Learning to Grasp under Uncertainty

Freek Stulp, Evangelos Theodorou, Jonas Buchli, Stefan Schaal

***Abstract*— We present an approach that enables robots to learn motion primitives that are robust towards state estimation uncertainties. During reaching and preshaping, the robot learns to use fine manipulation strategies to maneuver the object into a pose at which closing the hand to perform the grasp is more likely to succeed. In contrast, common assumptions in grasp planning and motion planning for reaching are that these tasks can be performed independently, and that the robot has perfect knowledge of the pose of the objects in the environment.**

We implement our approach using Dynamic Movement Primitives and the probabilistic model-free reinforcement learning algorithm Policy Improvement with Path Integrals (PI^2). The cost function that PI^2 optimizes is a simple boolean that penalizes failed grasps. The key to acquiring robust motion primitives is to sample the actual pose of the object from a distribution that represents the state estimation uncertainty. During learning, the robot will thus optimize the chance of grasping an object from this distribution, rather than at one specific pose.

In our empirical evaluation, we demonstrate how the motion primitives become more robust when grasping simple cylindrical objects, as well as more complex, non-convex objects. We also investigate how well the learned motion primitives generalize towards new object positions and other state estimation uncertainty distributions.

I. Introduction

In force-closure analysis and motion planning for grasping, the object and its pose are often assumed to be known accurately [18], [3]. However, as Zheng and Qian [18] argue, "friction uncertainty and contact position uncertainty may have a disastrous effect on the closure properties of grasps.". Effective grasps also depend on the kinematic structure of the hand, and the movement of the hand towards the grasp, i.e. hand preshaping [1]. That theoretically successful grasps and motion plans for reaching need not necessarily be successful in practice has been demonstrated empirically [13], [3]. The aim of this paper is to address these issues by model-free learning of motion primitives for integrated reaching, preshaping and grasping, and which have *intrinsic robustness* towards state estimation uncertainty.

That this is feasible in principle has been shown in recent psychophysics experiments. Christopoulos and Schrater [5] demonstrate that humans adapt their grasp trajectories to directional uncertainty in the position of the object to be grasped. It is also shown that these adaptations (changes in approach direction and maximum grip aperture) lead to significantly better force-closure performance. As depicted in Fig. 1, the problem humans face here is to execute a movement that is likely to be successful for *all* of the possible positions the cylinder might have.

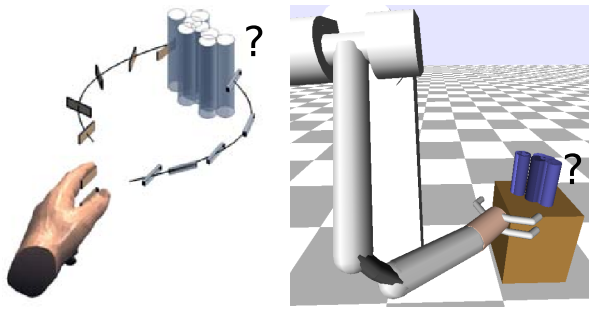

Fig. 1. Humans are able to generate grasping trajectories that are adapted to the direction of the state estimation uncertainty, which leads to better force-closure performance [5] (image reproduced with permission of the authors). The aim in this paper is to achieve similar adaptations in robots through reinforcement learning.

In this paper, we use a similar experimental methodology to that proposed in [5], to acquire movement primitives for a robotic manipulator. These primitives are implemented as Dynamic Movement Primitives (DMPs) [9]. First, a DMP is initialized with a preshape and grasp posture as determined by the open-source grasp planner GraspIt! [12]. This DMP is then optimized for successful grasping with model-free probabilistic reinforcement learning, where the cost function is a boolean that specifies whether the grasp was successful or not.

The key to acquiring DMPs that are robust to state estimation uncertainty is to randomly sample the actual object pose from a probability distribution that represents a model of the uncertainty in the object's pose. Christopoulos and Schrater call this *environmentally induced position uncertainty* [5]. The robot is thus not learning to grasp the object at one position, but rather optimizes the probability of grasping the object at any of the possible positions given by the distribution.

