesMan, and propose learning skill templates based on transfer learning. Furthermore, we give some preliminary results.

2 Learning of Movement Primitives

In this section, we give an overview of methods for learning movement primitives, more specifically how movement primitives for simple tasks can be learned in the context of robotics. An overview paper summarizing recent work in this area was published by Peters et al. [19]. The most popular approach for learning movement primitives is RL. However, due to the large number of degrees of freedom and the continuous state and action spaces, classical, off-the-shelf RL algorithms are not well suited for the learning of movement primitives.

During the last years, several new RL algorithms from the field of direct policy search have been proposed that are specifically tailored to the learning of movement primitives: Peters et al. [21] propose an extension of 'vanilla' policy gradient methods by using the natural gradient. The method is shown to converge faster than classical policy gradient methods. Reward-weighted regression (RWR) [20] and "Policy learning by weighting exploration with the returns" (PoWER) [15] are based on the principle of reward-weighted self-imitation and allow to learn complex tasks such as the ball-in-a-cup benchmark. Relative Entropy Policy Search (REPS) [18] is a policy search method whose objective is to bound the loss of information during exploration. Hierarchical REPS [4] is a recent extension of REPS which allows in situations, where several locally optimal behaviors exist, to learn all these optima simultaneously. PI^2 [27] is a direct policy search method based on stochastic optimal control. PI^2 requires to specify an initial policy and a covariance matrix which governs exploration in weight space (often a multiple of the identity matrix). CMA-ES [8] is a metaheuristic for black-box optimization which is well suited for direct policy search [9].

All these methods are policy search approaches that optimize the weights of a fixed policy representation. The most popular class of policy representations that have been used to learn movement primitives for robotic manipulation are Dynamical systems Movement Primitives (DMPs) [10,13,16,17]. DMPs encode arbitrarily shapeable, goal-directed trajectories. The different variants of DMPs have in common that the encoded movement is governed by two superimposed attractor forces: (1) a fixed attraction to a goal position and (2) a modifiable and time-varying attractor force with decaying influence over time. Because the second attractor force loses its influence at the end of the movement, it is guaranteed that the goal is eventually reached. The second attractor force can be modified by adjusting a weight vector by means of imitation learning and RL.

Two main advantages of DMPs in comparison to other policy representations for our purposes are: (1) The DMP's goal position is always reached eventually, and thus, RL algorithms explore only movements that reach the goal and the exploration is more focused on promising policies. (2) Imitation learning is simplified for DMPs compared to other approaches, e.g., SEDS [11], since the optimal weights for a given demonstration can be determined by a closed-form formula.

3 Learning Platform

This section describes the architecture of the learning platform (see Fig. 2) which is used within the project BesMan for learning so-called skill templates. In general, learning complex behaviors at once is very time consuming or even impossible for artificial as well as biological agents. It is known from behavioral studies in rodents that learning of complex skill behavior takes place incrementally, i.e., smaller individual behavioral blocks are learned separately and later on combined during consolidation of the complex behavior by chunking them into sequences [7]. Considering this principal from skill learning in rodents, we have shown in a computational study that decomposing complex behavior into smaller behavioral blocks helps to simplify the learning problem and allows to learn more complex behavior than by a monolithic learning approach [1].

In BesMan, learning of new skill templates for a certain task is based on repeated demonstrations of successful behaviors for variations of this task by a human demonstrator. These demonstrations are recorded, preprocessed and synchronized. The preprocessed demon-

