In Proceedings of the 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids), Nashville, TN, December 2010

Improving Imitated Grasping Motions through Interactive Expected Deviation Learning

Kathrin Gräve, Jörg Stückler, and Sven Behnke

***Abstract*— One of the major obstacles that hinders the application of robots to human day-to-day tasks is the current lack of flexible learning methods to endow the robots with the necessary skills and to allow them to adapt to new situations. In this work, we present a new intuitive method for teaching a robot anthropomorphic motion primitives. Our method combines the advantages of reinforcement and imitation learning in a single coherent framework. In contrast to existing approaches that use human demonstrations merely as an initialization for reinforcement learning, our method treats both ways of learning as homologous modules and chooses the most appropriate one in every situation.**

We apply Gaussian Process Regression to generalize a measure of value across the combined state-action-space. Based on the expected value, uncertainty, and *expected deviation* of generalized movements, our method decides whether to ask for a human demonstration or to improve its performance on its own, using reinforcement learning. The latter employs a probabilistic search strategy, based on *expected deviation*, that greatly accelerates learning while protecting the robot from unpredictable movements at the same time. To evaluate the performance of our approach, we conducted a series of experiments and successfully trained a robot to grasp an object at arbitrary positions on a table.

I. Introduction

Today's industrial mass production would not be possible without the invention of robots that efficiently and precisely carry out repetitive manufacturing tasks. Just as manufacturing tasks, many of our everyday tasks are monotone, cumbersome or even dangerous. The development of autonomous service robots that one day might relieve humans from these kinds of tasks will thus attain significant importance in the future.

The requirements for service robots differ vastly from those of industrial applications. Manufacturing robots work in an isolated static environment where they fulfill specific tasks. Service robots, on the other hand, need to work in dynamic environments which are designed for human needs. They have to interact directly with humans and need to be able to perform many different, usually complex, tasks. Finally, they 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 possible situation it may ever encounter, the development of intuitive ways to teach a robot is of central importance in this research area. These would allow to continuously improve the robot's skills and to easily adapt them to new situations.

Many recent approaches (e.g.[1],[2]) to robot teaching have been inspired by methods that are known to be used by humans when teaching each other. In particular, these approaches offer the advantage to be intuitively accessible to humans which allows training to be carried out by laymen without specific expertise in programming or robotics. The most prominent examples of such human-inspired methods are *reinforcement learning* and *learning by imitation*.

Our method combines both paradigms of imitation and reinforcement seamlessly in a single coherent framework. It allows to intuitively teach a humanoid robot anthropomorphic motion primitives. We apply Gaussian Process Regression (GPR, [3]) to approximate a scalar value function over the combined state-action space. From the GP posterior distribution, we extract the best known action in a given situation. Based on *expected deviation* we also derive probabilistic indicators when to ask for demonstration or when to explore new actions. For reinforcement learning, we use GPs to determine actions that trade off *expected improvement* and *degradation*. By this, we combine the informed exploration of expected improvement optimization with a constraint on the predictability of the action's outcome.

In previous work, we detailed anthropomorphic motion primitives that capture and imitate grasping motions of a human teacher [4]. In this paper, we focus on our method to improve these skills by reinforcement. We present new results on the real robot. The advantages of our approach are twofold: Firstly, both ways of learning are known to be used by humans extensively and are thus intuitively exercisable by robot operators. Employing them for robot training makes robot programming accessible to non-expert users.

Secondly, the way in which we combine imitation and reinforcement learning allows us to make good use of each method's strengths to mitigate the other's shortcomings and to improve the overall learning quality. Acquiring anthropomorphic movements from human demonstrations offers a huge advantage over manual programming, as this kind of movements is usually complex and it is very hard to describe what makes a movement human-like. Reinforcement learning, on the other hand, reduces the number of demonstrations that are required by allowing the robot to learn on its own in many situations. At the same time, reinforcement learning can benefit from human demonstrations as these can be used as an initalization, thereby limiting the search space and greatly accelerating learning. Reinforcement learning can also be used to compensate differences between human and robot kinematics.

This paper is organized as follows: After discussing related work in Sec. II, we detail the methods of Gaussian Processes and expected deviation in Sec. III. Sec. IV contains a

All authors are with the Autonomous Intelligent Systems Group, University of Bonn, Germany. Email: graeve@ais.uni-bonn.de

brief description of the motion primitives used. In Sec. V, we describe our interactive learning approach and detail reinforcement learning using expected deviation. We evaluate our approach in experiments in Sec. VI.

