Learning to Grasp with Primitive Shaped Object Policies

Cristian C. Beltran-Hernandez[1*] Damien Petit[1] Ixchel G. Ramirez-Alpizar[1] Kensuke Harada[1,2]

***Abstract*— In this paper, we seek to improve the robotic grasping using reinforcement learning towards the automation of assembly tasks. We employed a reinforcement learning method based on the policy search algorithm, call Guided Policy Search, to learn policies for the grasping problem. The goal was to evaluate if policies trained solely using sets of primitive shaped objects, can still achieve the task of grasping objects of more complex shapes. The results show that even using simple shaped objects; the method can learn policies that generalize to more complex shapes. Additionally, a robustness test was conducted to show that the visual component of the policy helps to guide the system when there is an error in the estimation of the target object pose.**

I. Introduction

For the automation of the manufacturing industry, one essential component that needs to be accomplished is the assembly task. Assembly using manipulation robots, such as humanoid robots or industrial robot arms, is still considered a significant challenge due to the high complexity of the task and the difficulty in specifying the task for the robot for each new product. To overcome this challenge, tools such as an assembly planner has been proposed [1]. There are different types of assembly planner, here we consider the grasp/assembly planner, which includes grasping planning. In general, an assembly planner generates the instructions that the robot needs to complete a given task. The assembly task can be described by the components of the product and its relation to each other. For every component, one of the jobs of a grasp/assembly planner is to provide the most suitable grasping posture according to each specific manipulation task. However, for computing the grasping postures the planner requires an exact model of the component. This requirement makes the planner inflexible to deal with changes in the shape of the components of a product or changes in the workspace. To cope with this problem, we propose a novel approach where the assembly planner itself is solved by approximating the model's shape using a set of primitive shapes [2], and at the time of performing the grasping, the shape difference between each part and its approximated model is solved by using reinforcement learning.

[1] Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama-cho, Toyonaka, 560-8531, Japan.

[2] Manipulation Research Group, Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST), 1-1-1 Umezono, Tsukuba, 305-8560, Japan.

* Corresponding author e-mail: beltran@hlab.sys.es.osaka-u.ac.jp

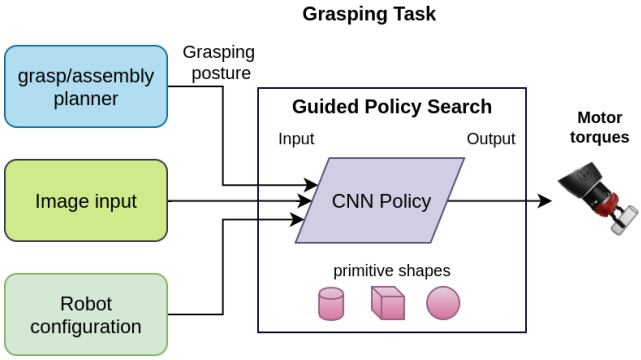

Fig. 1: Summary of the proposed method. Learning a convolutional neural network policy using the guided policy search algorithm. The policy uses the information from visual input and grasping posture from a planner. Training the policy using only sets of primitive shapes.

Recently, reinforcement learning algorithms have proven to have great potential to address a variety of problems, including robotic manipulation. The ability to explore and learn by itself prove to be very useful to handle novel situations outside the training scenarios. For this reason, here, we implemented a reinforcement learning algorithm to learn the grasping task. We used a policy search algorithm called Guided Policy Search [3]. The algorithm learns a policy to control a robotic arm based on visual feedback and the grasping posture. In our method, we propose representing the policy using a convolutional neural network that accepts an image, a grasping posture, and the robot configuration to control the robot arm, as shown in Figure 1.

The overall goal of our research is to develop a method for grasping objects with reinforcement learning taking advantage of information provided by the grasp/assembly planner, i.e., the grasping posture. More specifically, in this work our aim is to learn policies for grasping by training them using only a set of primitive shaped objects. Therefore, this paper seeks to give an insight for the question can a policy generalize to novel complex-shaped objects, when trained using combinations of different primitive shaped objects?

This paper is structured as follows. In Section II, we introduce related works. In Section III we give the details of the implementation of the guided policy search method for a Baxter robot[a], which will be essential for the learning of an effective policy in the final end-to-end framework.

