Towards Machine Learning of Motor Skills

Jan Peters[1,2], Stefan Schaal[2] and Bernhard Schölkopf[1]

(1) Max-Planck Institute for Biological Cybernetics, Spemannstr. 32, 72074 Tübingen
(2) University of Southern California, 3641 Watt Way, Los Angeles, CA 90802

Abstract. Autonomous robots that can adapt to novel situations has been a long standing vision of robotics, artificial intelligence, and cognitive sciences. Early approaches to this goal during the heydays of artificial intelligence research in the late 1980s, however, made it clear that an approach purely based on reasoning or human insights would not be able to model all the perceptuomotor tasks that a robot should fulfill. Instead, new hope was put in the growing wake of machine learning that promised fully adaptive control algorithms which learn both by observation and trial-and-error. However, to date, learning techniques have yet to fulfill this promise as only few methods manage to scale into the high-dimensional domains of manipulator robotics, or even the new upcoming trend of humanoid robotics, and usually scaling was only achieved in precisely pre-structured domains. In this paper, we investigate the ingredients for a general approach to motor skill learning in order to get one step closer towards human-like performance. For doing so, we study two major components for such an approach, i.e., firstly, a theoretically well-founded general approach to representing the required control structures for task representation and execution and, secondly, appropriate learning algorithms which can be applied in this setting.

1 Introduction

Despite an increasing number of motor skills exhibited by manipulator and humanoid robots, the general approach to the generation of such motor behaviors has changed little over the last decades [2,11]. The roboticist models the task as accurately as possible and uses human understanding of the required motor skills in order to create the desired robot behavior as well as to eliminate all uncertainties of the environment. In most cases, such a process boils down to recording a desired trajectory in a pre-structured environment with precisely placed objects. If inaccuracies remain, the engineer creates exceptions using human understanding of the task. While such highly engineered approaches are feasible in well-structured industrial or research environments, it is obvious that if robots should ever leave factory floors and research environments, we will need to reduce or eliminate the strong reliance on hand-crafted models of the environment and the robots exhibited to date. Instead, we need a general approach which allows us to use compliant robots designed for interaction with less structured and uncertain environments in order to reach domains outside industry.

Such an approach cannot solely rely on human knowledge but instead has to be acquired and adapted from data generated both by human demonstrations of the skill as well as trial and error of the robot.

The tremendous progress in machine learning over the last decades offers us the promise of less human-driven approaches to motor skill acquisition. However, despite offering the most general way of thinking about data-driven acquisition of motor skills, generic machine learning techniques, which do not rely on an understanding of motor systems, often do not scale into the domain of manipulator or humanoid robotics due to the high domain dimensionality. Therefore, instead of attempting an unstructured, monolithic machine learning approach to motor skill aquisition, we need to develop approaches suitable for this particular domain with the inherent problems of task representation, learning and execution addressed separately in a coherent framework employing a combination of imitation, reinforcement and model learning in order to cope with the complexities involved in motor skill learning. The advantage of such a concerted approach is that it allows the separation of the main problems of motor skill acquisition, refinement and control. Instead of either having an unstructured, monolithic machine learning approach or creating hand-crafted approaches with pre-specified trajectories, we are capable of aquiring skills, represented as policies, from demonstrations and refine them using trial and error. Using learning-based approaches for control, we can achieve accurate control without needing accurate models of the complete system.

2 Foundations for Motor Skill Learning

The principal objective of this paper is to find the foundations for a general framework for representing, learning and executing motor skills for robotics. As can be observed from this question, the major goal of this paper requires three building blocks, i.e., (i) appropriate representations for movements, (ii) learning algorithms which can be applied to these representations and (iii) a transformation which allows the execution of the kinematic policies in the respective task space on robots.

2.1 Essential Components

We address the three essential components, i.e., representation, learning and execution. In this section, we briefly outline the underlying fundamental concepts.