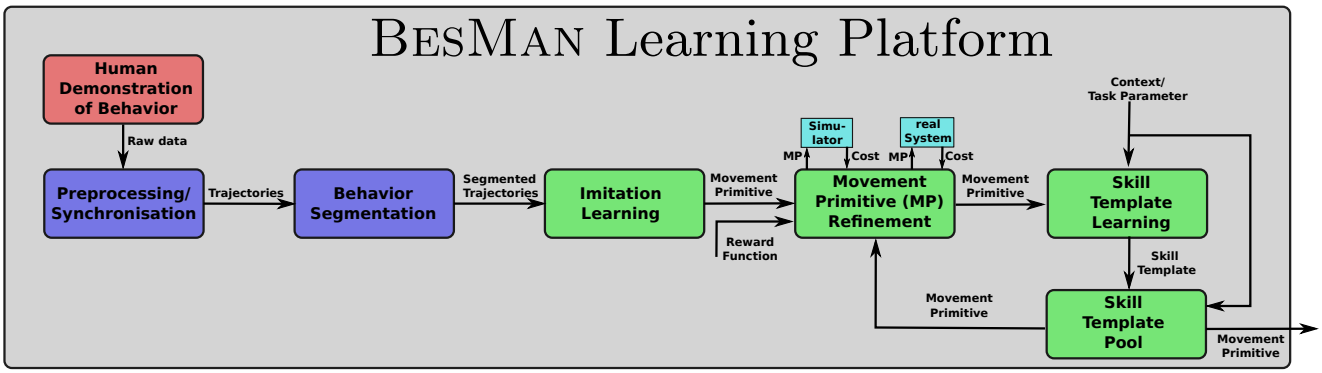

Fig. 2 Dataflow-diagram of the BesMan Learning Platform. Target behavior is demonstrated by a human and segmented into simpler behavioral building blocks. Movement primitives corresponding to these building blocks are learned using imitation learning and refined using reinforcement learning. The resulting movement primitives are specific for a certain task context but can be generalized to more generic skill templates. Once a new task is encountered, the skill template is instantiated and yields a task-specific movement primitive. This movement primitive can either be applied directly or can be refined further.

strations are then automatically decomposed into behavioral blocks by the module "Behavior Segmentation" of the learning platform. This module uses unsupervised behavior segmentation algorithms to automatically identify sequences in the recorded trajectories which correspond to behavioral building blocks. The segmentation algorithms used in BesMan are based on Bayesian inference. Each behavior segment is represented by a linear regression model, whose parameters are inferred by a segmentation algorithm. It is assumed that a new behavior segment begins when the underlying model changes. The multiple changepoint inference of Fearnhead and Liu [5] and the beta process autoregressive hidden Markov model (BPARHMM) of Fox et al. [6] are two methods to infer segments based on this assumption. In these probabilistic segmentation methods, noise in the data as well as variations in the execution of a behavior can be handled. Additionally, the BPARHMM labels the segments and assigns the same label to segments represented by the same regression model. In this way it gives the possibility to identify segments which belong to the same behavioral building block and thus represent multiple demonstrations for the same movement primitive.

For each of the identified segments, imitation learning methods are used to map the recorded trajectory segments onto movement primitives. These movement primitives describe trajectories in task space that mimic the trajectories presented by the human demonstrator while executing one behavioral building block. Because of potentially considerable differences in morphology and dynamic properties between human demonstrators and the robotic target systems, the mimicked movement primitives might not be optimal or even feasible for the respective target system and the optimal movement primitive will typically differ between target systems. To account for this, the "Movement Primitive Refinement" module adapts the learned movement primitives for the target system using RL. This requires interaction with the target system or with a simulation of it and the specification of a reward function which assesses the quality (the "cost") of the movement primitive for the setting.

The learned movement primitives are solutions for very specific tasks. In general, it is desirable to learn more generic capabilities that can be applied in an entire class of different but related tasks. This is achieved by the "Skill Template Learning" module which allows learning skill templates that generalize the learned movement primitives to new but similar settings (see Section 4). For this, demonstrations of behavior for different settings are required as well as examples of task settings and corresponding optimal movement primitives to generalize beyond these examples. A second capability of the module is to learn templates for complex behaviors that consists of several building blocks (and thus of a sequence of movement primitives). This requires to learn and handle dependencies between subsequent behavioral building blocks. Once a skill template has been learned and adapted to the specific target system, the skill template is added to the "Skill Template Pool". The skill templates in this pool can be used either directly during online operation or can be further refined for a specific task.