II. Related Work

In the general reinforcement learning setting, an agent chooses actions in its current situation that maximize the long-term expected reward for following its policy [5]. Learning comprises to find a policy that achieves optimal long-term reward from either a model of the environment or from experience gathered by interaction with the environment. In contrast to supervised learning in which a teacher would directly provide the optimal action, the agent receives only evaluative feedback for its actions. This reward is typically given as a scalar signal.

As many methods, reinforcement learning is prone to the *curse of dimensionality*. Early approaches were thus limited to low-dimensional and discrete state and action spaces [6]. 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 [7]. One shortcoming of these methods is their data-inefficiency as they require lots of training examples but discard most of the information contained therein. Lizotte et al. [8] on the other hand, take a probabilistic approach employing Gaussian Process Regression that selects actions based on the most probable improvement criterion. Their approach makes efficient use of all available data due to the properties of Gaussian Processes, yet it does not take the possibility of degradation into account. We propose an improved search strategy that not only makes efficient use of available data, but also balances the probabilities for improvement and degradation by considering a measure that we call the *expected deviation*. This also helps to protect the robot from damages it might suffer from executing arbitrary movements.

Imitation learning methods comprise a set of supervised learning algorithms where the teacher not only gives the 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* [9].

Consequently, there are many approaches to apply imitation learning to robots [2]. Recently, Calinon proposed a probabilistic framework to teach robots simple manipulation tasks [10]. 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. In [11], this approach is extended to encode trajectories with continuous Hidden Markov Models. The authors apply an acceleration-based controller to generalize the trained model to similar situations.

To overcome the limitations of the approaches above, Schaal [12] 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 [13] 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. The seamless integration of both learning types in our framework is in contrast to existing approaches that non-interactively chain imitation and reinforcement learning. It allows both modules to complement each other.

III. 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.

A. Gaussian Process Regression

We apply Gaussian Process Regression (GPR, [3]) 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, i.e.

$$f(X) \sim \mathcal{N}(0, K),$$

where the elements of the covariance matrix K are determined by the kernel function $K_{nm} = k(x_n, x_m)$.

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 $D := \{(x_i, y_i)\}_{i=1}^N$:

$$\mu(x_*) = K_*^T C^{-1} y, \tag{1}$$

$$\sigma^2(x_*) = K_{**} - K_*^T C^{-1} K_*, \tag{2}$$

where $C = K + \sigma_0^2 I$ and $y := (y_1, \ldots, y_N)^T$. The matrices K_{**} and K_* contain the covariances between the query points x_*, and between x_* and the training points X, respectively.

We model similarity in a local context of the state-action space by means of the Radial Basis kernel function

$$k(x, x') = \theta \ \exp\left(-\frac{1}{2}(x - x')^T \Sigma^{-1}(x - x')\right) \quad (3)$$

with $\Sigma = \operatorname{diag}\left(\sigma_1^2, \ldots, \sigma_M^2\right)$, where $M := \dim(x)$ and θ is the vertical length scale. In regions that are far away from training examples, large predicted variance indicates high uncertainty in the estimate.

B. 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 [14]. We use this measure to decide, when an action is unsafe, or to find promising actions during optimization.

Let

$$\begin{aligned} D^{\oplus}(x, \overline{f}) &:= \max(f(x) - \overline{f}, 0) \text{ and} \\ D^{\ominus}(x, \overline{f}) &:= \max(\overline{f} - f(x), 0) \end{aligned} \quad (4)$$

be the improvement and degradation from $\overline{f}$, respectively.

If $f(x)$ is normal distributed with mean $\mu(x)$ and variance $\sigma^2(x)$, one can derive a closed-form solution for the expected deviation at x [15]:

$$\begin{aligned} \mathbf{E}D^{(\cdot)} &= \int_{-\infty}^{+\infty} D^{(\cdot)} \cdot p(D^{(\cdot)}) \, dD^{(\cdot)} \\ &= \sigma(x) \cdot \left[u^{(\cdot)} \cdot \Phi(u^{(\cdot)}) + \phi(u^{(\cdot)})\right] \end{aligned} \quad (5)$$

where $D^{(\cdot)}$ is either the improvement or the degradation from $\overline{f}$ at x, and $u^{(\cdot)}$ is defined as $u^{\oplus} := \frac{\mu(x) - \overline{f}}{\sigma(x)}$ and $u^{\ominus} := \frac{\overline{f} - \mu(x)}{\sigma(x)}$, respectively. The functions $\Phi(u)$ and $\phi(u)$ are the cumulative distribution function and the density of the standard normal distribution.