Representation. For the representation of motor skills, we can rely on the insight that humans, while being capable of performing a large variety of complicated movements, restrict themselves to a smaller amount of primitive motions [10]. As suggested by Ijspeert et al. [4,3], such primitive movements can be represented by nonlinear dynamic systems. We can represent these in the differential constraint form given by

$$\boldsymbol{A}_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, t)\ddot{\boldsymbol{x}} = \boldsymbol{b}_{\boldsymbol{\theta}_i}(\boldsymbol{x}_i, \dot{\boldsymbol{x}}_i, t), \tag{1}$$

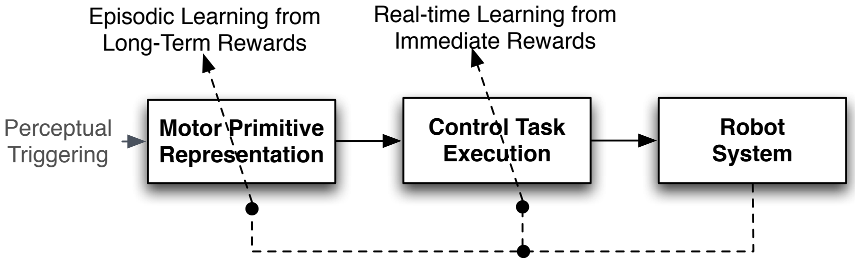

Fig. 1. This figure illustrates our general approach to motor skill learning by dividing it into motor primitive and a motor control component. For the task execution, fast policy learning methods based on observable error need to be employed while the task learning is based on slower episodic learning.

where $i \in \mathbb{N}$ is the index of the motor primitive in a library of movements, $\boldsymbol{\theta}_i \in \mathbb{R}^L$ denote the parameters of the primitive i, t denotes time and $\boldsymbol{x}_i$,$\dot{\boldsymbol{x}}_i$,$\ddot{\boldsymbol{x}}_i \in \mathbb{R}^n$ denote positions, velocities and accelerations of the dynamic system, respectively.

Learning. Learning basic motor skills[1] is achieved by adapting the parameters $\boldsymbol{\theta}_i$ of motor primitive i. The high dimensionality of our domain prohibits the exploration of the complete space of all admissible motor behaviors, rendering the application of machine learning techniques which require exhaustive exploration impossible. Instead, we have to rely on a combination of supervised and reinforcement learning in order to aquire motor skills where the supervised learning is used in order to obtain the initialization of the motor skill while reinforcement learning is used in order to improve it. Therefore, the aquisition of a novel motor task consists out of two phases,i.e., the 'learning robot' attempts to reproduce the skill acquired through supervised learning and improve the skill from experience by trial-and-error, i.e., through reinforcement learning.

Execution. The execution of motor skills adds another level of complexity. It requires that a mechanical system

$$\boldsymbol{u} = \boldsymbol{M}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)\ddot{\boldsymbol{q}} + \boldsymbol{F}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t), \tag{2}$$

with a mapping $\boldsymbol{x}_i = \boldsymbol{f}_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$ can be forced to execute each motor primitive $\boldsymbol{A}_i\ddot{\boldsymbol{x}}_i = \boldsymbol{b}_i$ in order to fulfill the skill. The motor primitive can be viewed as a mechanical constraint acting upon the system, enforced through accurate computation of the required forces based on analytical models. However, in most cases it is very difficult to obtain accurate models of the mechanical system. Therefore it can be more suitable to find a policy learning approach which replaces the control law based on the hand-crafted rigid body model. In this paper,

[1] Learning by sequencing and parallelization of the motor primitives will be treated in future work.

we will follow this approach which forms the basis for understanding motor skill learning.

2.2 Resulting Approach

As we have outlined during the discussion of our objective and its essential components, we require an appropriate general motor skill framework which allows us to separate the desired task-space movement generation (represented by the motor primitives) from movement control in the respective actuator space. Based on the understanding of this transformation from an analytical point of view on robotics, we presente a learning framework for task execution in operational space. For doing so, we have to consider two components, i.e., we need to determine how to learn the desired behavior represented by the motor primitives as well as the execution represented by the transformation of the motor primitives into motor commands. We need to develop scalable learning algorithms which are both appropriate and efficient when used with the chosen general motor skill learning architecture. Furthermore, we require algorithms for fast immediate policy learning for movement control based on instantly observable rewards in order to enable the system to cope with real-time improvement during the execution. The learning of the task itself on the other hand requires the learning of policies which define the long-term evolution of the task, i.e., motor primitives, which are learned on a trial-by-trial basis with episodic improvement using a teacher for demonstration and reinforcement learning for self-improvement. The resulting general concept underlying this paper is illustrated in Figure 1.

