ISR / ROBOTIK 2010
In Proceedings of Joint 41th International Symposium on Robotics and 6th German Conference on Robotics, Munich, June 2010.

Learning Motion Skills from Expert Demonstrations and Own Experience using Gaussian Process Regression

Kathrin Gräve, Jörg Stückler, and Sven Behnke
Autonomous Intelligent Systems Group, Institute of Computer Science VI, University of Bonn, Germany

Abstract

While today's industrial robots achieve high performance in a wide range of tasks, the programming and adaptation of such robots to new tasks is still time-consuming and far from convenient. This shortcoming is a major obstacle for the deployment of robots in small to medium-size companies and also for service applications. Robot learning by imitation has been identified as an important paradigm to tackle this practicability issue. However, such approaches are only applicable in controlled static scenarios. To perform less constrained tasks, the robot needs the ability to generalize its skills and to adapt to changing conditions. It is thus a natural idea to formulate the problem as learning from experience and to incorporate demonstrations by an expert in the learning process. We present a new framework for learning of motion skills that seamlessly integrates demonstration learning from an expert teacher and further optimization of the demonstrated skill by own experience. Our method employs Gaussian Process regression to generalize a measure of performance to new skills that are close to already learned skills in the state-action space. We evaluate our approach for the task of grasping an object.

1 Introduction

Today's industrial mass production would not be possible without the invention of robots that efficiently carry out repetitive manufacturing tasks. These robots usually work in an isolated, static environment and are programmed to fulfil a single, specific task. In the future, robots may be employed to relieve humans of even more tasks which are cumbersome, monotone or even dangerous for humans in industrial, as well as service applications. These robots will have to be flexible enough to easily adapt to new tasks and unexpected changes in the environment.

As it is not feasible to preprogram the robot for every situation it may ever encounter, the development of intuitive ways to *teach* a robot is of central importance in this area of research. For personal robots, this also requires intuitive interfaces that are accessible to inexperienced users and allow owners of robots to adapt them to their personal needs. In order to achieve this, we combine imitation and reinforcement learning in a single coherent framework. Both ways of learning are well known from human teaching psychology. Applying them to robot learning allows a teacher to intuitively train a robot.

In our approach, the actions generated by either learning module are executed by the robot and their performance is measured as a scalar reward. We assume that the reward is continuous over the combined space of states and actions and apply Gaussian Process regression to approximate its value over the entire space. This allows to generalize experiences from known situations. Furthermore, it provides a measure of uncertainty regarding the achievable reward in any situation. In each situation, we decide for imitation or reinforcement learning based on a measure that trades off large reward and predictability of actions.

For imitation learning, we extract the parameters of a controller from the demonstrated trajectory. This controller is able to generate anthropomorphic arm movements from a small set of parameters. For reinforcement learning, we determine new actions that promise high reward but also offer a predictable outcome.

The advantages of our approach are manifold: The way in which we integrate both learning paradigms allows each of them to mitigate the other's shortcomings and improves the overall learning quality. The imitaition learning method narrows the search space of reinforcement learning which greatly improves learning speed. Furthermore, it allows to acquire anthropomorphic movements from human demonstrations. This offers a huge advantange over manual programming as this kind of movements is usually complex and it is very hard to describe what makes a movement human-like. Using reinforcement learning helps to reduce the number of demonstrations that are required to successfully learn a task. Furthermore, it is used to improve the results of imitation learning and to compensate for the kinematic differences between the human and the robot. In effect, our proposed learning method makes robot training accessible to non-experts in robotics or programming.

2 Related Work

Imitation learning methods comprise a set of supervised learning algorithms in which the teacher does not give the