Given a function level $\overline{f}$, we define the expected improved and expected degraded function values at x towards this level as

$$\begin{aligned} \mu^{\oplus}(x, \overline{f}) &:= \overline{f} + \mathbf{E}D^{\oplus}(x, \overline{f}) \text{ and} \\ \mu^{\ominus}(x, \overline{f}) &:= \overline{f} - \mathbf{E}D^{\ominus}(x, \overline{f}), \end{aligned} \quad (6)$$

respectively.

IV. Low-dimensional Movement Representation

The efficiency of learning algorithms crucially depends on the dimensionality of the parameter space. In order to represent movements in a low-dimensional space in contrast to the high-dimensional trajectory space we developed a controller which is able to generate versatile arm movements from 31 input parameters. Special emphasis was put on the ability to reproduce anthropomorphic movements similar to those of human demonstrations. Our controller divides a reaching movement into two segments at a so-called via point. Among the 31 parameters are kinematic features of the trajectory such as the coordinates of the initial, via and final points in task space, as well as the desired directions at these points. The trajectory is generated by interpolating directions to auxiliary points on the tangent lines at each segment's start and end. The locations of the auxiliary points are determined by further parameters such as scaling factors that are applied to the distance from the respective tangent's boundary points. Finally, there are parameters that determine the shape of the generated trajectory, such as its curvature. We also added two offsets to the target position of the trajectory to compensate for kinematic differences between the robot and a human demonstrator. We refer the interested reader to [4] for further details on the parametrization of the controller.

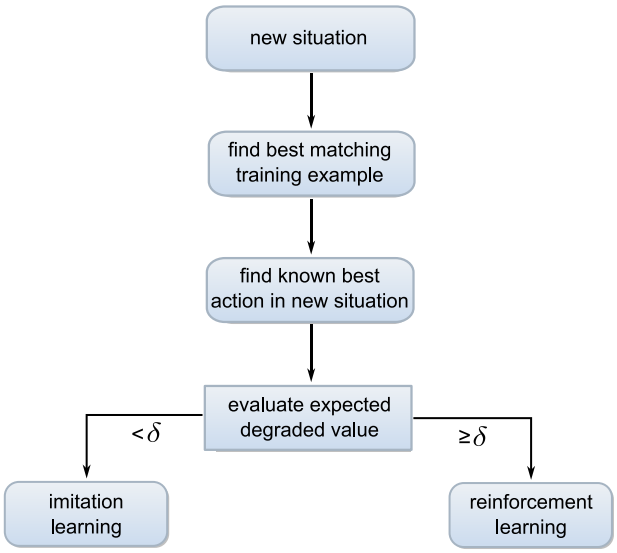

Fig. 1. When the robot encounters a new situation, we first determine a close already known state-action example with high value. We adapt the found action to the known best action in the new situation. Depending on the expected performance of this action either imitation or reinforcement learning is selected. For this decision we consider the expected degraded value of the action to rate its similarity to the known actions and the safety of its execution.

V. Interactive Expected Deviation Learning

In our learning framework, we implement imitation and reinforcement learning as two alternative control flows as depicted in Fig. 1. 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.

A. 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 is close to the current situation s_{curr} by minimizing

$$\hat{x} = \underset{(s,a) \in X}{\operatorname{argmin}} \ (Q_{max} - \mu(s, a)) + \alpha \left\| s_{\text{curr}} - s \right\|_2 \quad (7)$$

where Q_{max} is the optimal value and α trades between high value and similarity of the situation.

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

$$a_{\text{best}} = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ \mu(s_{\text{curr}}, a) - \sigma(s_{\text{curr}}, a) \tag{8}$$

through Rprop [16], [17] 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 $\mu^{\ominus}(x_{\text{best}}, Q_{\text{best}})$ at the solution $x_{\text{best}} := (s_{\text{curr}}, a_{\text{best}})$, where $Q_{\text{best}} := \mu(x_{\text{best}})$. 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.

B. Imitation Learning

For imitation learning, the user is asked to demonstrate the motion for the given situation. From the demonstrated trajectory, we extract an action in the form of a parametrized motion primitive. The robot evaluates the demonstrated action and stores the new training example.