[a]Rethink Robotics Baxter, Available http://www.rethinkrobotics.com/baxter/

In Section IV, we describe the experiment conducted to demonstrate the capabilities of the grasping of different shape objects via primitive guided policies. Details on the materials and training have been given so that the results can be reproduced. Conclusions are discussed in Section V.

II. Related work

Reinforcement learning (RL) seeks to solve the problem of how to learn new behavior automatically -how to map situations to actions- from only high-level cost/reward specifications [4], [5]. RL policy search methods have been used in robotics for a variety of manipulation tasks such as playing table tennis [6], ball-in-a-cup games [7] and object manipulation [8], [9], [10]. Most of these works use manual engineered representations to design specialized policies classes or features. However, the high dimensional complexity of robotic systems and unstructured environments demand additional constraints to the general reinforcement learning formulation to enable its application in the real world.

Recently, research in deep learning methods has proven effective in solving tasks such as static image recognition [11], where this method achieves a recognition considerably better than human ability. Thus, deep learning combined with RL methods have been developed to solve complex tasks using general purpose deep neural networks representations, this alleviates some of the burdens of manual engineered representations by using expressive policy classes. Moreover, end-to-end learning approaches have been proposed [12] where the system learns a joint model using vision input in a data-driven manner, such that, after collecting thousands of samples of successful and unsuccessful manipulations, the robot learn a model which controls the manipulation directly from input images. For instance, Pinto & Gupta [13] trained a convolutional neural network (CNN) for the task of predicting grasp locations by collecting a dataset of 50K data points; over 700 hours of robot grasping attempts needed to create the dataset. Also, Levine et al. [14], trained a deep CNN to predict the probability that the gripper will result in successful grasps using a dataset of 900K grasp attempts to learn hand-eye coordination for grasping; over two months and eight robots working simultaneously were used to collect this dataset. These methods reduce the burden of manual feature engineering though typically required a massive amount of training data, which is not practical for application.

Guided policy search (GPS) methods [3], [15] seek to address this challenge by decomposing the policy search into trajectory optimization and supervised learning of a general high-dimensional policy. These algorithms transform the policy search problem into a supervised learning problem, with supervision provided by simple trajectory-centric reinforcement learning methods. These trajectory-centric methods instead of training the policy directly, they provide supervision for training a nonlinear deep neural network policy from multiple different instances of the task (e.g. different poses of a target object). Thus, from an initial policy -that can be a random Gaussian controller- the GPS method iterates from drawing samples from the current policy, using this samples to fit the dynamics that are used to improve the trajectory distribution, and training the policy using the trajectories as training data. GPS has been applied to various robotic tasks [16], [17], [18], even in tasks that require contact-rich manipulation skills [contact-rich GPS], where it has proved to be able to acquire fast, fluent behaviors and can learn robust controllers for complex tasks. Since the GPS method allows a sample efficient learning of a policy, in this research, we consider using this method to learn a policy represented by a deep convolutional neural network for the grasping task of an assembly task.

Models based on sets of shape primitives have been used in grasp planner such as [2]. Moreover, the primitive shaped model has been used to guide the grasping posture and the actual shaped object has been used to plan a grasping posture. On the other hand, our method calculates a grasping posture of the primitive shaped model, and the difference between the actual object and its primitive shaped model is solved by using the reinforcement learning.

In this work, we study the ability to generalize a motor skill policy on the task of grasping objects with different shapes. This step is necessary in order to develop a learning process for a versatile assembly task using reinforcement learning.

III. Problem formulation

We consider the robot manipulation task of grasping a target object using a robot arm equipped with a simple parallel gripper. First, a grasp/assembly planner, such as the proposed in previous work [1], can be used to compute the desired grasping posture of the target object. On this step, we assume that the pose of the object is known in order to compute the grasping posture, error in this pose estimation can be mitigated thanks to the visual component of the policy as shown in Section IV-C. The grasping posture is computed by approximating the object shape using a set of shape primitives [2] and then estimating the appropriate posture for the manipulation task, e.g., a different posture might be obtained depending on the target pose of the object. In this work, we assumed that this process has already been done and that the grasping posture is available at the time of executing the policy.

The method proposed in this paper uses the guided policy search algorithm to learn a visuomotor policy that performs the grasping of an object at a given position.