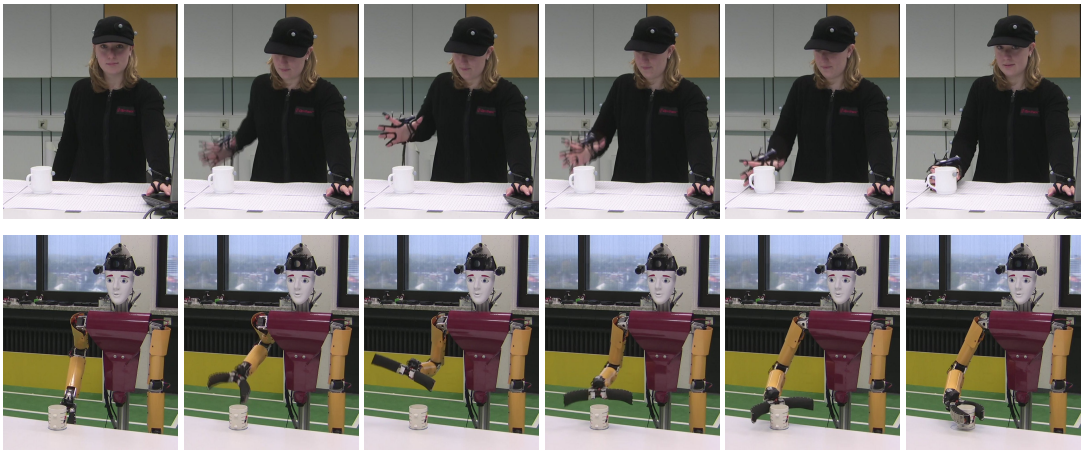

Figure 1: The teacher demonstrates a grasping movement (top). From the recorded trajectory, our approach extracts a parameterized motion primitive. The robot imitates the grasp by executing the extracted motion (bottom).

final solution to a problem but rather demonstrates the necessary steps to achieve that solution. Although imitation learning has been studied for decades, its mechanisms are not yet fully understood and there are many open questions that have been grouped into the broad categories relating to the questions of *whom to imitate*, *what to imitate*, *how to imitate* and *when to imitate* [1].

Consequently, there are many approaches to apply imitation learning to robots [1]. In early approaches to programming by demonstration [2], motion is recorded by teach-in through guiding or teleoperation by a human expert. The demonstrated trajectories are played back to generate the desired motion. Recently, Calinon proposed a probabilistic framework to teach robots simple manipulation tasks [3]. Demonstrations were given by teleoperating a robot. Its motions in relation to the objects in the world were subsequently encoded in a Gaussian Mixture Model after reducing their dimensionality. By applying Gaussian Mixture Regression and optimization, the robot was able to reproduce the demonstrated movements in perturbed situations. In order to facilitate this generalization, several demonstrations of a movement need to be given. From these demonstrations the robot captures the essence of the task in terms of correlations between objects.

Reinforcement learning offers a way to reduce the number of required training examples. However, it is known to be prone to the *curse of dimensionality*. Early approaches were thus limited to low-dimensional and discrete state and action spaces [4]. More recently, strategies have been proposed that allow for continuous representations. For instance, so called *policy gradient* methods have been developed and successfully applied to optimize the gait of Sony AIBO robots [5]. These methods, however, discard most of the information contained in the training examples. To improve data-efficiency, Lizotte et al. [6] proposed to select actions probabilistically, based on Gaussian Process Regression and the most probable improvement criterion. We propose an improved search strategy that not only makes efficient use of available data, but also balances the probabilities for improvement and degradation.

To overcome the limitations of the approaches above, Schaal [7] was among the first who proposed to combine reinforcement learning and imitation learning. In his work, a robot was able to learn the classical task of pole balancing from a 30s demonstration in a single trial. In more recent work, Billard and Guenter [8] extended their imitation learning framework by a reinforcement learning module, in order to be able to handle unexpected changes in the environment. However, both approaches merely used the human demonstrations to initialize the reinforcement learning, thus reducing the size of the search space. In our approach, further demonstrations can be incorporated at any point in time.

3 Expected Deviation in Gaussian Processes

Central to our approach is the idea that the performance of movements can be measured as scalar reward and that this measure can be generalized across situations and actions. Thus, we form a combined state-action space and define a scalar continuous value function Q on it.

3.1 Gaussian Process Regression

We apply Gaussian Process Regression (GPR, [9]) to generalize value across the state-action space and to cope with the uncertainty involved in measuring reward and executing actions. The basic assumption underlying Gaussian Processes (GPs) is that for any finite set of points $X = \{x_i\}_{i=1}^{N}$ the function values $f(X)$ are jointly normal distributed. In GPR, observations y_i at points x_i are drawn from the noisy process

$$y_i = f(x_i) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma_0^2).$$

GPR allows to predict Gaussian estimates for any points x_* based on training examples $D := \{(x_i, y_i)\}_{i=1}^{N}$. We model similarity in a local context of the state-action space by means of the Radial Basis kernel function. In regions that are far away from training examples, large predicted variance indicates high uncertainty in the estimate.