Our parametrized motion controller as described in Sec. IV uses two types of parameters: Parameters corresponding to geometric or kinematic features, such as the location of the first and last points of the trajectory and the direction at these points, can be directly extracted from demonstrated trajectories. Other parameters that determine the trajectory at intermediate points can only be found iteratively. We optimize for similarity in duration and shape to fit the motion controller parameters into the demonstrated trajectory. Our objective function is a weighted sum of a point to line metric, measuring the spatial distance between the generated and demonstrated trajectories, and the squared difference in duration. Due to the non-differentiability of this function, we employ Nelder and Mead's Downhill Simplex Method [18] to find good parameter values.

The details of our imitation learning approach and the way the individual parameters are determined are fully described in [4].

C. Expected Deviation Learning

In reinforcement learning (cf. Fig. 2), 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 for the Gaussian Process that contributes to future predictions.

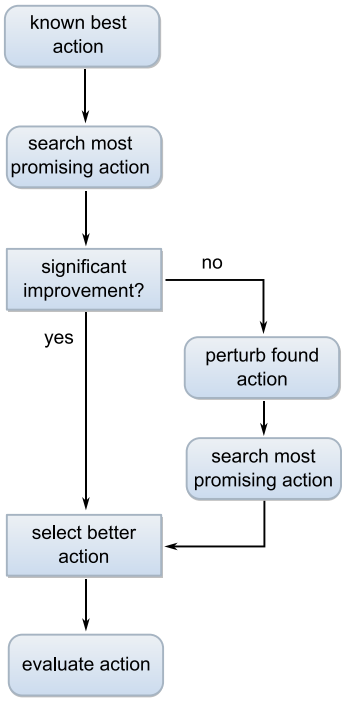

Fig. 2. Expected deviation learning first searches for the most promising action in the current situation. We optimize a criterion that finds a trade-off between expected improvement and expected degradation to the initial action. By this, we combine the informed exploration of expected improvement optimization with a constraint on the predictability of the action's outcome. We also detect when the optimization may end in local maxima. In this case, we randomly perturb the found action, and locally search for the most promising action again. Our approach finally selects the more promising action for evaluation.

Reinforcement learning, in contrast to imitation learning, is not a supervised learning algorithm. The robot is not provided with the desired solutions at training points but only gets a reward, indicating its performance. The robot has to come up with possible solutions on its own. Depending on the problem, the reward may be given by a human teacher or may be computed by the robot by evaluating the effects of its actions on the environment using its sensors.

In order to limit the costs for finding a good solution, it is crucial to intelligently choose the next action to evaluate. The approach presented here uses Gaussian Process Regression to generalize rewards from known situations and actions. It then uses expected deviation to define a function over actions for optimization. As this function is differentiable, extrema of this function can easily be found using Rprop gradient descent.

Expected improvement optimization [14] 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 a very efficient search strategy that also protects the

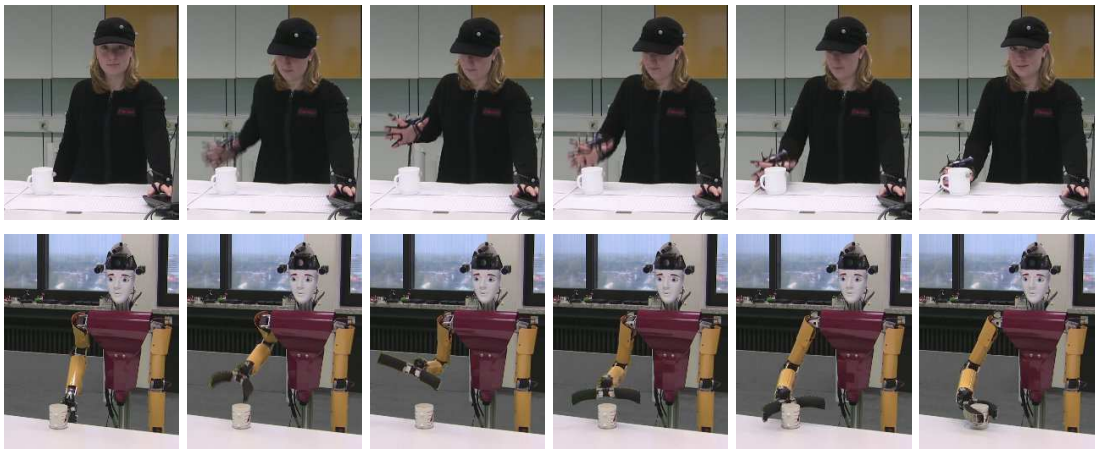

Fig. 3. The teacher demonstrates a grasping movement (top). From the recorded trajectory, our approach extracts a parameterized motion primitive. The robot imitates the grasp by executing the extracted motion (bottom).