4 Skill Template Learning

Transfer Learning: A crucial factor for the performance of policy search methods is that the search is properly initialized, i.e., starts from a policy which is not too different from a successful solution of the task. As discussed, one way for obtaining such a policy is imitation

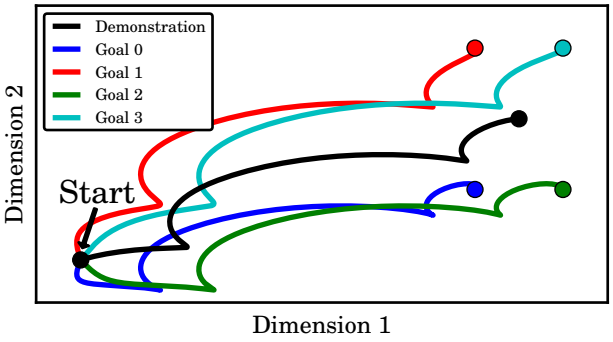

Fig. 3 DMPs are able to generalize over some metaparameters, e.g., the goal of a movement. The 2D trajectory of the original DMP is indicated by the black line. Changing the goal position to the colored dots results in the trajectories represented by the correspondingly colored lines. Note that the trajectories' shapes remain similar despite different goals.

learning. However, if movement primitives for a large number of different tasks shall be learned, this approach is cumbersome since every movement primitives needs to be demonstrated separately.

A more promising approach is thus to transfer knowledge learned in one task to a different but related task. If movement primitives are represented by DMPs, they contain some basic adaptivity inherently. That is, since they are controlled by metaparameters like the movement duration and the goal of the movement, they allow to generalize to some extent over these parameters. For instance, a DMP can be adapted directly to a slightly different goal position (see Fig. 3). Furthermore, it is also possible to modify the DMP formulation to incorporate more metaparameters, for example the velocity at the end of the movement [16]. However, for more complex problem classes, the dimensions of variations may not be easily encoded in the structure of the DMP. For instance, when learning throwing movements, it is not straightforward to encode the target of the throw as a metaparameter in the DMP formulation.

Another approach is to learn an explicit mapping which defines how movement primitives can be transferred from one specific task to a different but related task. To this end, Kober et al. propose Cost-regularized Kernel Regression (CrKR) [14], which is an extension of the RWR method. In contrast to RWR, the algorithm can change the policy based on task parameters supplied and thus generalize over a larger variety of task configurations. CrKR is shown to outperform RWR on a robotic dart-throwing task with varying goal positions (the task parameters). da Silva et al. [24] have proposed a method for learning parametrized skill which is built on top of the PoWER method: first, PoWER learns the optimal solutions for several different task configurations. Thereupon, supervised and manifold learning methods are used to generalize these example solutions to the entire task space. In this work, we propose an extension of this approach for learning so-called *skill templates*, which are similar to parametrized skills but in addition allow the agent to initialize the exploration strategy properly as we explain below.

Approach: Skill templates are learned in the context of parameterized reinforcement learning problems. We assume for a parameterized reinforcement learning problem class T that a task, i.e., an instance of the problem class, is defined by a task parameter vector $\tau \in T \subseteq \mathbb{R}^n$ and an associated interpretation. Likewise, a movement primitive is considered as a parameterized policy with parameter vector $\theta \in \mathbb{R}^m$. Using this notation, a *parameterized skill* [24] is a mapping Θ from task vector τ to a movement primitive vector θ_τ, i.e., $\Theta : \tau \mapsto \theta_\tau$. Let $J(\theta, \tau)$ be the expected return of the movement primitive parametrized by θ in task τ; the goal of learning parameterized skills is a mapping Θ^* such that $\Theta^* = \arg\max_\Theta \int P(\tau) J(\Theta(\tau), \tau) d\tau$, where $P(\tau)$ is the task distribution. As the parametrized skill Θ will typically not predict the optimal $\theta^*_\tau = \arg\max_\theta J(\theta, \tau)$, it is desirable to not only learn a point-estimate of θ^*_τ but also to give a measure of uncertainty of this prediction. We propose to learn a so-called skill template $\Psi = (\Theta, \Omega)$, which contains a function $\Omega : \tau \mapsto \Sigma_\tau$ with $\Sigma_\tau \in \mathbb{R}^{m \times m}$ that provides this uncertainty. Σ_τ can be interpreted as the covariance of a Gaussian distribution over the movement primitive's parameter space. Thus, a skill template Ψ can be seen as a mapping from a task to a Gaussian distribution over the movement primitive's parameter space, with Θ predicting the distribution's mean and Ω predicting the distribution's covariance.