3.2 Expected Deviation

In our approach, we make extensive use of predicted uncertainty: From mean and variance for a state-action pair we determine a measure of expected deviation from a given value level. This deviation can be defined as either the expected improvement or the expected degradation [10]. We use this measure to decide when an action is unsafe, or to find promising actions during optimization.

4 Motion Primitives

Humans intuitively recognize motions as looking human-like or being artificial. Yet it turns out to be surprisingly hard to describe exactly which characteristics are responsible for this judgement and to apply these insights to robot motions. Simple mathematical models usually are not sufficient and motions of contemporary robots often are described as being clumsy or jerky.
On the other hand, it is generally accepted that human motions are smooth. In particular, analyses of human reaching movements show that these usually do not follow a straight line but rather proceed along a curve at varying speed.
In order to evaluate our learning framework on human grasping movements, we developed a controller that is

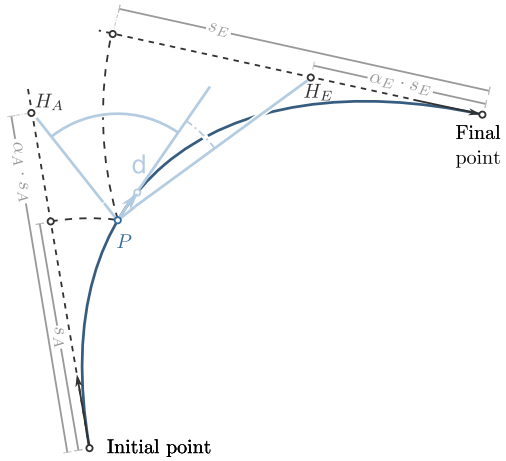

Figure 2: Generation of intermediate points along the desired trajectory. Dashed lines indicate the tangent lines at the initial and final points P_A and P_E. The distances s_a and s_E of the current location P towards P_A and P_E are scaled to obtain the auxiliary points H_A, H_E. The final approach direction d is a linear combination of the directions from P to the auxiliary points.

able to generate a wide range of anthropomorphic movements, which are determined by 31 parameters (see **Table 1**). This allows for a compact, low-dimensional representation of movements compared to the original trajectories. The controller generates a trajectory which contains poses of the end-effector at regular intervals, as well as the degree to which the hand is opened. Our controller is thus not limited to a particular robot arm or design.

4.1 Trajectory Generation

The first step in generating a movement is to determine a via point on the trajectory, in addition to the initial and final points. The trajectory is then split at this point to obtain two segments. For our purpose, the generation of grasping movements, the highest point of the trajectory has proven to be a suitable choice for the via point.
The movement generation process consecutively processes the trajectory's segments, each in the same fashion. Every segment is defined by its initial point P_A and its final point P_E. Starting from the initial point, our controller generates poses along the trajectory by successively moving towards a series of intermediate goal points. The generation of these intermediate points is illustrated in **Figure 2**.
First, the direction towards the next intermediate point is determined by considering the tangent lines on the trajectory, passing through the segment's initial and final points. These specify the desired direction of movement at these points and are the same for all intermediate points of a segment. To generate a smooth shift between these directions, an auxiliary point is chosen on each tangent line. Its position is determined by scaling the distance between the current position and each tangent's boundary point according to eq. (1):

$$\begin{aligned} H_A &= P_A + \alpha_A \cdot \|P_A - P\| \cdot d_A \\ H_E &= P_E - \alpha_E \cdot \|P_E - P\| \cdot d_E, \end{aligned} \tag{1}$$

where d_A and d_E represent the directions of the tangent lines and α_A and α_E are scaling factors. They are independent for every segment. The choice of the scaling factors determines how much the generated trajectory is bent. That way, our controller allows for movements reaching far back as well as for direct movements. The direction d towards the next intermediate point is formed by averaging the directions to the auxiliary points according to a weighting factor f:

$$d = f \cdot \frac{H_A - P}{\|H_A - P\|} + (1 - f) \cdot \frac{H_E - P}{\|H_E - P\|}. \tag{2}$$

The value of f starts near 1 and gradually shifts towards 0 as the movement generation progresses towards the final point of the segment. At a certain distance, which can be determined by another input parameter θ, the influence of the initial direction vanishes completely and the direction to the next intermediate point is only determined by the auxiliary point on the final tangent. The weighting factor

f is determined by

$$f = 1 - \frac{\|P_E - P\|}{\theta \cdot \|P_E - P_A\|}; \qquad 0 \leq f \leq 1. \tag{3}$$

The next pose P_* is generated by moving along the determined direction according to the current velocity v:

$$P_* = P + v \cdot \Delta t \cdot d. \tag{4}$$

This value initially is zero and increases towards the via point where it reaches its maximum value v_{max}, which also is an input parameter to our controller. From the via point, the velocity decreases towards the final point. The velocity, however, is not a direct product of our controller. It only becomes apparent implicitly through the distance of the generated poses.