This approach yields a very challenging reinforcement learning (RL) problem, because: 1) the problem space is continuous and high-dimensional; 2) the terminal reward is boolean, and hence not very informative; 3) the actual and observed location vary randomly during learning, so the same grasp might sometimes succeed and sometimes fail, which leads to a noisy reward signal; 4) the reaching trajectory and actual grasp are not independent, and must be optimized simultaneously. Only recently have RL methods scaled up to

Computational Learning and Motor Control Lab, University of Southern California, Los Angeles, CA 90089. Contact: stulp@clmc.usc.edu

This research was supported in part by National Science Foundation grants ECS-0325383, IIS-0312802, IIS-0082995, IIS-9988642, ECS-0326095, ANI-0224419, the DARPA program on Learning Locomotion, the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637), and the ATR Computational Neuroscience Laboratories. F.S. was supported by a Research Fellowship from the German Research Foundation (DFG). J.B. was supported by an advanced researcher fellowship from the Swiss National Science Foundation. E.T. was supported by a Myronis Fellowship.

tasks of this complexity, and we show that our state-of-the-art reinforcement learning algorithm (Policy Improvement with Path Integrals – PI^2 [17]) is able to learn robust grasp trajectories in a reasonable number of exploration trials.

The resulting motion primitive does not explicitly reason about state estimation uncertainties on-line; these uncertainties are rather compiled implicitly into the motion primitive during learning. Therefore we call such motion primitives *intrinsically robust* to state estimation uncertainty. That this robustness indeed leads to better performance, the ultimate goal of this work, is verified in an empirical evaluation.

The rest of this paper is structured as follows. In the next section, we discuss related work. After summarizing our reinforcement learning algorithm PI^2 in Section III, we demonstrate how it is used to learn intrinsically robust motion primitives in Section IV. After presenting the empirical evaluation in Section V, we conclude with Section VI.

II. Related Work

In the experiments by Christopoulos and Schrater [5], human subjects were required to grasp a cylindrical object. Successful grasping is defined as being able to lift the object. The cylinder is visible initially, but occluded during the actual reaching movement. During occlusion, the cylinder is moved (with a robot) to a random position sampled from a 2D Gaussian distribution. This is the environmentally induced position uncertainty. Before reaching, the cylinder is moved to 5 random positions from the same distributions, allowing subjects to build a model of the distribution. The variation along the major axis of the covariance matrix is almost 50 times that of the minor axis, and the direction of the major axis is rotated over different trials. It is shown that humans adapt to the state estimation uncertainty by changing their angle of approach, and increasing their grip aperture. On average, this leads to better force-closure of the grip. The approach presented in this paper is similar to the experimental protocol of Christopoulos and Schrater [5], but here the learner is a robot rather than a human. For the robot, we see very similar adaptations, which increase the chance of a successful grasp.

In *compliant* or *fine* motion planning [11], mechanical compliance, .i.e. contact between the manipulator and object, is used to reduce uncertainty. Using this approach, it has been shown that a parts feeder with a parallel-jaw gripper is able to manipulate polygonal objects into a specific orientation *without any sensing* [6]. Lozano et al. [11] introduced the concept of a *preimage*, a region in configuration space in which a certain motion command guarantees that goal will be achieved. Using *preimage backchaining*, the same guarantees can be made for a sequence of commands, even under pose and action uncertainty. However, subsequent work demonstrated that this approach has prohibitive computational cost [4]. In contrast to fine motion planning, our approach 1) is model-free; 2) has negligible computational cost on-line; 3) uses one motion primitive instead of a sequence of actions; 4) considers manipulators with higher kinematic complexity. In essence, fine motion planning is not an explicit goal of our approach; the robot simply learns that fine motion planning is required to solve the task.

Planning with Partially Observable Markov Decision Processes (POMDPs) is an example of reasoning on-line about the effects of uncertainty in state on the outcome of an action. POMDPs have been used to determine when and which exploratory actions are required to reduce the uncertainty about the object and its pose to levels that allow for grasping [8]. Note that our methods do not preclude exploratory actions, or reasoning about state estimation uncertainty during task execution. As we shall see in Section V-D, failure rates after learning might still be too high if the level of state estimation uncertainty is high, simply because no trajectory that grasps all position in the distribution exist. In these cases, a robot would be wiser to perform an exploratory action to reduce uncertainty, rather than execute a motion primitive with a 20% chance of failing. But whether used in an active perception or high-level planning context, we believe it is always preferable to use a motion primitive that has been trained to deal with state estimation uncertainty, rather than one that has not.