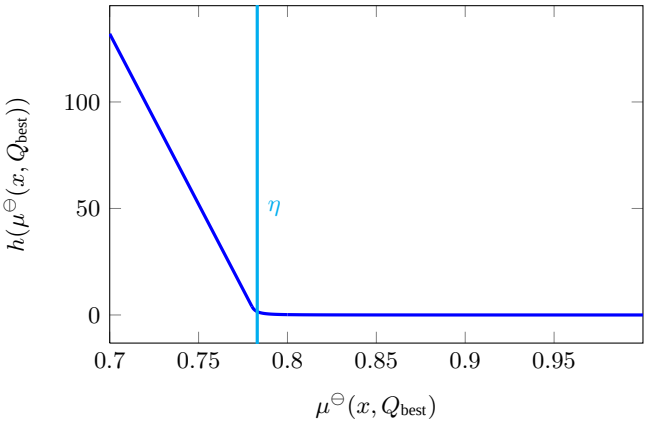

Fig. 4. The continuous function h thresholds the expected degraded value towards Q_{best} at η to prevent the execution of actions with unpredictable outcome.

robot from executing actions with unpredictable outcome. 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.

1) Optimizing Expected Deviation: To find the next action, we search for a maximum on the surface defined by

$$g(x) := \alpha\, \mathbf{E}D^{\oplus}(x, Q_{\text{best}}) - h(\underbrace{Q_{\text{best}} - \mathbf{E}D^{\ominus}(x, Q_{\text{best}})}_{=\,\mu^{\ominus}(x, Q_{\text{best}})}), \quad (9)$$

where α is a weighting factor and h is a differentiable thresholding function (cf. Fig. 4) that is approximately 0 for $\mu^{\ominus}(x, Q_{\text{best}}) \geq \eta$ and grows linearly with a large slope when $\mu^{\ominus}(x, Q_{\text{best}}) < \eta$.

The influences of expected improvement and expected degradation depend on the expected degraded value $\mu^{\ominus}(x, Q_{\text{best}})$ in x to the best expected value Q_{best} so far. As long as this difference is greater than a safety threshold η, we do not expect actions to be dangerous for the robot. In this case, the influence of the expected degradation is negligible and the value of the surface function is almost exclusively determined by the expected improvement. However, as $\mu^{\ominus}(x, Q_{\text{best}})$ approaches η, this influence increases. Once $\mu^{\ominus}(x, Q_{\text{best}})$ gets smaller than η, the value of the surface function is predominantly determined by the expected degradation, leading to a value that prohibits the choice of the corresponding action. The protection of the robot thus takes precedence over the choice of actions that promise a great improvement.

2) Enforcing Goal-Directedness: In order to further improve the search efficiency, we adapt the threshold η to Q_{best} such that it increases with the performance of the known actions. This makes the search more goal-directed. By enforcing a lower bound on η, the execution of dangerous actions is prevented. An upper bound on η avoids getting stuck in local extrema.

3) Escaping Local Maxima: Still, Rprop descent on the surface (9) may result in local maxima. To circumvent this, we detect when the goal-directed optimization yields only a marginal improvement to the starting action. In this case, we conduct a second search for a new locally optimal action according to Eq. (9). We initialize the search by randomly perturbing the found action with a zero-mean Gaussian noise term. The variance of this term depends on Q_{best}, similar to η.

Finally, the better action is evaluated and the training example is incorporated into the GP posterior.

VI. Experiments

We validate our approach in a series of experiments. As an exemplary task, we choose the task of grasping an object at an arbitrary position on a table, which is relevant to many real-world applications for service robots. The experiments are carried out on our robot to demonstrate the validity of our approach in the real setting. We also evaluate our approach in a simulated environment.

A. Experiment Setup

For our experiments, we used a robot with an anthropomorphic upper body scheme [19]. 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 whose parts are independently actuated by another two motors. The robot can carry objects up to a weight of 1kg.

In the trunk, the robot is equipped with a Hokuyo URG-04LX laser range finder which we use to localize objects on the table. A linear actuator in the trunk can lift the entire upper body by approximately 1m. By this, the robot can adjust its upper body to the height of the table and apply the learned grasping motions on varying heights.

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. We record the motions of the teacher with an optical motion capture rig and a data glove.

