2009 Autonome Mobile Systeme (AMS)

Learning new basic Movements for Robotics

Jens Kober[1] and Jan Peters[1]

Max Planck Institute for Biological Cybernetics, Tübingen, Germany

Abstract. Obtaining novel skills is one of the most important problems in robotics. Machine learning techniques may be a promising approach for automatic and autonomous acquisition of movement policies. However, this requires both an appropriate policy representation and suitable learning algorithms. Employing the most recent form of the dynamical systems motor primitives originally introduced by Ijspeert et al. [1], we show how both discrete and rhythmic tasks can be learned using a concerted approach of both imitation and reinforcement learning, and present our current best performing learning algorithms. Finally, we show that it is possible to include a start-up phase in rhythmic primitives. We apply our approach to two elementary movements, i.e., Ball-in-a-Cup and Ball-Paddling, which can be learned on a real Barrett WAM robot arm at a pace similar to human learning.

1 Introduction

When humans learn new motor skills, e.g., paddling a ball with a table-tennis racket or hitting a tennis ball, it is highly likely that they are represented as elementary or primitive movements and use imitation as well as reinforcement learning [2]. In contrast, most robots are still programmed by a human operator using task and domain knowledge. Such programming is highly efficient but can also become very expensive and is limited to the considered situations. Learning techniques are a plausible alternative for more autonomous skill acquisition and improvement. Inspired by the biological insight, we will discuss the technical counterparts in this paper and show how both single-stroke and rhythmic tasks can be learned efficiently by mimicking the human presenter with subsequent reward-driven self-improvement.

Unfortunately however, off-the-shelf machine learning techniques do not scale into the high-dimensional domains of anthropomorphic robotics. Instead, robot learning requires methods that employ both representations and algorithms appropriate for the domain. If a favorable function approximator is chosen in this context, ideally one that is linear in its parameters, then learning can be sufficiently fast for application in robotics in real-time.

Recently, the idea of using dynamical systems as motor primitives was put forward by Ijspeert et al. [1, 3] as a general approach for representing control policies for basic movements. The resulting movement generation has a variety of favorable properties, i.e., rescalability with respect to both time and amplitude, basic stability properties and the possibility to encode either single-stroke

or rhythmic behaviors. Previous applications include a variety of different basic motor skills such as tennis swings [1], T-ball batting [4], planar biped walking [5], constrained reaching tasks [6] and even in tasks with potential industrial application [7]. Nevertheless, most of the previous work in motor primitive learning (with the exceptions of [4] and [6]) has focused on learning by imitation *without* subsequent self-improvement. In real life, a human demonstration is rarely ever perfect nor does it suffice for near-optimal performance. Thus, additional reinforcement learning is essential for both performance-based refinement and continuous adaptation of the presented skill.

In this paper, we present our current best performing setups for motor primitive learning with both the required methods for imitation and reinforcement learning. The appropriate imitation and reinforcement learning methods are given in Section 2. In Section 3, we show how the resulting framework can be applied to both learning *Ball-in-a-Cup* as a discrete task and *Ball-Paddling* as a rhythmic task on a real Barrett WAM[1]. The ball-paddling task is of particular interest as we show how the combination of different motor primitives is possible. It is among the first applications where both rhythmic and discrete dynamical systems motor primitives [1] are used in conjunction to achieve the task.