Planning for manipulation tasks commonly assumes that the robot has perfect knowledge of the geometry and pose of the objects [3]. However, theoretically successful grasps and motion plans must not necessarily be successful in practice, as has been demonstrated empirically [13], [3]. Berenson et al. address these problems by generating a motion plan that is consistent with *all* object pose hypothesis [3]. This has very high computational cost, and requires an accurate model. Our approach rather replays from memory a motion that has been optimized off-line w.r.t. state estimation uncertainty.

III. Policy Improvement with Path Integrals - PI^2

Dynamic Movement Primitives (DMPs) are a flexible representation for motion primitives [9], which consist of a set of dynamic system equations that generate goal-directed movements. In the context of this paper, two DMP parameters are important: 1) the weights $\boldsymbol{\theta}$, which determine the shape of the movement, e.g. the end-effector trajectory $[\ddot{q}_t, \dot{q}_t, q_t]$ over time; 2) the goal g, which determines the final destination of the movement $q_{t=t_{final}}$, e.g. the position of an end-effector at the end of the movement.

In this paper, the aim is to determine the parameters $\boldsymbol{\theta}$ that lead to end-effector and hand posture movements that are most robust to state estimation uncertainty. To do, so we use PI^2 , our model-free probabilistic reinforcement learning algorithm, which is derived from first principles of stochastic optimal control [17]. PI^2 optimizes the DMP parameters $\boldsymbol{\theta}$ w.r.t. a cost function. As depicted in Fig. 2, it does so by executing a DMP K times, each time with slightly different parameters $\boldsymbol{\theta}+\epsilon$, where ϵ is Gaussian noise which is added to explore the parameter space. These 'noisy' DMP parameters generate slightly different movements $[\ddot{q}_t^k, \dot{q}_t^k, q_t^k]$, which each lead to different costs $\mathbf{S_t^k}$. Given the costs and noisy parameters of the K DMP executions, called *roll-outs*, PI^2 then updates the parameter vector $\boldsymbol{\theta}$ such that it is expected to generate movements that lead to lower costs in the future. After an update, the DMP is evaluated by executing it with

the new $\boldsymbol{\theta}^{new}$, without noise, i.e. $\epsilon = 0$. The process then continues with the new $\boldsymbol{\theta}$ as the basis for exploration. This generic loop is similar to other reinforcement learning algorithms [14]. PI^2 achieves its superior performance in the parameter update step, which is derived in full in [17].

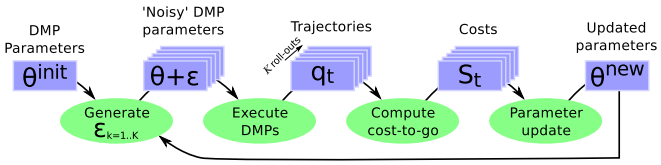

Fig. 2. Overview of the PI^2 algorithm.

Detailed descriptions of DMPs and PI^2 are found in [9] and [17] respectively. Open-source implementations of DMPs and PI^2 are available at:

`http://www-clmc.usc.edu/Resources/Software`

IV. Acquiring Intrinsically Robust Motion Primitives with (Environmentally Induced) Position Uncertainty

In this section, we describe how PI^2 is used to learn intrinsically robust motion primitives.

Initialization of the DMP. First, we determine the best grasp and preshape posture of the hand with GraspIt!, an open-source grasp planner that searches for good grasps by optimizing quality measures based on grasp wrench space [12]. Then, we generate a minimum-jerk trajectory that consists of two parts. *1) Reaching:* Move the end-effector from its initial pose to the preshape pose, which is a 6D trajectory representing position and orientation of the end-effector. During this end-effector movement, the hand posture is the preshape posture determined by GraspIt! *2) Grasping:* Move the hand posture from the preshape to the final grasp, represented as a n-dimensional minimum-jerk trajectory in the n-dimensional joint space of the hand. In this second part, the end-effector pose is kept constant. Finally, a 6+n-dimensional DMP is trained with this trajectory, as described in [9].