To guarantee a smooth transition between the first and second segment of a movement, the final point and tangent of the first segment correspond to the first point and tangent of the second segment. By adding further via points and segments, arbitrary trajectories can be generated.

Parameter	Dimensionality
Initial point of the movement	3
Via point	3
Final point of the movement	3
Direction at the initial point	3
Direction at the via point	3
Direction at the final point	3
Opening degree of the hand	3
Accelerations	2
Limitation of the influence of the initial directions	2
Scaling factors on tangents	4
Offsets	2

Table 1: Parameters of our controller

4.2 Orientation and Opening of the Hand

Just like the position of the hand during a movement, its orientation is a characteristic feature of anthropomorphic movements. Our controller thus generates grasping trajectories that increasingly turn the hand towards the object to be grasped as the hand approaches the object. In the course of the first segment of a movement, the hand maintains a convenient pose, independent of the object to be grasped. Once the via point has been crossed, the hand is gradually turned towards the object to be grasped, by linearly interpolating its orientation.

Our controller also controls the degree to which the hand is opened according to three dedicated input parameters. At the beginning of the movement, the hand is closed. While it is moved to the via point, the hand is opened up to a degree that is determined by one of the three parameters. The two remaining parameters determine the maximum degree to which the hand will be opened and the distance to the target position, at which it should be attained. From this point on, the hand is closed until it grasps the object at the target position. This behaviour reflects the behaviour of humans when grasping objects. Humans start to close the hand before actually reaching the desired object.

5 Interactive Learning

In our learning framework, we implement imitation and reinforcement learning as two alternative control flows. When the robot encounters a new situation, GPR provides Gaussian estimates on the value of actions. Based on this knowledge, our algorithm selects the adequate form of learning: In the case of insufficient knowledge about promising actions, the robot acquires additional knowledge from human demonstration. Otherwise, it uses reinforcement learning to improve its actions.

5.1 Decision for Imitation or Reinforcement Learning

To make the decision for imitation or reinforcement learning, we have to determine the best known action in the current state s_{curr}. In a first step, we search for a training example $\hat{x} = (\hat{s}, \hat{a})$ that has high value and that is close to the current situation s_{curr} by minimizing

$$\hat{x} = \underset{(s,a)\in X}{\operatorname{argmin}} \; (1 - \mu(s,a)) + \|s_{curr} - s\|_2 \,. \tag{5}$$

Next, we use the action $\hat{a}$ from this training example $\hat{x}$ to initialize the search for the best known action a_{best} in the current state s_{curr}. For this purpose, we maximize

$$a_{\text{best}} = \underset{a}{\operatorname{argmax}} \; \mu(s_{curr}, a) - \sigma(s_{curr}, a) \tag{6}$$

through Rprop [11] gradient descent, which is faster than standard gradient descent and is less susceptible to end in local maxima. This optimization finds an action with large expected value but small uncertainty.

Finally, we decide for a learning method according to the expected degraded value [12] at the solution $x_{\text{best}} := (s_{\text{curr}}, a_{\text{best}})$ towards $Q_{\text{best}} := \mu(x_{\text{best}})$. If this indicator is below some threshold δ, there is no training example that is sufficiently similar to the current situation and our algorithm asks for a human demonstration. Otherwise, the training examples contain enough information to predict the outcome of actions and to safely use reinforcement learning, without risking damage to the robot by executing arbitrary movements.

5.2 Imitation Learning

For imitation learning, the user is asked to demonstrate the motion for the given situation. In doing so, his motions are recorded using a motion capture rig and a data glove, yielding trajectory data for the human's body parts. As our robot features a human-like joint configuration and upper-body proportions, the mapping from human to robot kinematics is straightforward.