A. Guided Policy Search

The reinforcement learning problem is considered as follows; there is an agent described by a state x. The agent can perform actions u and observe only part of the state, denoted as observations o. We consider a finite interaction of the agent with its environment during T time steps, a complete run of these time steps is denoted as an episode. In policy search algorithms, the goal is to learn a policy $\pi(u_t, o_t)$ for taking actions u_t conditioned on the observations o_t to

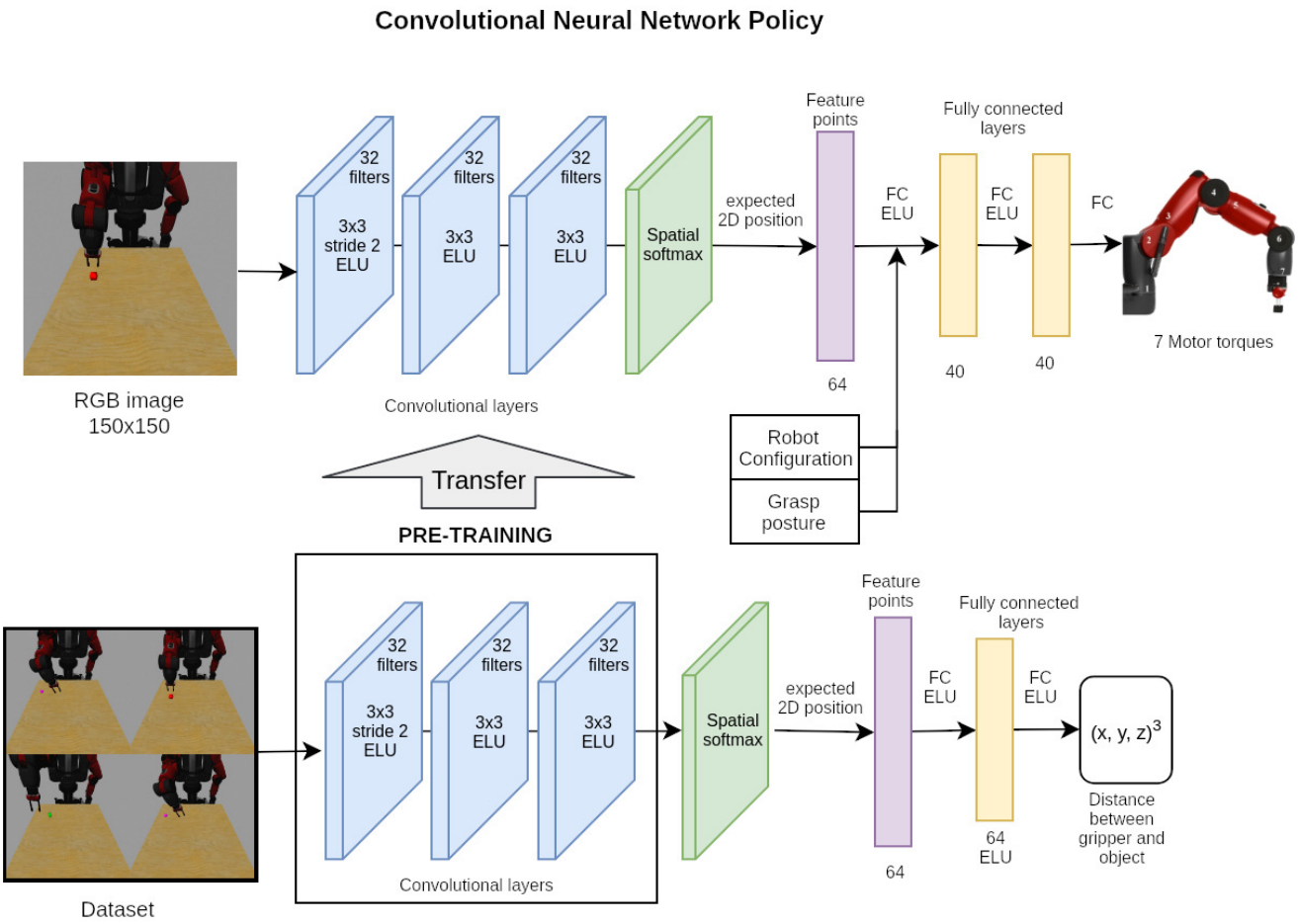

Fig. 2: Policy Architecture. Above is the convolutional neural network policy that accepts as input an image, the robot configuration, and the grasping posture. Below is the pre-training scheme network used to learn the basic visual features, a separate network trained to predict the distance between the robot gripper and the target object. The weights of the three convolutional layers are transferred to the policy network.

control a dynamical system. Given an stochastic dynamics $p(x_{t+1}|x_t, u_t)$ and a cost function $l(x, u)$, the goal is to minimize the expected cost under the policy's trajectory distribution, $\sum_{t=1}^{T} l(x_t, u_t)$.