B. 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 perceive the location with its laser range finder. After the robot performed the grasping movement, we measure 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

$$r(s,a) = \begin{cases} -1 & \text{collision or workspace failure,} \\ 0 & \text{object missed,} \\ 1-\Delta & \text{otherwise.} \end{cases} \tag{10}$$

In reinforcement learning, the Q value estimates the expected long-term reward for state-action pairs. Since our task takes only one state-action pair to finish, Q is equal to r.

Rather than approximating the reward function with a single fixed kernel width, we apply GPR for each component and combine the predictions in a Gaussian mixture model. This allows us to assign different weights to the components by choosing individual kernel widths. The state-action space consists of the parameters of the movement controller and the 2D-position of the cup. 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. By choosing narrower kernel widths for them, the unsuccessful cases are given a more local influence. We set the decision threshold on the expected degraded value of the known best action to $\delta = 0.8$.

C. 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. Due to the generalization properties of the GP, 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. Finally, the robot executes the new action. Fig. 3 exemplarily shows the robot imitating a demonstrated motion.

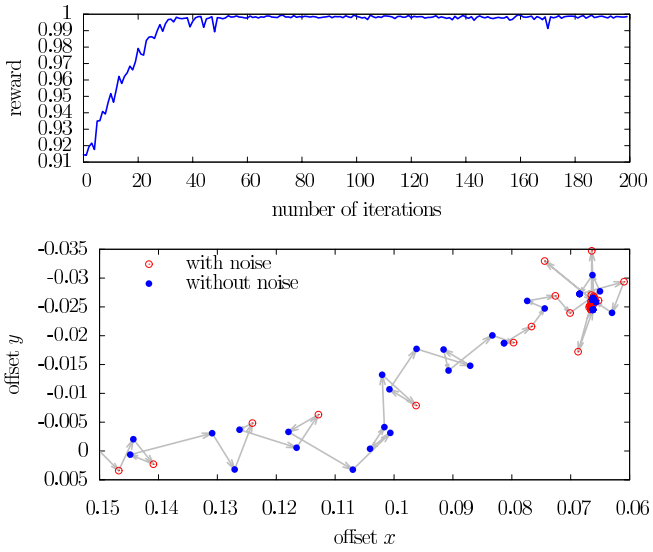

Fig. 5. Evaluation of the optimization strategy for a fixed object position in simulation. Top: Reward of the executed motion during optimization. Bottom: Visualization of the optimized position offset. Filled blue circles indicate selected actions without random perturbation. Actions at unfilled red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

D. Expected Deviation Learning Experiments

We now describe experiments to validate our reinforcement learning approach. The goal of reinforcement learning is to improve the movements learned from demonstration and to adapt them to similar situations.

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 offset of 15cm to the grasping position parameters.

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.

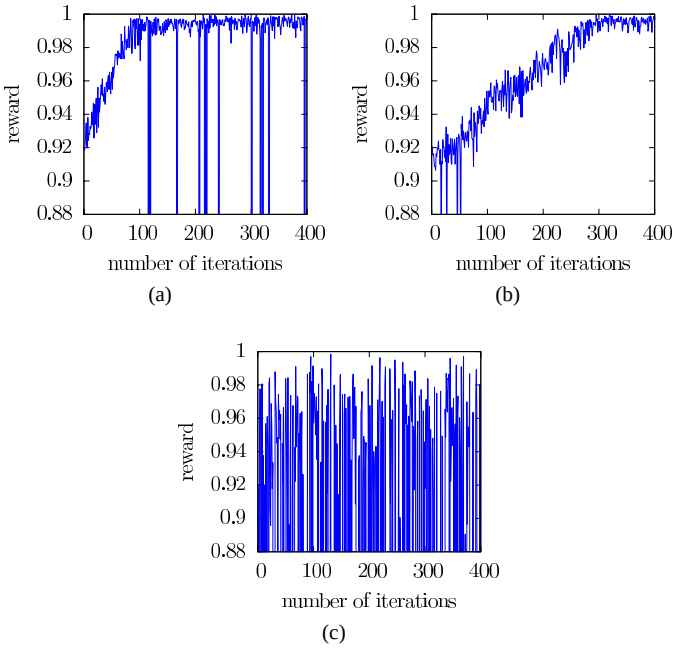

Fig. 6. Comparison of (a) expected deviation learning with random search strategies with (b) small Gaussian noise $\sigma = 0.005$, (c) large noise $\sigma = 0.04$. Random search strategies simply add noise to the known best action in the current situation.