In simulation, provide the probability distribution for the state estimation uncertainty in the object pose. State-of-the-art sensors for robots, such as cameras, laser-scanners, and sonar sensors, are continuously improving in terms of fidelity and cost. However, these sensors are never perfect, and the algorithms that operate on them are only able to localize objects within some hypothetical probability distribution of pose estimates. When applied in simulation, the approach we present here *mimics* these imperfections by sampling object poses from a model of the probability distribution for a given robot. We thus *induce* position uncertainty. In principle, any (multi-modal) distribution can be used, but in this paper we use a 2D Gaussian, as in [5]. We assume that the z-coordinate of the object is fixed, as it stands on a table of known height.

When learning on a real robot, there is no need to specify a distribution, as position uncertainty must not be induced, but arises 'for free' from the robot's state estimation. Note that the learning algorithm does not require a model of the state estimation uncertainty. In fact, it is not even aware that state estimation uncertainty exists at all. It simply performs exploratory manipulation actions, observes the result, and adapts future behavior to optimize the cost function. So on the real robot, instead of adapting to a model of the robot's uncertainty, PI^2 adapts the reaching and grasping trajectories to the *actual* state estimation uncertainty, and the stochasticity in the motor system.

Reinforcement Learning. During learning, the robot always assumes the object is at the mode of the probability distribution representing the position of the object. But the actual positions of the objects with which the movement is executed are sampled randomly from the distribution for each of the K roll-outs during exploration. The cost function for PI^2 is simple: 0 cost if the objects is grasped successfully, 1 if it is not. A grasp is deemed successful if the relative position of the object to the gripper is within 3mm of the relative position determined by GraspIt! This implies that the actual grasp that GraspIt! computed is achieved.

Evaluation. The object positions for evaluation are sampled from the same distribution as the training positions. Instead of resampling the evaluation positions after each update, we rather sample them once for an entire learning session. This means fluctuations in the learning curve are really due to learning, and not due to random changes in the evaluation positions.

V. Empirical Evaluation

The experiments described in this paper are performed in simulation, with a Barret arm (7DOF) and hand (4DOF), as depicted in Fig. 1. The robot is modeled in the SL simulator [15], which accurately models robot dynamics, and robot-object contacts. We have integrated the Open Dynamics Engine [16] in SL, to accurately model the dynamics of objects, and object-object contacts.

A. Grasping cylindrical objects

For the first set of evaluations, we use a cylindrical object. The simplicity of this shape makes it feasible to interpret features of the resulting movement primitives. We also chose a cylindrical object to be able to compare our results to similar experiments in psychophysics [5]. The best grasp for this object (as determined by GraspIt!) that satisfies the constraints given by the table and robot arm kinematics is a straightforward power grasp at the center of the cylinder.

In this paper, the state estimation uncertainty in object position is modelled as a 2D Gaussian, where the standard deviation along the two main axes are 5cm and 0.5cm, unless stated otherwise. A standard deviation of 5cm to model the state estimation uncertainty in object position is also used in [8]. Learning was performed for 4 distributions, where the main axis of the 2D Gaussian is rotated along the vertical z-axis. In the remainder of this paper, we will refer to the 'rotation/orientation along the z-axis of the main axis of the distribution of objects' simply as α for brevity. The 4 distributions are visualized in Fig. 3/4, where the gray circle represents the cylinder at position μ, and the colorful star-shaped figure the $\mu + \sigma$ of the 4 different distributions with $\alpha = \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$.

The number of roll-outs per PI^2 updates is $K = 20$, i.e. the DMP is executed 20 times with varying parameters $\boldsymbol{\theta} + \epsilon$ for every PI^2 update. The initial variance of the exploration noise ϵ is chosen such that the average maximum mean absolute error along the three Cartesian end-effector coordinates is 1.4cm, along the orientation of the end-effector is 0.5°, and along the joint angles of the hand 6.4°. This means, for instance, that the maximum deviation from the noise-less trajectory along the x-coordinate of the movement is 1.4cm, averaged over all exploration trials.

B. Learning curves and analysis of the resulting motions

The learning curves for these four different orientations are depicted in Fig. 3. The initial success rate over the 20 evaluation trials lies between 0.85 and 0.75, so approximately 1 out of 5 grasps fails. Within 25 PI^2 updates, corresponding to 500 roll-outs, the success rate is increased to 1.0 for all object distributions. PI^2 is learning motion primitives that are able to grasp all the 20 objects in the evaluation set, and thus becoming intrinsically robust towards state estimation uncertainty.