In guided policy search, this optimization problem is addressed by dividing the problem into two components: a trajectory optimization part and a supervised learning one. Starting with an initial policy, which may be a random policy, sample the policy by running it on the robot, for each different training condition. These samples are store as trajectories of the form $\{x_t, u_t, x_{t+1}\}$. Then, in the trajectory optimization part, learn a linear controller using iterative linear quadratic Gaussian (iLQG) algorithm, the optimization is constrained to be closed to the trajectory described by the policy. Afterwards, more samples are drawn from the robot by executing the learned controllers. This dataset is used to learn a new policy in a supervised learning fashion. This completes one iteration of learning using GPS. Refer to [15], [12] for a complete description of the method. A summary of the method is displayed in Figure 2.

Our implementation is based on the work of Levine et al. [12], as to use a deep convolutional neural network to represent the control policy. In their original work, the algorithm guided policy search with Bregman Alternating direction Method of Multipliers (BADMM) was used to learn an optimal policy. However, in our work, we switch to the algorithm Mirror Descent Guided Policy Search (MDGPS). This method showed better performance than BADMM and requires substantially less manual tuning of hyper-parameters [15]. Our implementation was based on the open-sourced project GPS [19].

B. Convolutional Neural Network Policy

The policy is represented with a convolutional neural network to train vision and motor skill jointly. We propose the following architecture: the network contains three convolutional layers each composed of 32 filters, followed by a spatial softmax that describe the visual features extracted from the input image. The filter size of these convolutional layers is inspired on the model for image classification, Inception-v3 [20]. The visual features are then concatenated with the robot configuration and the grasping posture, then

	Policies						
	Cylinder	Cube	Sphere	Cylinder-Cube	Sphere-Cylinder	Sphere-Cube	Sphere-Cylinder-Cube
Cylinder	100%	90%	100%	100%	100%	100%	100%
Cube	100%	80%	100%	100%	100%	100%	100%
Sphere	100%	100%	100%	100%	95%	100%	100%
Duck	90%	80%	85%	100%	95%	95%	100%
Nut	100%	95%	85%	90%	100%	100%	100%
Mechanical part	95%	100%	95%	100%	100%	100%	100%
Mean	97.5%	90.83%	94.17%	98.33%	98.33%	99.17%	**100%**

TABLE I: Generalization experiment: The seven learned policies using different sets of primitive shaped objects. Success rates of each policy on each of the target object, including the object not seen during training. For each task 20 trials were performed.

passed through 2 fully connected layers, each one of 40 units, to produce the joint torques. Figure 2 shows a summarized description of the overall structure of the proposed policy representation, including the pretrain scheme that is discussed below.

To speed up the overall learning process, we follow the idea of pretraining independently each component and then performing a joint training. In the case of the visual components, we trained a separate convolutional neural network to predict the distance between the gripper position and the object position from an input image. First, we collected a dataset of about 2000 images containing the primitive shaped objects and the robot arm in different arbitrary positions. Then, we construct a CNN consisting of the same first 3 convolutional layers proposed for the policy, the spatial softmax layer and a fully connected layer followed by an Exponential Linear Unit (ELU) activation function that produces the prediction of the position (an array of 6 values), we found this activation function to perform better than a Rectifier Linear Unit (RELU). The filters in the first layer were initialized with weights of the Inception-v3 [20] model trained on ImageNet [21] classification dataset. The position of both the gripper and the object was encoded as 3 points in the space as shown in Figure 3. The training of this CNN was done using batch optimization with the Adam optimizer. After training, the weights in the convolutional layers are transferred to the policy network, enabling the robot to learn the appearance of the objects before learning the behavior.

Fig. 3: Robot end effector encoded as 3 points in space. Each point is represented by its x, y, z component.