Fig. 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 Fig. 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 (cf. Fig. 5).

2) Comparison to other Strategies: To evaluate the efficiency of our learning approach, we compare our optimization strategy with different random search strategies. Again, we perform the experiments in simulation. The object to be grasped is randomly placed within an area of $5cm \times 10cm$ at the demonstrated location.

In each of the random strategies, we simply add noise to the known best action a_{best} in the current situation which is also used as starting point for expected deviation learning.

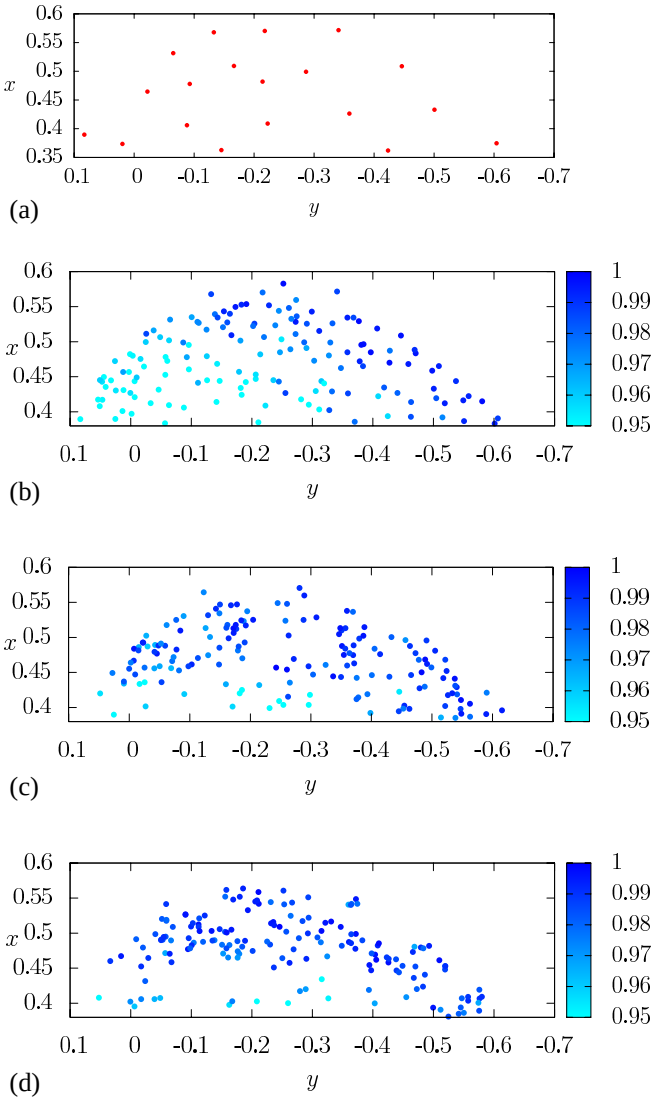

Fig. 7. Interactive expected deviation learning in the real setting. (a) Locations of 19 demonstrations required to teach the robot. (b)-(d) Reward and object location of first, intermediate, and final 166 actions, respectively (cf. Fig. 8).

Both strategies add constant normal distributed noise with standard deviations $\sigma = 0.005$ and $\sigma = 0.04$.

Fig. 6 shows results of expected deviation learning and the random search strategies. With small variance (Fig. 6(b)) the search reaches a similar level like expected deviation learning (Fig. 6(a)). However, random search requires almost twice as much iterations. While the number of required iterations can be reduced with larger variance (Fig. 6(c)), this search strategy does not converge. Even worse, this strategy frequently fails to grasp the object.

E. Interactive Expected Deviation Learning Experiments

Finally, we evaluate our combined approach to reinforcement and imitation learning on our real robot. In this experiment, we placed the object within the complete workspace of the robot.

During the experiment, the teacher places the object to

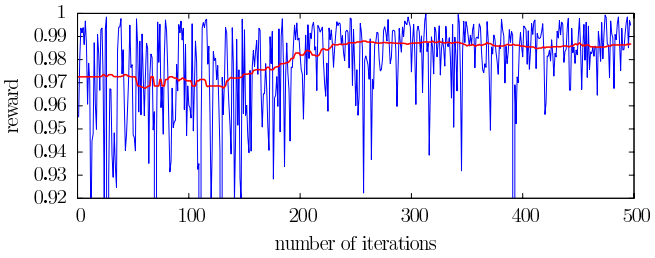

Fig. 8. Reward (blue) and its local median over 100 evaluations (red) during the experiment on interactive expected deviation learning in the real setting.

cover the workspace. Fig. 7(a) shows the object locations relative to the robot, at which the robot asks for demonstration. Only 19 demonstrations within the first 29 trials suffice to teach the robot grasping movements on the table.