After segmentation, we extract the demonstrated action from the recorded data in the form of a parameterized motion primitive that is suitable as input to our controller. For this step, we apply two different techniques for the two different categories of parameters of our controller.

Parameters corresponding to geometric or kinematic features, such as the location of the first and last points of the trajectory or the speed at a certain point, can be computed directly from the trajectory data using closed formulas that only depend on trajectory points. Out of the 31 parameters of our controller, the majority (23) belong to this group: the initial, via, and final points, the corresponding directions, and the parameters that determine the degree to which the hand is opened. These parameters also allow for an intuitive geometric interpretation.

The remaining eight parameters cannot be computed in closed form and thus need to be determined iteratively. In this work, we use the Downhill Simplex method by Nelder and Mead [13] to optimize a non-differentiable cost function and to determine the optimal parameter values. The cost function was designed to punish distance in space and time between the human's trajectory T_D and the robot's trajectory $T_R(\theta)$:

$$c(\theta) = \mathrm{dist}(T_D, T_R(\theta)) + \lambda \left(1 - \frac{t_D}{t_R(\theta)}\right)^2. \quad (7)$$

The variables t_D and $t_R(\theta)$ signify the duration of the movements and λ is a weighting factor. The distance measure dist computes the mean squared distance between the two trajectories by averaging a point-to-line metric which was inspired by [14].

Once all parameter values have been determined, the corresponding movement is generated by the controller and evaluated by the robot. The resulting reward is stored, along with the computed parameter values, as a new training example.

5.3 Reinforcement Learning

In reinforcement learning, we use Gaussian Process prediction to determine promising actions. To this end, we propose an explorative optimization strategy that safely maximizes value. We achieve this by trading off expected improvement and degradation. The chosen action along with the experienced reward eventually becomes a new training example that contributes to future predictions.

Expected improvement optimization [10] selects points that achieve highest expected improvement compared to the current best value. The expected improvement considers mean and variance of the GP posterior. Thus, the optimization strategy performs informed exploration which is based on the knowledge gathered so far. We also incorporate a lower bound on the expected degraded value into our optimization criterion. By considering the expected improvement as well as the expected degradation, our approach is able to produce an efficient search strategy that also protects the robot from executing unpredictable actions. Additionally, an adjustable lower bound on the admissible degraded value allows us to gradually focus the search on relevant parts of the search space with increasingly high performance.

6 Experiments

6.1 Experiment Setup

For our experiments, we used a robot with an anthropomorphic upper body scheme [15]. In particular, its two anthropomorphic arms have been designed to match the proportions and degrees of freedom (DoF) of their average human counterparts. From trunk to gripper each arm consists of a 3 DoF shoulder, a 1 DoF elbow, and a 3 DoF wrist joint. This endows the robot with a workspace that is similar to that of humans, which greatly facilitates imitation learning. Attached to the arm is a two-part gripper.

In our setup, the robot is situated at a fixed position in front of the table. The human teacher demonstrates grasping motions on a second table.

6.2 Task Description

We define the task of the robot as grasping an object as precisely as possible from arbitrary positions on the table. Thus, the state for our learning problem is the two-dimensional location of the object. The robot can measure the location with its laser range finder. After the robot performed the grasping movement, we score its performance by the displacement Δ (in meters) of the object to its initial location. Additionally, we penalize collisions, motions outside the workspace, and the missing of the object by the reward function

$$r(s,a) = \begin{cases} -1 & \text{collision or workspace failure,} \\ 0 & \text{object missed,} \\ 1 - \Delta & \text{otherwise.} \end{cases} \quad (8)$$

Obviously, this reward function does not meet the basic underlying assumption for GPs that any set of function values is jointly normal distributed. To still be able to apply our approach for this function, we apply GPR for each component and combine the predictions in a Gaussian mixture model.

Throughout our experiments we set the kernel widths for the object location to 0.2 for the successful and 0.02 for the unsuccessful cases. For the motion parameters we set the kernel widths to 0.02 and 0.002, respectively. We set the decision threshold on the expected degraded value of the known best action to $\delta = 0.8$.