2.3 Novel Learning Algorithms

As outlined before, we need two different styles of policy learning algorithms, i.e., methods for long-term reward optimization and methods for immediate improvement. Thus, we have developed two different classes of algorithms, i.e., the Natural Actor-Critic and the Reward-Weighted Regression.

Natural Actor-Critic. The Natural Actor-Critic algorithms [8,9] are the fastest policy gradient methods to date and “the current method of choice” [1]. They rely on the insight that we need to maximize the reward while keeping the loss of experience constant, i.e., we need to measure the distance between our current path distribution and the new path distribution created by the policy. This distance can be measured by the Kullback-Leibler divergence and approximated using the Fisher information metric resulting in a natural policy gradient approach. This natural policy gradient has a connection to the recently introduced compatible function approximation, which allows to obtain the Natural Actor-Critic. Interestingly, earlier Actor-Critic approaches can be derived from this new approach. In application to motor primitive learning, we can demonstrate that the Natural Actor-Critic outperforms both finite-difference gradients as well as ‘vanilla’ policy gradient methods with optimal baselines.

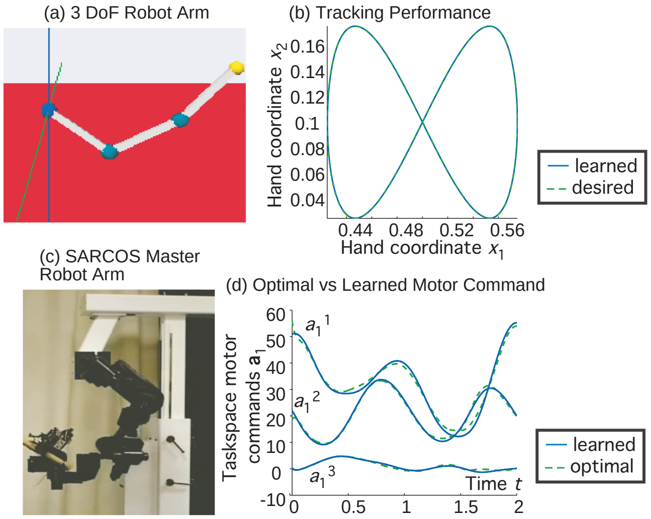

Fig. 2. Systems and results of evaluations for learning operational space control: (a) screen shot of the 3 DOF arm simulator, (c) Sarcos robot arm, used as simulated system and for actual robot evaluations in progress. (b) Tracking performance for a planar figure-8 pattern for the 3 DOF arm, and (d) comparison between the analytically obtained optimal control commands in comparison to the learned ones for one figure-8 cycle of the 3DOF arm.

Reward-Weighted Regression. In contrast to Natural Actor-Critic algorithms, the Reward-Weighted Regression algorithm [6,5,7] focuses on immediate reward improvement and employs an adaptation of the expectation maximization (EM) algorithm for reinforcement learning instead of a gradient based approach. The key difference here is that when using immediate rewards, we can learn from our actions directly, i.e., use them as training examples similar to a supervised learning problem with a higher priority for samples with a higher reward. Thus, this problem is a reward-weighted regression problem, i.e., it has a well-defined solution which can be obtained using established regression techniques. While we have given a more intuitive explanation of this algorithm, it corresponds to a properly derived maximization-maximization (MM) algorithm which maximizes a lower bound on the immediate reward similar to an EM algorithm. Our applications show that it scales to high dimensional domains and learns a good policy without any imitation of a human teacher.