When it does not ask for demonstrations, the robot optimizes the demonstrated motions to increase its grasping precision as can be seen in Fig. 7(b) to Fig. 7(d). After only about 500 iterations, the robot generalized and improved its skills to a good performance throughout its workspace. Fig. 8 shows the rewards obtained during the experiment.

VII. Conclusion

In this paper, we present *Interactive Expected Deviation Learning*, an approach that combines both paradigms of imitation and reinforcement seamlessly in a single coherent framework. We improve imitations of grasping motions of a human teacher through reinforcement learning. In each situation, our method decides which learning method to use.

In our approach, we assume that reward can be generalized across actions and situations. Further properties of the value function allow us to predict normal distributed estimates for any action and situation by the means of Gaussian Process Regression.

From these predictions we determine the *expected deviation* of actions towards value baselines. We use this measure to judge if the outcome of actions is sufficiently predictable. In the reinforcement learning setting, we also apply this measure to find new promising but safe actions based on the knowledge from previous experience. We designed our optimization strategy to trade off exploitation, exploration, and safety constraints.

We evaluated our approach for the task of grasping an object from an arbitrary position on a table. We demonstrated that learning of this task is possible from only few demonstrations. Reinforcement learning adapts the imitated motion primitives to the specifics of the robot. We compare our optimization strategy with other strategies and demonstrate superior performance of our approach.

The advantages of our approach are twofold: Firstly, both ways of learning are known to be used by humans extensively and are thus intuitively exercisable by robot operators. Secondly, the way in which we combine imitation and reinforcement learning allows us to make good use of each method's strengths to mitigate the other's shortcomings and to improve the overall learning quality.

Acknowledgment

This work was supported in part by NRW State within the B-IT Research School and the German Research Foundation (DFG) under grant BE 2556/2.

References

[1] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[2] Aude Billard, Sylvain Calinon, Rüdiger Dillmann, and Stefan Schaal. Robot Programming by Demonstration. In Bruno Siciliano and Oussama Khatib, editors, *Handbook of Robotics*, pages 1371–1394. Apringer, 2008.

[3] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[4] K. Gräve, J. Stückler, and S. Behnke. Learning motion skills from expert demonstrations and own experience using gaussian process regression. In *Proc. of the Joint Conference of the 41st Int. Symp. on Robotics (ISR 2010) and the 6th German Conf. on Robotics (ROBOTIK 2010)*, 2010.

[5] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, USA, 1998.

[6] Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[7] Nate Kohl and Peter Stone. Policy Gradient Reinforcement Learning for Fast Quadrupedal Locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2619–2624, May 2004.

[8] Daniel J. Lizotte, Tao Wang, Michael H. Bowling, and Dale Schuurmans. Automatic Gait Optimization with Gaussian Process Regression. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 944–949, 2007.

[9] K. Dautenhahn and C.L. Nehaniv. *Imitation in animals and artifacts*. MIT Press Cambridge, MA, USA, 2002.

[10] Sylvain Calinon, Florent Guenter, and Aude Billard. On Learning, Representing and Generalizing a Task in a Humanoid Robot. *IEEE transactions on systems, man and cybernetics, Part B. Special issue on robot learning by observation, demonstration and imitation*, 37(2):286–298, 2007.

[11] Sylvain Calinon, Florent D'halluin, Darwin Caldwell, and Aude Billard. Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework. pages 582 –588, 2009.

[12] Stefan Schaal. Learning From Demonstration. *Advances in neural information processing systems*, 9:1040–1046, 1997.

[13] Florent Guenter and Aude G. Billard. Using Reinforcement Learning to Adapt an Imitation Task. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1022–1027, 2007.

[14] Donald R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21(4):345–383, December 2001.

[15] Phillip Boyle. *Gaussian Processes for Regression and Optimisation*. PhD thesis, Victoria University of Wellington, 2007.

[16] Martin Riedmiller and Heinrich Braun. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP algorithm. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 586–591, San Francisco, CA, 1993.

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

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

[19] Jörg Stückler and Sven Behnke. Integrating Indoor Mobility, Object Manipulation and Intuitive Interaction for Domestic Service Tasks. In *Proceedings of the 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2009.