On the other hand, as mentioned in Section III, for the motor skill component the policy search can be trained starting from a random policy, however, in order to speed up the learning process we train a linear quadratic controller, for each initial condition, without the visual component until it is able to succeed at the task at least 50% of the time. These controllers are used as the guiding trajectory distributions. Finally, we fully trained the policy combining the pretrained controllers and CNN.

C. Cost function

The cost function is a very important component of a reinforcement learning algorithm, it lets the robot know what is the objective of the task. In this work, the cost function for the grasping task was defined in terms of the distance from the gripper to the target object and then a reward is given to the robot if the grasp is successful. The following equation gives the cost function used:

$$\begin{aligned} l(x_t, u_t) =& w_{l_2} d_t^2 + w_{log} log(d_t^2 + \alpha) \\ &+ w_u ||u_t||^2 + w_g C_{grp} \end{aligned} \tag{1}$$

where d_t is the distance between three points in the space of the end-effector (Fig. 3) and their target positions, the weights are set to $w_{l_1} = 1.0$, $w_{l_2} = 10.0$, $w_u = 1.0$, and $w_g = 1.0$. The quadratic term encourages moving the end-effector toward the target when it is far, while the logarithm term encourages placing it precisely at the target location, as discussed in [18]. The term involving the action u_t is used to motivate the agent to find an energy-efficient trajectory. Additionally, the C_{grasp} term was defined as a reward for performing a successful grasp, this reward is given only at the last step of each episode. The reward was defined as

$$C_{grasp} = \begin{cases} \text{-10}, & \text{if grasp is successful} \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

Grasping is considered successful if after the robot attempts the grasp and retrieve the arm to the initial position, the object is still held by the robot gripper.

IV. Experiment and results

A. Parameters

All the experiments were conducted on a simulated Baxter robot on the Gazebo simulator. This simulator accounts for the elasticity of the joints actuators. The robot was controlled at 40Hz via torque control. The state of the robot was defined as:

$$x = \begin{bmatrix} q \\ \dot{q} \\ eef \\ \dot{eef} \\ vf \end{bmatrix}$$

where q is the robots joint angles, seven joints angles of the right arm of Baxter. The grasping posture (gp) and the end effector was expressed in the same frame and encoded as 3 points in space (see Figure 3), so eef is the difference between the current end-effector pose and the grasping posture $eef = eef_c - gp$, where eef_c is the current end-effector pose at any given time. This way, the target pose for any task is always 0. Additionally, the state includes vf, the visual features extracted from an RBG image input of size 150x150x3 through the CNN layers. The camera was kept fixed in each experiment. Each episode was 100 steps in length.

B. Generalization

The goal of this experiment was to see the effect of learning policies trained on different sets of primitive shaped objects, i.e, its ability to cope with new objects. The task for the robot is to grasp the object presented on the scene, one item is given at a time. The objects considered for training were a cylinder, a cube, and a sphere. Seven policies were trained, one for each possible set: only sphere, only cylinder, only cube, sphere-cylinder, sphere-cube, cylinder-cube, and sphere-cylinder-cube. Each of these objects had a similar dimension of 3.6cm of width, height, and length. Only consider one grasping posture was considered for all the target objects.

For training, each policy was the result of 15 iterations of pretraining the motor skills and three iterations of full training as described in Section III. On every iteration, 20 samples were collected using the controller learned at the previous iteration. When training a policy with a set of two different shapes, half of the samples were drawn using one of the shapes as the target object and half for the other one. For the policy that included all the primitive shapes, 30 samples were collected, 10 with each shape.

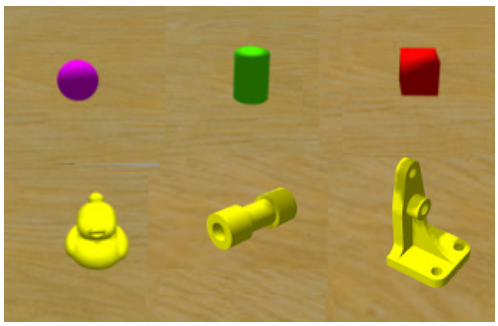

Fig. 4: Top, objects used for training: cylinder, cube, sphere. Bottom, objects, not seeing on any training, used for testing: duck, nut, mechanical part.

For testing, the policies were executed on each target primitive shaped object, additionally, they were also tested against novel objects that were not seen in any training: a duck, a nut and a mechanical part. All objects are displayed in Figure 4. Each test consisted of 20 trials. The experimental results are summarized in Table I.