3 Robot Application

The general setup presented in this paper can be applied in robotics using analytical models as well as the presented learning algorithms. The applications

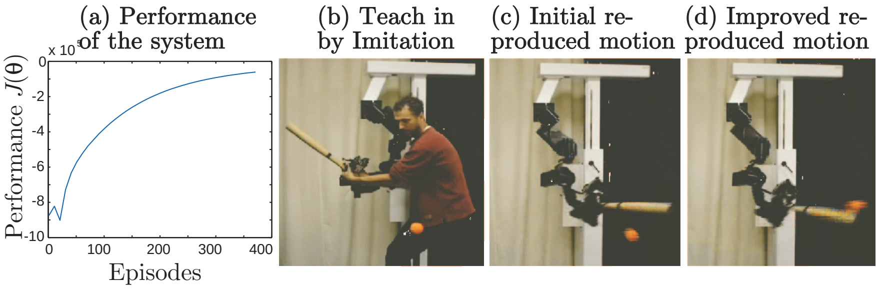
(a) Performance of the system
(b) Teach in by Imitation
(c) Initial reproduced motion
(d) Improved reproduced motion

Fig. 3. This figure shows (a) the performance of a baseball swing task when using the motor primitives for learning. In (b), the learning system is initialized by imitation learning, in (c) it is initially failing at reproducing the motor behavior, and (d) after several hundred episodes exhibiting a nicely learned batting.

presented in this paper include motor primitive learning and operational space control.

3.1 Learning Operational Space Control

Operational space control is one of the most general frameworks for obtaining task-level control laws in robotics. In this paper, we present a learning framework for operational space control which is a result of a reformulation of operational space control as a general point-wise optimal control framework and our insights into immediate reward reinforcement learning. While the general learning of operational space controllers with redundant degrees of freedom is non-convex and thus global supervised learning techniques cannot be applied straightforwardly, we can gain two insights, i.e., that the problem is locally convex and that our point-wise cost function allows us to ensure global consistency among the local solutions. We show that this can yield the analytically determined optimal solution for simulated three degrees of freedom arms where we can sample the state-space sufficiently. Similarly, we can show the framework works well for simulations of the both three and seven degrees of freedom robot arms as presented in Figure 2.

3.2 Motor Primitive Improvement by Reinforcement Learning

The main application of our long-term improvement framework is the optimization of motor primitives. Here, we follow essentially the previously outlined idea of acquiring an initial solution by supervised learning and then using reinforcement learning for motor primitive improvement. For this, we demonstrate both comparisons of motor primitive learning with different policy gradient methods, i.e., finite difference methods, 'vanilla' policy gradient methods and the Natural Actor-Critic, as well as an application of the most successful method, the Natural Actor-Critic to T-Ball learning on a physical, anthropomorphic SARCOS Master Arm, see Figure 3.

4 Conclusion

In conclusion, in this paper, we have preseted a general framework for learning motor skills which is based on a thorough, analytically understanding of robot task representation and execution. We have introduced two classes of novel reinforcement learning methods, i.e., the Natural Actor-Critic and the Reward-Weighted Regression algorithm. We demonstrate the efficiency of these reinforcement learning methods in the application of learning to hit a baseball with an anthropomorphic robot arm on a physical SARCOS master arm using the Natural Actor-Critic, and in simulation for the learning of operational space with reward-weighted regression.

References

1. Douglas Aberdeen. POMDPs and policy gradients. In *Proceedings of the Machine Learning Summer School (MLSS)*, Canberra, Australia, 2006.
2. J.J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Prentice Hall, Upper Saddle River, NJ, 2005.
3. A. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 1547–1554, Cambridge, MA, 2003. MIT Press.
4. J. A. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Washinton, DC, May 11-15 2002.
5. J. Peters and S. Schaal. Learning operational space control. In *Proceedings of Robotics: Science and Systems (RSS)*, Philadelphia, PA, 2006.
6. J. Peters and S. Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2007.
7. J. Peters and S. Schaal. Reinforcement learning for operational space. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Rome, Italy, 2007.
8. J. Peters, S. Vijayakumar, and S. Schaal. Reinforcement learning for humanoid robotics. In *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)*, Karlsruhe, Germany, September 2003.
9. J. Peters, S. Vijayakumar, and S. Schaal. Natural actor-critic. In *Proceedings of the 16th european conference on machine learning (ecml 2005)*, pages 280–291. springer, 2005.
10. S. Schaal, A. Ijspeert, and A. Billard. Computational approaches to motor learning by imitation. In C. D. Frith and D. Wolpert, editors, *The Neuroscience of Social Interaction*, pages 199–218. Oxford University Press, Oxford, UK, 2004.
11. L. Sciavicco and B. Siciliano. *Modeling and control of robot manipulators*. MacGraw-Hill, Heidelberg, Germany, 2007.