2 Learning Methods for Motor Primitives

It is likely that humans rely both on imitation and on reinforcement learning for learning new motor skills as both of these approaches have different functions in the learning process. Imitation learning has a given target and, thus, it allows to learn policies from the examples of a teacher. However, imitation learning can only reproduce a policy representing or generalizing an exhibited behavior. Self-improvement by trial-and-error with respect to an external reward signal can be achieved by reinforcement learning. Nevertheless, traditional reinforcement learning algorithms require exhaustive exploration of the state and action space. Given the high-dimensionality of the state-space of anthropomorphic robots (a seven degree of freedom robot defies exhaustive exploration), the "curse of dimensionality" [8] fully applies and we need to rely on local reinforcement learning methods which improve upon the preceding imitation instead of traditional 'brute force' approaches. To some extent, this mimicks how children acquire new motor skills with the teacher giving a demonstration while the child subsequently attempts to reproduce and improve the skill by trial-and-error. However, note that not every task requires reinforcement learning and some can be learned purely based on imitations. Nevertheless, few tasks are known which are directly learned by reinforcement learning without preceding mimicking [9]. Thus, we first review how to do imitation learning with dynamical systems motor primitives in Section 2.1 and, subsequently, we show how reinforcement learning can be applied in this context in Section 2.2. The latter section will outline our reinforcement learning algorithm for the application in motor primitive learning.

[1] Accompanying video: http://www.youtube.com/watch?v=cNyoMVZQdYM

2.1 Imitation Learning for Dynamical Motor Primitives

In the presented framework, we initialize the motor primitives by imitation learning as in [9]. This step can be performed efficiently in the context of dynamical systems motor primitives as they represent a deterministic policy in the form $\bar{\mathbf{a}} = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\mu}(\mathbf{s})$, where $\boldsymbol{\mu}(\mathbf{s})$ are basis functions [9] depending on the state $\mathbf{s}$ (namely positions, velocities and a phase variable), $\boldsymbol{\theta} \in \mathbb{R}^N$ are policy parameters and $\bar{\mathbf{a}}$ are the actions (namely desired positions, velocities and accelerations). This policy is linear in parameters, thus, we have a standard locally-weighted linear regression problem that can be solved straightforwardly. This general approach has originally been suggested in [1]. Estimating the parameters of the dynamical system is slightly more daunting, i.e., the movement duration of discrete movements is extracted using motion detection and the time-constant is set accordingly. Similarly, the base period for the rhythmic dynamical motor primitives was extracted using first repetitions and, again, the time-constants are set accordingly. As the start-up phase in rhythmic presentations may deviate significantly from the periodic movement, the baseline of the oscillation often needs to be estimated based on the later part of the recorded movement, the amplitude is determined as the mean of the amplitudes of individual oscillations in this part.

2.2 Reinforcement Learning with PoWER

Reinforcement learning [10] of motor primitives is a very specific type of learning problem where it is hard to apply generic reinforcement learning algorithms [4, 11]. For this reason, the focus of this paper is largely on novel domain-appropriate reinforcement learning algorithms which operate on parametrized policies for episodic control problems.

Reinforcement Learning Setup When modeling our problem as a reinforcement learning problem, we always have a high-dimensional state $\mathbf{s}$ and as a result, standard RL methods which discretize the state-space can no longer be applied. The action $\mathbf{a} = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\mu}(\mathbf{s}) + \boldsymbol{\epsilon}$ is the output of our motor primitives augmented by the exploration $\boldsymbol{\epsilon}$. As a result, we have a stochastic policy $\mathbf{a} \sim \pi(\mathbf{s})$ with parameters $\boldsymbol{\theta}$ which can be seen as a distribution over the actions given the states. After a next time-step δt, the actor transfers to a state $\mathbf{s}_{t+1}$ and receives a reward r_t. As we are interested in learning complex motor tasks consisting of a single stroke or a rhythmically repeating movement, we focus on finite horizons of length T with episodic restarts [10]. While the policy representation is substantially different, the rhythmic movement resembles a repeated episodic movement in the reinforcement learning process. The general goal in reinforcement learning is to optimize the *expected return* of the policy with parameters $\boldsymbol{\theta}$ defined by $J(\boldsymbol{\theta}) = \int_{\mathbb{T}} p(\boldsymbol{\tau})R(\boldsymbol{\tau})d\boldsymbol{\tau}$, where $\boldsymbol{\tau} = [\mathbf{s}_{1:T+1}, \mathbf{a}_{1:T}]$ denotes a sequence of states $\mathbf{s}_{1:T+1} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{T+1}]$ and actions $\mathbf{a}_{1:T} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_T]$, the probability of an episode $\boldsymbol{\tau}$ is denoted by $p(\boldsymbol{\tau})$ and $R(\boldsymbol{\tau})$ refers to the return of an episode $\boldsymbol{\tau}$ and $\mathbb{T}$ is the set of all possible paths. Using the Markov assumption, we can write the