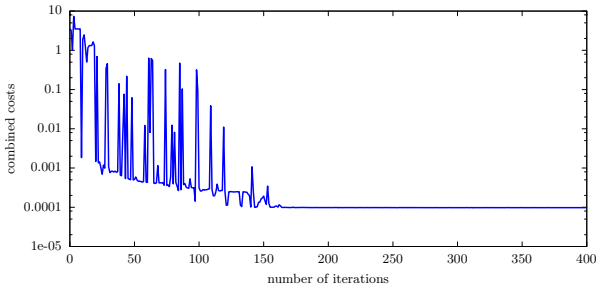

Figure 3: Evolution of the combined costs (7) during the optimization process, averaged over 30 example movements at different locations on the table.

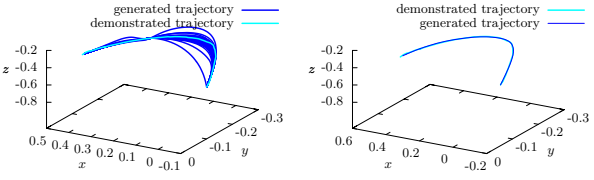

Figure 4: Left: Evaluated trajectories during optimization for a single movement. Right: Resulting trajectory.

6.3 Imitation Learning Experiments

In our approach, imitation learning is used to provide initial actions when not enough information on executable actions for the current location of the object are available. In such a situation, the robot asks for a demonstration. The human teacher places the object at a similar location in his workspace and demonstrates the grasping movement (cf. **Figure 1**). Due to the generalization properties of our approach, the objects do not have to be placed at exactly the same position. From the recorded trajectory, we extract the parameters of a motion primitive as action. As a measure of imitation quality, we used the similarity in space and time between the human demonstrations and the trajectories generated by our algorithm. Out of the 31 parameters that are required by our controller, 23 were determined directly from the demonstrated trajectory. The remaining 8 parameters were optimized iteratively.

Figure 3 shows how the combined costs for spatial and temporal dissimilarity decrease and finally converge to 10^{-4} after 150 iterations. These results were obtained by averaging the costs during the learning process for 30 demonstrated grasps at different locations on the table.

Figure 4 depicts how the reproduced trajectories approach the demonstration during optimization. The locations of the first, the last and the highest point of the trajectory are amongst the parameters that can be determined directly and are thus fixed during optimization. Our method is also robust against errors in the motion capture data and fills gaps smoothly in combination with our controller. In the end of the optimization process, the optimized trajectory is visually indistinguishable from the demonstration.

6.4 Reinforcement Learning Experiments

For the task at hand, our learning approach has to compensate for kinematic differences between the human teacher and the robot. In our experiments, we thus choose four parameters of the motion controller for optimization. These parameters determine offsets to the position of the grasp on the table and the closure of the hand along the trajectory. The other parameters of our controller describe the anthropomorphic characteristics of the movement and hence do not affect reward. They are solely determined by human demonstration. In simulation, the mapping between human and robot motion is close to perfect. We thus add an artificial x-offset of 15cm to the grasping position parameters.

6.4.1 Evaluation of the Optimization Strategy

In a first simulation experiment, we limit the optimization to the two offset parameters to visualize the learning strategy of our algorithm. We also keep the position of the object fixed during the experiment. First, we demonstrate a motion for the object location as a starting point for optimization. Then, we apply our reinforcement learning approach to reduce the error induced by the artificial offset.

6.4.2 Evaluation of the Optimization Strategy

Figure 5 (top) shows the evolution of reward in this experiment. Imitation learning already yields a grasping movement with a reward of 0.92. After about 30 reinforcement steps, the optimization converges to a mean reward of approx. 0.998 with standard deviation 0.0015, close to the optimal reward.

In **Figure 5 (bottom)**, we visualize the adjustments made to the offset parameters by our optimization strategy. Each circle represents an action chosen by our learning approach.

Filled blue circles indicate selected actions without random perturbation. Actions at unfilled red circles have been found after random reinitialization to escape potential local maxima. The optimization stategy proceeds in a goal-directed way towards the optimal parameters.

Note that after larger steps, the strategy often takes smaller steps into the direction of known points. This is due to the fact, that the expected improvement is still large in regions of the state space where GPR predicts high mean value and medium variance. As soon as the uncertainty in these regions shrinks, our strategy proceeds towards new promising regions. It is also interesting to note that close to the optimal parameters, the optimization exhibits a star-shaped exploration strategy.