Skill templates are learned based on a set of movement primitive weights that have been learned for specific task instances. Let $E = \{(\tau_i, \theta_{\tau_i}) | i = 1, \ldots, K\}$ be a training set consisting of experience collected in K tasks with $J(\theta_{\tau_i}, \tau_i) \approx J(\theta^*_{\tau_i}, \tau_i)$. Learning the parameterized skill Θ can be considered as a regression problem, trained with the pairs in E. While da Silva et al. [24] used Support Vector Regression for this regression task, we use Gaussian Process Regression (GPR) [22] since it naturally provides an uncertainty along with each prediction. Different ways of learning Ω from E are imaginable; for this paper, we only consider the case of diagonal Σ_τ with $(\Sigma_\tau)_{jj}$ either being proportional to the typical scale of the j-th component of $|\theta|$ in E or $(\Sigma_\tau)_{jj}$ being the uncertainty of the GPR's prediction for the j-th dimension of θ_τ.

We represent movement primitives by DMPs and use CMA-ES [8] for learning the DMP's parameters θ_{τ_i} in the training tasks $\tau_1, \ldots, \tau_K$ and thus generating E.

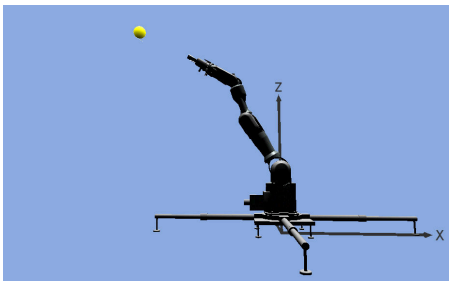

Fig. 4 Throwing a ball with a simulated Mitsubishi PA-10.

When faced with a new task τ, either the parameterized skill's prediction $\theta_\tau = \Theta(\tau)$ can be used as movement primitive parameters or the parameters can be learned by means of CMA-ES. While the former might suffer from generalization errors, the latter might require too many trials to be practical. Skill templates provide a reasonable compromise: instead of using the standard CMA-ES initialization, the skill template's prediction θ_τ can be used for the initial policy and the skill template's covariance Σ_τ as exploration matrix. This allows to explore more strongly in dimensions of the skill parameters where the GPR's prediction has larger uncertainty.

Results: We investigate how the prediction performance of parametrized skill depends on the size of the training set E. Furthermore, we empirically evaluate to which extent skill templates can reduce the sample complexity of policy search methods like CMA-ES. We give results on a physical simulation of a Mitsubishi PA-10 robot (see Fig. 4). The objective in this problem is to throw a simulated ball to an externally specified target position on the floor. Each task corresponds to a uniform-randomly sampled target position $\tau = (x_\tau, y_\tau) \in [-7, -1] \times [-5, 1]$, where the the unit is meter and the origin of the coordinate frame corresponds to the base of the PA-10. The robot is controlled in joint space and each joint's trajectory is determined by a separate DMP with 10 parameters. Additionally, the end position and velocity of each joint is learned. The resulting parameter vector θ is 96-dimensional. Reward is given only at the end of each trial based on the squared distance between the target position τ and the reached position (x, y), more precisely: $r = -||\tau - (x, y)||_2^2$. Note that there is typically not a single globally optimal solution. The skill weights θ_τ in the training set E have been learned using 200 rollouts with CMA-ES starting from initial weights which correspond to throwing the ball to position $(-3.5, -3.6)$.