path distribution as $p(\boldsymbol{\tau}) = p(\mathbf{s}_1)\prod_{t=1}^{T+1} p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)\pi(\mathbf{a}_t|\mathbf{s}_t, t)$ where $p(\mathbf{s}_1)$ denotes the initial state distribution and $p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ is the next state distribution conditioned on last state and action. Similarly, if we assume additive, accumulated rewards, the return of a path is given by $R(\boldsymbol{\tau}) = \frac{1}{T}\sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, t)$, where $r(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, t)$ denotes the immediate reward.

While episodic Reinforcement Learning (RL) problems with finite horizons are common in motor control, few methods exist in the RL literature (notable exceptions are model-free method such as Episodic REINFORCE [12] and the Episodic Natural Actor-Critic eNAC [4] as well as model-based methods, e.g., using differential-dynamic programming [13]). In order to avoid learning of complex models, we focus on model-free methods and, to reduce the number of open parameters, we rather use a novel Reinforcement Learning algorithm which is based on expectation-maximization [14]. Our new algorithm is called Policy learning by Weighting Exploration with the Returns (PoWER) and can be derived from the same higher principle as previous policy gradient approaches, see [15] for details.

Policy learning by Weighting Exploration with the Returns (PoWER) When learning motor primitives, we intend to learn a deterministic mean policy $\bar{\mathbf{a}} = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\mu}(\mathbf{s})$ which is linear in parameters $\boldsymbol{\theta}$ and augmented by additive exploration $\boldsymbol{\epsilon}(\mathbf{s}, t)$ in order to make model-free reinforcement learning possible. As a result, the explorative policy can be given in the form $\mathbf{a} = \boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\mu}(\mathbf{s}, t) + \boldsymbol{\epsilon}(\boldsymbol{\mu}(\mathbf{s}, t))$. Previous work in [4, 11], has focused on state-independent, white Gaussian exploration, i.e., $\boldsymbol{\epsilon}(\boldsymbol{\mu}(\mathbf{s}, t)) \sim \mathcal{N}(0, \Sigma)$, and has resulted into applications such as T-Ball batting [4] and constrained movement [6]. Alternatively, as introduced by [16], one could generate a form of structured, state-dependent exploration $\boldsymbol{\epsilon}(\boldsymbol{\mu}(\mathbf{s}, t)) = \boldsymbol{\varepsilon}_t^{\mathrm{T}}\boldsymbol{\mu}(\mathbf{s}, t)$ with $[\boldsymbol{\varepsilon}_t]_{ij} \sim \mathcal{N}(0, \sigma_{ij}^2)$, where σ_{ij}^2 are meta-parameters of the exploration that can be optimized in a similar manner. Each σ_{ij}^2 corresponds to one θ_{ij}. This argument results into the policy $\mathbf{a} \sim \pi(\mathbf{a}_t|\mathbf{s}_t, t) = \mathcal{N}(\mathbf{a}|\boldsymbol{\mu}(\mathbf{s}, \mathbf{t}), \hat{\boldsymbol{\Sigma}}(\mathbf{s}, t))$. This form of policies improves upon shortcomings of directly perturbed policies. Based on the EM updates for Reinforcement Learning as suggested in [11, 15], we can derive the update rule

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \frac{E_{\boldsymbol{\tau}}\left\{\sum_{t=1}^{T} \boldsymbol{\varepsilon}_t Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t, t)\right\}}{E_{\boldsymbol{\tau}}\left\{\sum_{t=1}^{T} Q^{\pi}(\mathbf{s}_t, \mathbf{a}_t, t)\right\}}, \tag{1}$$

where $Q^{\pi}(\mathbf{s}, \mathbf{a}, t)$ is the state-action value function. Note that this algorithm does not need the learning rate as a meta-parameter. In order to reduce the number of trials in this on-policy scenario, we reuse the trials through importance sampling [10].