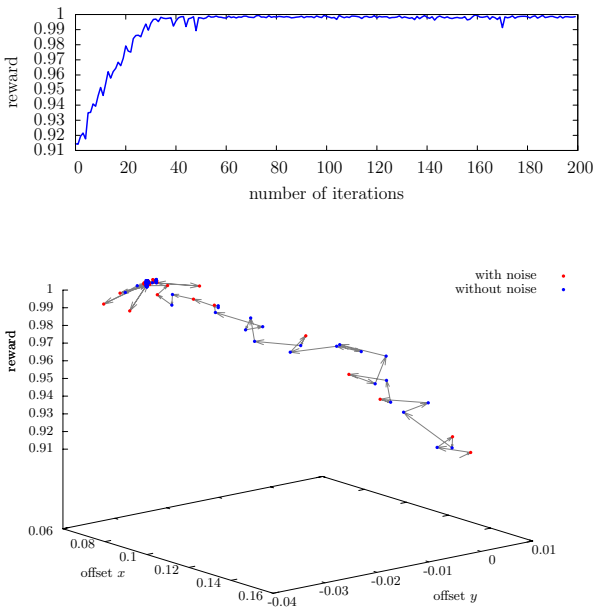

Figure 5: Evaluation of the optimization strategy for a fixed object position in simulation. Top: Reward of the executed motion during optimization. Bottom: Proceeding of the optimization strategy. Filled blue circles indicate selected actions without random perturbation. Actions at unfilled red circles have been found after random reinitialization.

6.4.3 Reinforcement Learning with 4 Parameters on the Real Robot

We repeated the experiment of the preceding section with the real robot, this time learning all 4 parameters. The position of the cup remained fixed. **Figure 6** depicts the evolution of reward over time. It shows that the initialisation through an imitation already achieved a reward of 0.96 in this experiment which was improved to 0.995 after 55 iterations. This corresponds to a displacement of 5 mm. In contrast to the experiments in the simulated enviroment, the reward does not increase further because of uncertainties in the robot kinematics which were not present in simulation.

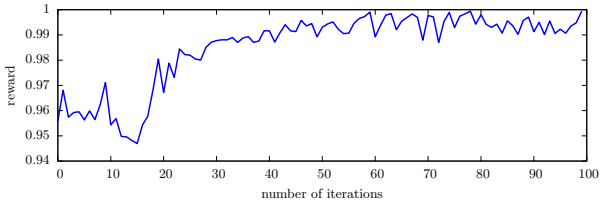

Figure 6: Learning curve of the experiment using the real robot. The object was placed at a fixed position 40cm in front of the robot and 20cm to its right.

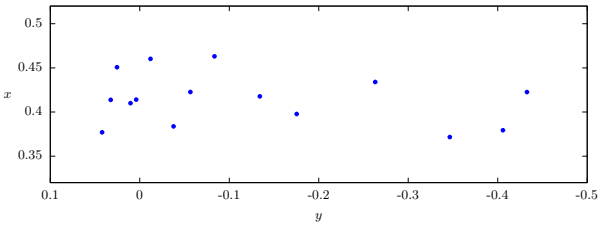

Figure 7: Top-down view of the robot's entire workspace. The blue dots represent positions of the object during our simulated experiment, where the robot asked for a demonstration.

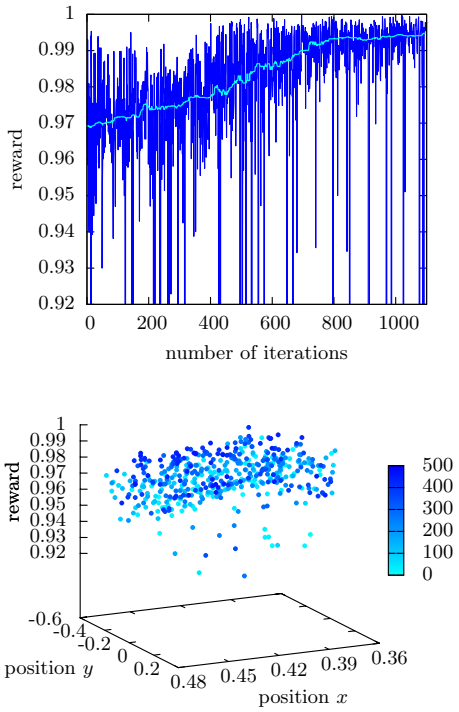

Figure 8: Top: Performance when learning grasping movements within a 600cm^2 area in simulation. Obtained reward is shown in dark blue/gray. For values below the plotted range the robot failed to grasp the object and received zero reward. The light blue/gray line shows the median reward within a window of 100 iterations. Bottom: Achieved reward for different object locations. The iteration count is encoded by the dots' color. For clarity, only the first 500 iterations are plotted and failures to grasp the object were omitted.