The upper graph of Fig. 5 depicts the relation of the number of training tasks K and the performance (distance from target) for movement primitive weights

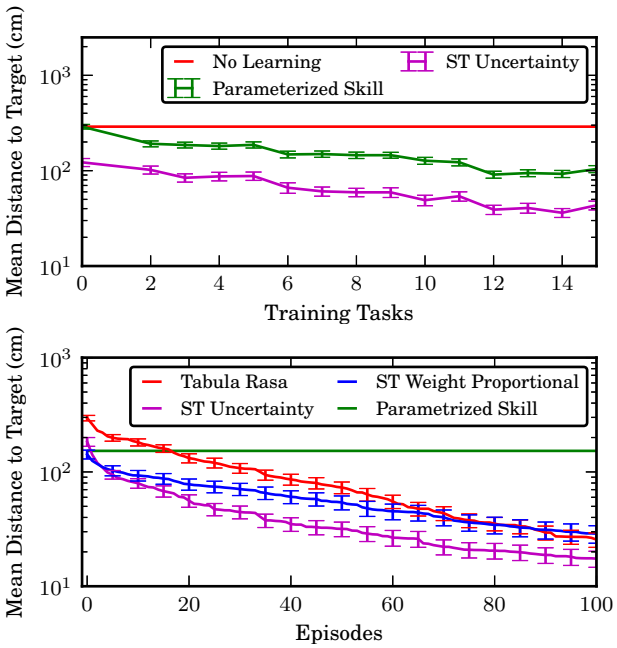

Fig. 5 Top: Quality of parametrized skill and skill template (after 25 episodes learning) for varying number of training tasks $|E| = K$. Bottom: Learning curves of CMA-ES under different initializations for $K = 10$. Shown are mean and standard error of mean of the distance to target averaged over a 8×8 grid over task space. Note the logarithmic scale.

learned with different approaches: weights predicted by the parameterized skill Θ and weights learned after 25 rollouts with CMA-ES starting from Θ's prediction and using the skill template's uncertainty for exploration ("ST Uncertainty"). The figure shows that (a) the parameterized skill's prediction improves with the number of training tasks K to a level considerably better than the baseline behavior ("No Learning") and (b) the skill template allows to learn considerable better weights for small K even after only 25 additional rollouts in the target task. Both curves show a similar pattern with the parameterized skill reaching a comparable level of performance as the skill template for approximately 6 more training tasks. Thus, 6 additional training tasks are worth approximately 25 rollouts in the target tasks.

The graph at the bottom of Fig. 5 depicts learning curves for $K = 10$. Shown are CMA-ES in the tabula rasa case and CMA-ES starting from the skill template with weight proportional Σ_τ ("ST Weight Proportional") and with Σ_τ based on the GPR's uncertainty. One can see that it takes tabula rasa CMA-ES approx. 20 rollouts to reach the parametrized skill's performance. The skill template with uncertainty exploration performs consistently better than tabula rasa learning. In contrast, weight-proportional exploration quickly loses the advantage compared to tabula rasa learning. Thus, the results show that learning the ex-

ploration matrix Σ_τ along with the prediction is important.

5 Conclusion and Outlook

We have outlined the concept of a learning platform which is developed in the project BesMan and presented first results on the learning of movement primitives and skill templates in a simulated robotic problem. The results show that effective transfer of learned movement primitives between related tasks is possible. Furthermore, exploiting the uncertainty in this transfer for controlling exploration increases the learning speed in the target task considerably. Since the proposed approach is not tailored to the specific problem, we expect that similar results can be obtained in other robotic problems.

Future work is to implement all parts of the learning platform and to test them in more realistic robotic scenarios with robots like AILA. Transfer learning and hierarchical approaches will play a key role in the learning of versatile and reusable skills. Furthermore, we will conduct behavioral studies with human subjects in order to investigate how movement primitives are transferred, adapted, and combined in human subjects. The resulting insights will be closely connected to the approaches implemented in the learning platform.