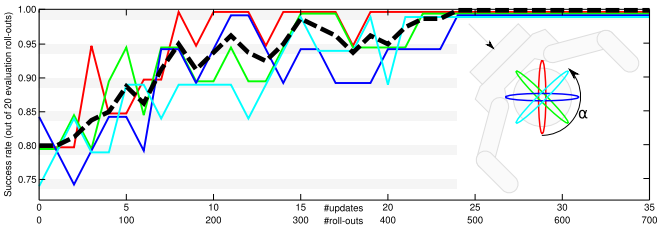

Fig. 3. Learning curves for the 4 orientations of the object distributions (α). Their average is the thick dashed graph.

Adaptation in end-effector pose. Fig. 4A depicts the trajectory of the end-effector position for each of the object distributions. As can be seen, the position of the end-effector at $t = T/2$ (i.e. when the end-effector motion stops and the hand closes) is adapted to the object distribution. In most cases, the end-effector moves beyond the initial position, such that it comes into contact with the object, and pushes it forward on the table before grasping it. By moving further, the robot is able to grasp objects that are farther away, whilst closer objects are pushed into place before grasping. Essentially, the robot is learning to use exploit physical contact with the object to increase the chance of successfully manipulating it. It is thus *learning* to perform fine manipulation *without* a model.

In Fig. 4B, the orientation of object distribution is plotted against the orientation of the end-effector in the x, y-plane at $t = T/2$. Although the effect is minor (the end-effector orientation ranges from 1.64 to 1.68 rad, which is just 2°) there is a strong and significant correlation between them ($R = 0.998$, $p = 0.002$), which indicates adaptation.

A more substantial effect is seen when considering the position of the end-effector relative to its initial position before learning. In Fig. 4C, a line through each of the preshape position of the end-effector of the learned trajectories is drawn through the preshape position of the initial trajectory. The orientations of these lines correlates strongly with the orientation of the distribution ($R = 0.992$, $p = 0.008$) This means that there is structure in the adaptation of the end-effector position: it is a 1-dimensional translation relative to the initial position along an axis rotated by α.

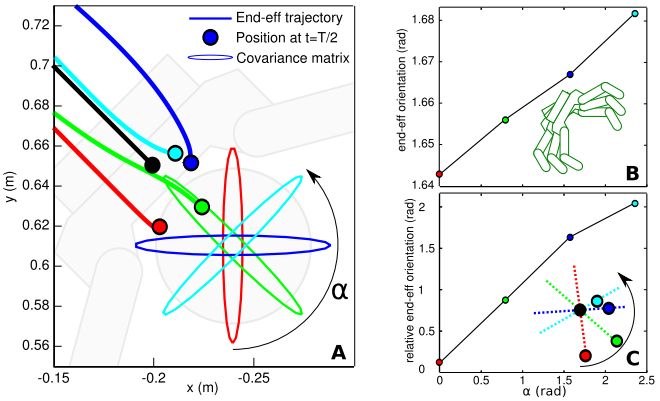

Fig. 4. A) End-effector trajectories near the object for different object distributions. Trajectories are plotted in the x, y-plane up to $t = T/2$. The position of the end-effector at $t = T/2$ is depicted by big dot. Colors of the covariance matrices representing the object distributions correspond to the colors of the trajectories. The initial trajectory before learning is black. B) Correlation between orientation of the main axis of the covariance matrix of the object distribution and the end-effector orientation... C) and relative orientation of the end-effector position to the initial position.

Adaptation in hand posture. In Fig. 5, the posture of the hand over time is depicted. The upper graph depicts the finger span (i.e. the distance between the fingers) for the 4 learned trajectories. In comparison to the hand posture before learning as determined by GRASPIT!, the maximum grip span increases for all trajectories. On average it is 31% more. By opening up the gripper, the robot is less likely to collide with the object during the approach. This might seem like an obvious improvement over the preshape provided by GRASPIT! Nevertheless, that is only because we know from experience that it is better to be safe than sorry when we do not know the exact position of an object. Any method (e.g. a grasp or motion planner) that does not take state estimation uncertainty or possible inaccuracies of models into account does *not* have this experience, and has no reason to prefer a more open gripper over one that moves very closely past the object.