3 Robot Evaluation

The methods presented in this paper are evaluated on two learning problems on a real, seven degree of freedom Barrett WAM, i.e., we learn the discrete task of

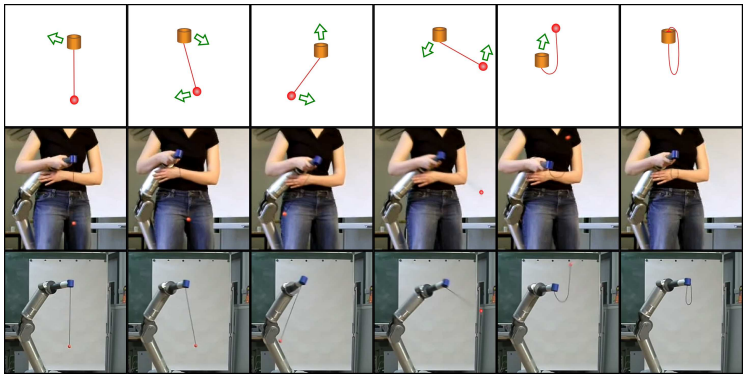

Fig. 1. This figure shows schematic drawings of the *Ball-in-a-Cup* motion, the final learned robot motion as well as a kinesthetic teach-in. The green arrows show the directions of the current movements in that frame. The human cup motion was taught to the robot by imitation learning with 31 parameters per joint for an approximately three seconds long trajectory. The robot manages to reproduce the imitated motion quite accurately, but the ball misses the cup by several centimeters. After roughly 75 rollouts, we have good performance and at the end of the 100 rollouts we have virtually no failures anymore.

Ball-in-a-Cup and the rhythmic task *Ball-Paddling*. The resulting simplicity and speed of the learning process demonstrate the suitability of the motor primitive-based learning framework for practical application.

3.1 Discrete Movement: Ball-in-a-Cup

The children motor game Ball-in-a-Cup, also known as Balero and Bilboquet [17] is challenging even for a grown up. The toy has a small cup which is held in one hand (or, in our case, is attached to the end-effector of the robot) and the cup has a small ball hanging down on a string (the string has a length of 40cm for our toy). Initially, the ball is hanging down vertically in a rest position. The player needs to move fast in order to induce a motion in the ball through the string, toss it up and catch it with the cup, a possible movement is illustrated in Figure 1 in the top row.

The state of the system can be described by joint angles and joint velocities of the robot as well as the the Cartesian coordinates and velocities of the ball. The actions are the joint space accelerations where each of the seven joints is driven by a separate motor primitive with one common canonical system. The movement uses all seven degrees of freedom and is not on a plane. All motor primitives are perturbed separately but employ the same joint final reward. The reward is based on the minimal distance between the center of the ball and the center of the cup.

Due to the complexity of the task, Ball-in-a-Cup is even a hard motor task for children who usually only succeed after observing another person presenting a demonstration first, and after subsequent trial-and-error-based learning. Mimicking how children learn to play Ball-in-a-Cup, we first initialize the motor primitives by imitation and, subsequently, improve them by reinforcement learning. We recorded the motions of a human player by kinesthetic teach-in in order to obtain an example for imitation as shown in Figure 1 (middle row). As expected, the robot fails to reproduce the presented behavior even if we use all the recorded details for the imitation. Thus, reinforcement learning is needed for self-improvement. We determined by cross-validation that 31 shape-parameters per motor primitive are needed.