References

1. Abdenebaoui, L., Kirchner, E.A., Kassahun, Y., Kirchner, F.: A connectionist architecture for learning to play a simulated BRIO labyrinth game. In: Proceedings of the 30th Annual German Conference on Artificial Intelligence (KI07), pp. 427–430. Springer (2007)
2. Argall, B.D., Chernova, S., Veloso, M., Browning, B.: A survey of robot learning from demonstration. Robotics and Autonomous Systems **57**(5), 469–483 (2009)
3. Barto, A.G., Mahadevan, S.: Recent advances in hierarchical reinforcement learning. Discrete Event Dynamic Systems **13**(4), 341–379 (2003)
4. Daniel, C., Neumann, G., Peters, J.: Hierarchical relative entropy policy search. In: Proceedings of the 15th International Conference on Artificial Intelligence and Statistics, pp. 273–281 (2012)
5. Fearnhead, P., Liu, Z.: On-line inference for multiple change point models. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **69**, 589–605 (2007)
6. Fox, E., Sudderth, E., Jordan, M., Willsky, A.: Sharing features among dynamical systems with beta processes. In: Neural Information Processing Systems 22. MIT Press (2010)
7. Graybiel, A.: The basal ganglia and chunking of action repertoires. Neurobiol Learn Mem **70**(1-2), 119–36 (1998)
8. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9**, 159–195 (2001)
9. Heidrich-Meisner, V., Igel, C.: Evolution strategies for direct policy search. In: Parallel Problem Solving from Nature PPSN X, pp. 428–437 (2008)
10. Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S.: Dynamical movement primitives: learning attractor models for motor behaviors. Neural computation **25**(2), 328–373 (2013)
11. Khansari-Zadeh, S.M., Billard, A.: Learning stable nonlinear dynamical systems with gaussian mixture models. IEEE Transactions on Robotics (2013)
12. Kober, J., Bagnell, J.A., Peters, J.: Reinforcement learning in robotics: A survey. International Journal of Robotics Research (2013)
13. Kober, J., Muelling, K., Kroemer, O., Lampert, C.H., Scholkopf, B., Peters, J.: Movement templates for learning of hitting and batting. In: IEEE International Conference on Robotics and Automation (2010)
14. Kober, J., Oztop, E., Peters, J.: Reinforcement learning to adjust robot movements to new situations. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 1–6 (2011)
15. Kober, J., Peters, J.: Policy search for motor primitives in robotics. Machine Learning (2010)
16. Muelling, K., Kober, J., Kroemer, O., Peters, J.: Learning to select and generalize striking movements in robot table tennis. International Journal of Robotics Research **32**(3), 263–279 (2013)
17. Pastor, P., Hoffmann, H., Asfour, T., Schaal, S.: Learning and generalization of motor skills by learning from demonstration. In: Proceedings of the 2009 IEEE International Conference on Robotics and Automation, pp. 1293–1298 (2009)
18. Peters, J., Mülling, K., Altun, Y.: Relative entropy policy search. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (2010)
19. Peters, J., Mülling, K., Kober, J., Nguyen-Tuong, D., Krömer, O.: Robot skill learning. In: Proceedings of the European Conference on Artificial Intelligence (2012)
20. Peters, J., Schaal, S.: Reinforcement learning by reward-weighted regression for operational space control. In: Proceedings of the International Conference on Machine Learning, pp. 745–750 (2007)
21. Peters, J., Schaal, S.: Natural Actor-Critic. Neurocomputing **71**(7-9), 1180 – 1190 (2008)
22. Rasmussen, C., Williams, C.: Gaussian Processes for Machine Learning. MIT Press (2006)
23. Schaal, S.: Learning from demonstration. In: Advances in Neural Information Processing Systems 9, pp. 12–20 (1997)
24. da Silva, B.C., Konidaris, G., Barto, A.G.: Learning parameterized skills. In: Proceedings of the 29th International Conference on Machine Learning. Edinburgh, Scotland (2012)
25. Stulp, F., Schaal, S.: Hierarchical reinforcement learning with motion primitives. In: 11th IEEE-RAS International Conference on Humanoid Robots (2011)
26. Taylor, M., Stone, P.: Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research **10**, 1633–1685 (2009)
27. Theodorou, E., Buchli, J., Schaal, S.: A generalized path integral control approach to reinforcement learning. Journal of Machine Learning Research **11**, 3137–3181 (2010)