6.5 Interactive Learning Experiments

Finally, we evaluate our combined approach to reinforcement and imitation learning within the complete workspace of the robot in simulation. The area has a size of 10x60cm. The robot was asked to grasp a object in random positions. It then had to decide whether to try to grasp

the object based on its experience or to demand a demonstration. In the latter case, a human trainer put a cup at a similar place in his/her one workspace and demonstrated the desired movement. **Figure 7** shows a top down view of the robots workspace and the blue points indicate positions at which the robot asked for demonstrations. All of these 15 demonstrations were required during the first 42 iterations, 13 of them even within the first 24 iterations.

Figure 8 shows the corresponding evolution of reward. Within 600 trials it achieves high performance for random object locations. Considering the large workspace and the number of different grasping movements this is a persuasive result. The layered structure of the points is an indication of generalisation ability, as the reward increases simultaneously all over the workspace.

7 Conclusion

In this work, we presented a new approach to intuitively and efficiently teach robots anthropomorphic movement primitives. For this, we seamlessly integrate the well known paradigms of learning from demonstration and reinforcement in a single coherent framework. In contrast to previous approaches, that chained an imitation and a reinforcement learning phase, our method implements them as two alternative control paths. In every situation, the system decides which one to use to exploit each method's strengths while mitigating their shortcomings.

To facilitate this, our approach relies on Gaussian Process Regression (GPR) to generalize a measure of reward across the combined state-action-space. From this, we compute the expected deviation which is the basis upon which our system decides for imitation learning or reinforcement learning. Furthermore, we use it to select actions during reinforcement learning that trade off high reward versus predictable actions.

To evaluate our approach, we considered the task of grasping an object at an arbitrary position on a table. Our imitation learning algorithm was able to produce movements that are visually indistinguishable from the demonstrations. The reinforcement strategy produced movements with near-optimal reward in a goal-directed fashion, avoiding outlier movements that pose a threat to the robot. Our combined approach was able to teach the grasping task for arbitrary positions in a $600\mathrm{cm}^2$ workspace to our robot from only 15 demonstrations.

The advantages of our approach are manifold. First, the use of learning methods that are known from human teaching psychology renders our method intuitively operable and helps in making robot programming accessible to average users. Second, our unique combination of both methods allows for flexible learning that uses the most appropriate method in every situation. This way we achieve data-efficient learning and circumvent shortcomings of the individual methods.

References

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot Programming by Demonstration. In *Handbook of Robotics*. Springer, 2008.

[2] G. Biggs and B. Macdonald. A survey of robot programming systems. In *Proc. of the Australasian Conf. on Robotics and Automation, CSIRO*, 2003.

[3] S. Calinon, F. Guenter, and A. Billard. On Learning, Representing and Generalizing a Task in a Humanoid Robot. *IEEE Trans. on systems, man and cybernetics*, 2007.

[4] L. Pack Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *J. of artificial intelligence research*, 1996.

[5] N. Kohl and P. Stone. Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004.

[6] D. J. Lizotte, T. Wang, M. H. Bowling, and D. Schuurmans. Automatic Gait Optimization with Gaussian Process Regression. In *Proc. of the Int. J. Conf. on Artificial Intelligence*, 2007.

[7] S. Schaal. Learning From Demonstration. *Advances in neural information processing systems*, 1997.

[8] F. Guenter and A. Billard. Using Reinforcement Learning to Adapt an Imitation Task. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007.

[9] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[10] D. R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *J. of Global Optimization*, 2001.

[11] M. Riedmiller. Rprop-description and implementation details. Technical report, University of Karlsruhe, 1994.

[12] K. Gräve. Lernen von Greifbewegungen aus Imitation und eigener Erfahrung. Master's thesis, Universität Bonn, November 2009.

[13] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, January 1965.

[14] A. Censi. An ICP variant using a point-to-line metric. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2008.

[15] J. Stückler and S. Behnke. Integrating Indoor Mobility, Object Manipulation and Intuitive Interaction for Domestic Service Tasks. In *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots*, 2009.