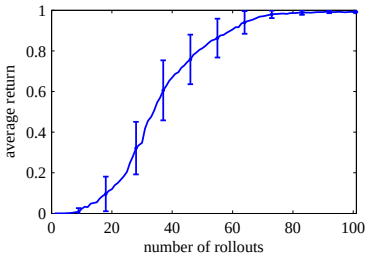

Fig. 2. This figure shows the expected return of the learned policy in the Ball-in-a-Cup evaluation averaged over 20 runs.

In [15] we benchmarked our novel algorithm and several widely used algorithms on tasks having characteristics similar to this one. As a result we employ our best algorithm, PoWER. Figure 2 shows the expected return over the number of rollouts where convergence to a maximum is clearly recognizable. The robot regularly succeeds at bringing the ball into the cup after approximately 75 rollouts. A nine year old child got the ball in the cup for the first time after 35 trials while the robot got the ball in for the first time after 42 rollouts. However, after 100 trials, the robot exhibits perfect runs in every single trial while, from our experience, the child does not have a comparable success rate. Of course, such a comparison with a child is contrived as a robot can precisely reproduce movements unlike any human being and that children can most likely adapt faster to changes in the setup.

3.2 Rhythmic Movement with start-up phase: Ball-Paddling

In Ball-Paddling, we have a table-tennis ball that is attached to a table-tennis paddle by an elastic string. The goal is to have the ball bouncing above the paddle. The string avoids that the ball is falling down but also pulls the ball back towards the center of the paddle if the ball is hit sufficiently hard (i.e., the string is also stretched sufficiently as a consequence). The task is fairly easy to perform open-loop once the player has determined appropriate amplitude and frequency for the motion. Furthermore, the task is robust to small changes of these parameters as well as to small perturbations of the environment. We again recorded the motions of a human player using kinesthetic teach-in in order to obtain a demonstration for imitation learning as shown in Figure 3. From the imitation, it can be determined by cross-validation that 10 shape-parameters per motor primitive are sufficient.

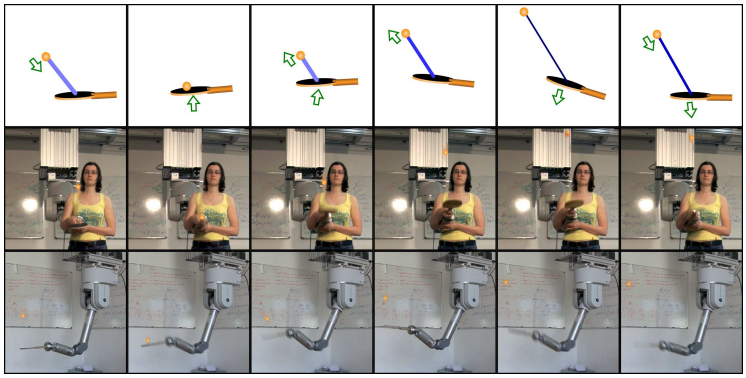

Fig. 3. This figure shows schematic drawings of the *Ball-Paddling* motion, a kinesthetic teach-in as well as the performance of the robot after imitation learning. When the string is stretched it is shown as thinner and darker. The human demonstration was taught to the robot by imitation learning with 10 parameters per joint for the rhythmic motor primitive. An additional discrete motor primitive is used for the start-up phase. Please see Section 3.2 and the accompanying video for details.