The lower graph in Fig. 5 depicts the three learned joint angles of the fingers, averaged over all object distributions. The initial minimum-jerk trajectories (which are hardly discernible from each other because all fingers close simultaneously) are again depicted as black lines. From these graphs, it is clear that the fingers open up more, which we already knew from analyzing the increased maximum finger span in the upper graph. More interestingly, the right finger of the hand (i.e. the one that grasps the cylinder at a higher position) opens more than the left finger. This effect happens throughout almost the entire movement, but is only significant (t-test, $p < 0.05$) around $t = 2.0s$ as indicated by the thicker lines.

We verified the adaptive value of this difference on another 100 object positions, randomly sampled from the distribution with $\alpha = \frac{\pi}{4}$. With the learned movement for this distribution,

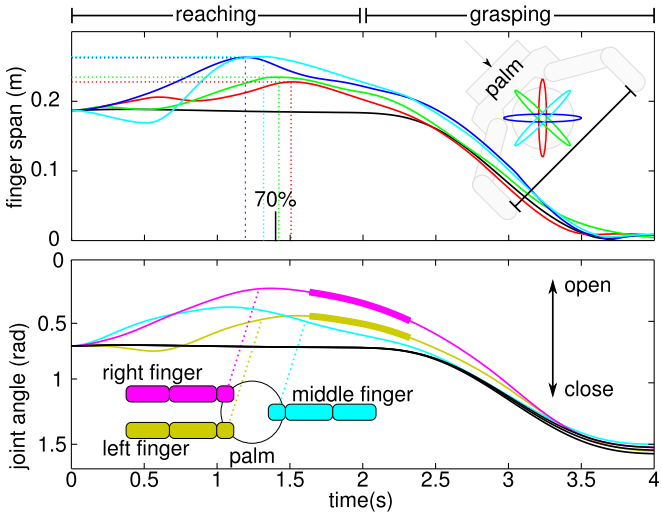

Fig. 5. Above: The finger span over time for each trajectory. Below: The joint angles trajectories of the three fingers, averaged over the 4 trajectories.

3 objects were not grasped successfully. We then used the joint trajectory of the left finger for both left/right fingers, in which case 10 objects were not grasped successfully. When setting both fingers to the trajectory of the right finger, this number is 7. So not only is the difference in joint angles between the left and right finger significant at $t = 2.0$, it also has adaptive value, as they exchanging them (in an otherwise symmetric hand) leads to lower performance.

The reason this is is that the right 'upper' finger grasps the cylinder just above its center of gravity, and the left 'lower' finger just below. Therefore, when coming into contact with the object, the upper finger is more likely to knock over the object, whereas the lower finger is more likely to shove it into a graspable position through fine manipulation. For this reason, the left finger on average opens up more than the right finger, to avoid premature contact with the object. So around the $t = 2.0s$ mark, the lower finger first pushes the object inwards towards the gripper, after which the upper finger closes in afterwards to complete the grasp. This strategy had not occurred to us before seeing the results of learning, and demonstrates that even very subtle movements in preshaping and grasping may contribute to successful manipulation. We believe that experience-based learning is the key to discovering such strategies.

C. Generalization to different positions

DMPs represent an attractor landscape towards a goal state g. By changing g, the attractor landscape changes, and the DMP is able to generalize to different goal states than the one with which it was trained [9]. In this experiment, we change the position of the cylinder on the table, and give this as the goal state g to the DMP. Note that only two values in g are changed; those relating to the x, y coordinates of the novel goal state. The main question we seek to answer here is: do the trajectories of the other dimensions in the DMP (i.e. those representing the orientation of the end-effector and the joint angles of the hand) also generalize to these different positions? To this end, we ran 20 evaluation trials with random object positions for $\alpha = 0.0$ at these different positions, and measured the success rate at each position. The results are depicted in Fig. 6. For good measure, the trajectory *before* learning is also evaluated for all positions.