However, as we start with a still robot where the ball rests on the paddle, we require a start-up phase in order to perform the task successfully. This initial motion has to induce more energy in order to get the motion started and to extend the string sufficiently. For our setup, the start-up phase consists (as exhibited by the teacher's movements) of moving the paddle slower and further up than during the rhythmic behavior. This kind of movement can easily be achieved in the dynamical systems motor primitives framework by imposing another discrete primitive that gradually adapts the period parameter globally and the amplitude modifier to the ones encountered in the rhythmic behavior. The discrete modifier motor primitive is applied additively to the two parameters. The goal parameter of this modifier primitive is zero and thus, its influence vanishes after the initialization. With this start-up phase, imitation learning from demonstrations suffices to reproduce the motor skill successfully. To our knowledge, this application is probably the first where both rhythmic and discrete dynamical systems primitives are used together to achieve a particular task.

4 Conclusion

In this paper, we present both novel learning algorithms and experiments using the dynamical systems motor primitive [1, 9]. For doing so, we have discussed both appropriate imitation learning methods by locally weighted regression and derived our currently best-suited reinforcement learning algorithm for this framework, i.e., Policy learning by Weighting Exploration with the Returns (PoWER). We show that two complex motor tasks, i.e., *Ball-in-a-Cup* and *Ball-Paddling*,

can be learned on a real, physical Barrett WAM using the methods presented in this paper. Of particular interest is the Ball-Paddling application as it requires the combination of both rhythmic and discrete dynamical systems primitives in order to achieve a particular task.

References

1. A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems (NIPS)*, 2003.
2. T. Flash and B. Hochner, "Motor primitives in vertebrates and invertebrates," *Current Opinions in Neurobiology*, vol. 15, pp. 660–666, 2005.
3. S. Schaal, J. Peters, J. Nakanishi, and A. J. Ijspeert, "Learning movement primitives," in *International Symposium on Robotics Research 2003 (ISRR)*, ser. Springer Tracts in Advanced Robotics, 2004, pp. 561–572.
4. J. Peters and S. Schaal, "Policy gradient methods for robotics," in *Proceedings of the IEEE/RSJ 2006 International Conference on Intelligent RObots and Systems (IROS)*, 2006, pp. 2219 – 2225.
5. S. Schaal, J. Peters, J. Nakanishi, and A. J. Ijspeert, "Control, planning, learning, and imitation with dynamic movement primitives," in *Proceedings of the Workshop on Bilateral Paradigms on Humans and Humanoids, IEEE 2003 International Conference on Intelligent RObots and Systems (IROS)*, 2003.
6. 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.
7. H. Urbanek, A. Albu-Schäffer, and P.v.d.Smagt, "Learning from demonstration repetitive movements for autonomous service robotics," in *Proceedings of the IEEE/RSL 2004 International Conference on Intelligent RObots and Systems (IROS)*, 2004, pp. 3495–3500.
8. R. E. Bellman, *Dynamic Programming*. Princeton University Press, 1957.
9. S. Schaal, P. Mohajerian, and A. J. Ijspeert, "Dynamics systems vs. optimal control — a unifying view," *Progress in Brain Research*, vol. 165, no. 1, pp. 425–445, 2007.
10. R. Sutton and A. Barto, *Reinforcement Learning*. MIT PRESS, 1998.
11. J. Peters and S. Schaal, "Reinforcement learning for operational space," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2007.
12. R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.
13. C. G. Atkeson, "Using local trajectory optimizers to speed up global optimization in dynamic programming," in *Advances in Neural Information Processing Systems 6 (NIPS)*, 1994.
14. P. Dayan and G. E. Hinton, "Using expectation-maximization for reinforcement learning," *Neural Computation*, vol. 9, no. 2, pp. 271–278, 1997.
15. J. Kober and J. Peters, "Policy search for motor primitives in robotics," in *Advances in Neural Information Processing Systems (NIPS)*, 2008.
16. T. Rückstieß, M. Felder, and J. Schmidhuber, "State-dependent exploration for policy gradient methods," in *Proceedings of the European Conference on Machine Learning (ECML)*, 2008, pp. 234–249.
17. Wikipedia, "Ball-in-a-cup," January 2009. [Online]. Available: http://en.wikipedia.org/wiki/Ball_in_a_cup