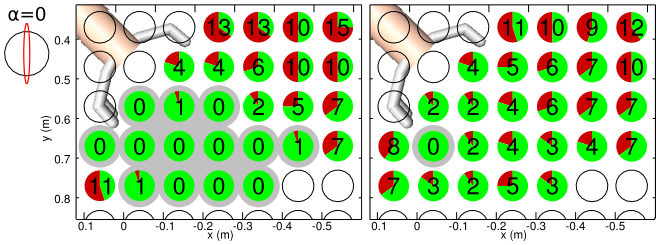

Fig. 6. Left: Generalization of the learned trajectory to new object positions for $\alpha = 0.0$. Right: Similar, for the initial trajectory before learning. The pie charts and the numbers represent the number of failed grasps out of 20. The positions in the upper left corner have not been tested, as they lead to an immediate collision with the end-effector. The lower positions are outside of the workspace of the robot. Positions for which the success rate $>= 0.95$ have been marked by a gray background.

The ability to grasp successfully in the face of state estimation uncertainty generalizes to quite a large area of approximately $0.4m \times 0.3m$, i.e. the number of failed grasps in the gray area in Fig. 6 is only 3 out of 260. The generalization is not good when the x or y coordinate of the position of the object is close to x or y coordinate of the initial position of the end-effector respectively. For these positions, a strategy that actively goes around the object is required.

The trajectory before learning only achieves a success rate of higher than 0.9 at one position. At this position, the movement of the end-effector towards the goal is exactly aligned with α. At positions where the learned trajectory does not perform well, the initial trajectory achieves similar or slightly better performance.

D. Generalization to different distributions

In a following experiment, the standard deviation is the same for both axes, but varied in magnitude, i.e. a circular distribution with standard deviations of 0.5, 1.0 and 5.0cm. The bold diagonal in Table I summarizes the results of learning after 30 updates.

Note that a standard deviation of 5cm in both directions is quite substantial, and it is not possible to learn a motion primitive that is sufficiently robust towards this level of uncertainty. The success rate does not go above 0.8, even when running 100 updates. This however, is not a shortcoming of our learning algorithm. It is simply not possible to successfully grasp all objects at the positions in the evaluation set with this manipulator in one motion. Essentially, the success rate of 0.8 gives us an indication of the best the robot can do, given the large amount of variance in the object's position.

We then used the three learned motion primitives, and cross-evaluated them on the other distributions. The off-diagonal values in Table I represent the success rates of these experiments. These values show that learning with high variance in object position generalizes well to low

		σ_{test} 0.5cm		1.0cm		5.0cm
σ_{train}	0.5cm	**1.00**		0.95		0.40
	1.0cm	1.00	←	**1.00**		0.60
	5.0cm	1.00	←	1.00	←	**0.80**

TABLE I

SUCCESS RATES WHEN TRAINING WITH A CERTAIN VARIATION, BUT TESTING WITH ANOTHER. ARROWS INDICATE GENERALIZATION.

variance, but not vice versa. The robot is thus not somehow 'overfitting' to high levels of variance. This is useful to know, because it means the robot can be trained with high variance, without compromising performance when variance is lower.

E. Grasping more complex objects

We now consider a set of more complex, non-convex objects, consisting of several boxes, similar to the ones used in [2]. For each object, we determine several possible grasps with GRASPIT! These grasps must: 1) have form-closure; 2) not collide with the table on which the object is standing; 3) lie within the workspace of the robot (i.e. some grasps from the top are not possible). Fig. 7 summarizes the results. The top row shows the three objects, and the grasps used on these objects, ordered according to (descending) grasp quality. From all the grasps proposed by GRASPIT! **H3** and **T2** represent two grasps with the highest volume quality measure (see [12] for details), and **T1** the lowest. The graph depicts the learning curves for all these grasps.

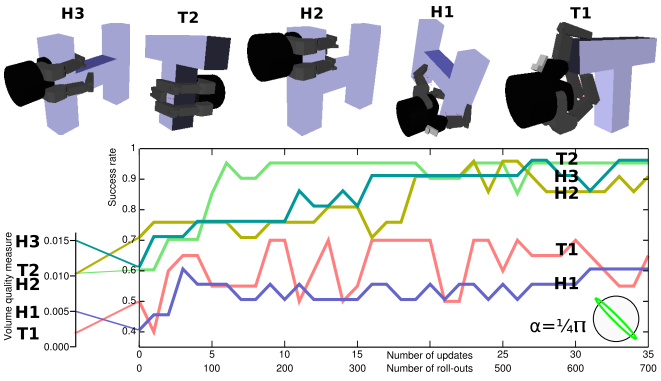

Fig. 7. Top: Several grasps for two non-convex objects, generated by GRASPIT! Bottom: Learning curves for these grasps when $\alpha = \frac{\pi}{4}$. The volume grasp quality measure as determined by GRASPIT! is plotted on the axis to the far left.

Initially, the success rates of the grasps determined by GRASPIT! lie between 0.40 and 0.75, and after learning, three grasps have a success rate around 0.95, and two around 0.6. The latter two do not improve, even after 100 updates. Note that the volume grasp quality determined by GRASPIT! is a good predictor of the initial and final success rates. And it also verifies that **T1** and **H1** are "cases where a grasp satisfies our quality metrics, but would require a degree of precision that cannot be obtained in real-life execution" [7].

Overall, the failure rates after learning are higher than in the case of the cylinder, due to higher chances of collision with protruding parts of the object. Nevertheless, it demonstrates that PI^2 is also able to learn motion primitives that are more robust towards state estimation uncertainty for more complex objects.

VI. CONCLUSION

In this paper, we present an approach that optimizes grasp success probability in the face of state estimation uncertainty. The optimization is done with a probabilistic reinforcement learning algorithm that takes only a simple boolean cost function that penalizes failed grasps, but does not require a model. Since reaching, preshaping and grasping are optimized simultaneously, they are not distinct phases, but rather an integrated motion, in which even subtle differences lead to large differences in performance, such as the order in which the fingers are closed. These trajectories are more robust towards state estimation uncertainty than a baseline trajectory based on preshape and grasp hand postures determined with the open-source grasp planner GRASPIT!

We are currently evaluating this approach on physical manipulation platforms. In these experiments, it will be interesting to see the adaptations to state estimation uncertainty that is not induced, but rather stems from the platform itself. Since our methods are model-free, they should be indifferent to the source of the uncertainty.

REFERENCES

[1] J. Bae, S. Arimoto, Y. Yamamoto, H. Hashiguchi, and M. Sekimoto. Reaching to grasp and preshaping of multi-DOFs robotic hand-arm systems using approximate configuration of objects. In *Proc. of the IEEE Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[2] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner. Grasp planning in complex scenes. In *IEEE-RAS International Conference on Humanoid Robots*, 2007.

[3] D. Berenson, S. Srinivasa, and J. Kuffner. Addressing pose uncertainty in manipulation planning using task space regions. In *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[4] J. Canny. On computability of fine motion plans. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, 1989.

[5] V. Christopoulos and P. Schrater. Grasping objects with environmentally induced position uncertainty. *PLOS Computational Biology*, 5(10), 2009.

[6] K. Goldberg. Orienting polygonal parts without sensors. *Algorithmica, Special Issue on Computational Robotics*, 10(3):201–225, 1993.

[7] C. Goldfeder, M. Ciocarlie, H. Dang, and P. Allen. The columbia grasp database. *International Conference on Robotics and Automation*, 2009.

[8] K. Hsiao, L. Kaelbling, and T. Lozano-Perez. Task-driven tactile exploration. In *Proceedings of Robotics: Science and Systems*, 2010.

[9] A. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Proc. of the IEEE International Conf. on Robotics and Automation (ICRA)*, 2002.

[10] M. Jeannerod. The timing of natural prehension movements. *Journal of Motor Behavior*, 16:235–254, 1984.

[11] T. Lozano-Perez, M. T. Mason, and R. H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1):3–24, 1984.

[12] A. Miller and P. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4), 2004.

[13] A. Morales, E. Chinellato, A. H. Fagg, and A. P. del Pobil. Using experience for assessing grasp reliability. *2004*, 1(4):671–691, International Journal of Humanoid Robotics.

[14] J. Peters. *Machine learning of motor skills for robotics.* PhD thesis, Department of Computer Science, 2007.

[15] S. Schaal. The SL simulation and real-time control software package. Technical report, University of Southern California, 2009.

[16] R. Smith. Open dynamics engine, 2004. `http://www.ode.org`.

[17] E. Theodorou, J. Buchli, and S. Schaal. A generalized path integral approach to reinforcement learning. *Journal of Machine Learning Research*, 11(Nov):3137–3181, 2010.

[18] Y. Zheng and W. Qian. Coping with the grasping uncertainties in force-closure analysis. *International Journal Robotics Research*, 24(4):311–327, 2005.