The policies learned using only the cylinder and sphere shapes, separately, were able to learn to grasp its corresponding target object correctly. It seems that the radially symmetric shape of these objects makes it easier for the policy to achieve the grasp. However, in the case of the policy trained using a cube, the gripper needs to correctly approach the cube in a specific orientation to succeed, making the task harder to accomplish. Nevertheless, these policies were able to achieve a successful grasping of the novel objects most of the time. On the other hand, for the policies trained using sets of different shapes, the algorithm was able to better capture the features common to all of the training objects. As a result, the performance achieved by these policies was greater for both, the shapes included during training and the novel ones. All these policies fulfilled the task with a considerably high success rate on every target object. On top of that, the sphere-cylinder-cube policy got the best overall achievement.

Therefore, we can conclude that the proposed method is appropriate for learning the grasping task of novel objects when trained using only basic shaped objects. Additionally, using a more diverse set of shapes to train a policy seems to improve the ability to generalize to novel target objects.

C. Robustness

For the second experiment, we tested the two best policies from the previous experiment. The objective was to evaluate how well the policy adapts to errors in the object pose estimation, based on the visual component. The test involved inputting the policy with the grasping posture that included an error offset of the actual position of the target object. The offset was defined as 0.5 cm, 1.0 cm, and 1.5 cm along the x-axis. Each policy was tested on all the objects, and five trials per test were carried out. The results are shown in Table II.

In this case, we can see that the sphere-cylinder-cube policy was able to complete the task considerably well up to 1.0 cm of error in the position of the target. Nonetheless, the sphere-cube policy was able to adapt to most of all the test including the novel objects even at 1.5 cm of error. While it is not clear what exactly influenced this results, we can say that the learned policies do not depend only on the given grasping posture but also relies on the visual input, so that it can still achieve grasping of the objects despite the error in the object pose estimation. Furthermore, even for the targets that were not present during the training phase,

	Sphere-Cube			Sphere-Cylinder-Cube		
Error offset (cm)	0.5	1.0	1.5	0.5	1.0	1.5
Cylinder	5/5	5/5	5/5	5/5	5/5	5/5
Cube	5/5	5/5	5/5	5/5	3/5	3/5
Sphere	5/5	5/5	5/5	0/5	0/5	0/5
Duck	5/5	5/5	5/5	5/5	5/5	5/5
Nut	5/5	5/5	5/5	3/5	2/5	1/5
Mechanical part	5/5	4/5	0/5	5/5	5/5	0/5

TABLE II: Robustness experiment: the two best policies from the generalization experiment were tested including an error offset on the object position along the x-axis.

the policies were able to complete the task. It seems that the grasping posture helps guide the policy, regardless of the object, and the visual features help to correct for error in the pose estimation making it more likely to succeed.

V. Conclusion

In this paper, we presented a method for improving the grasping task in an assembly task using reinforcement learning, more specifically the Guided Policy Search algorithm. The proposed method was tested on different conditions to show that policies trained to grasp using only sets of simple shaped objects have the potential to generalize to scenarios with more complex shapes. Additionally, a robustness test was also performed to show that the visual features help the policy adapt to error on the target pose estimation. The results show the potential of using sets of basic shaped objects to learn grasping policies that can adapt to objects of more complex shapes, while guiding the overall task with a given grasping posture.

In this work, just one grasping posture was considered during training and testing. In addition, the cost function included only the grasping as successful or not. However, in the future, we plan to extend this method to achieve policies for the grasping task that are more flexible for a variety of grasping postures, where a successful grasp is evaluated with respect to the requested grasping posture. The overall goal is to combine the proposed method with the information available from a grasp/assembly planner to improve the assembly task. In addition, robotic experiments are also planned for future work.

Interesting topics for further extending the approach proposed here, includes applying online learning of the system dynamics, such as the proposed by Fu et al. [22], or reset-free guided policy search [23] for faster learning.

Acknowledgment

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

[1] W. Wan, K. Harada, and K. Nagata, "Assembly sequence planning for motion planning," *Assembly Automation*, vol. 38, no. 2, pp. 195–206, 2018.

[2] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2, pp. 1824–1829, IEEE, 2003.

[3] S. Levine and V. Koltun, "Guided policy search," in *International Conference on Machine Learning*, pp. 1–9, 2013.

[4] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[5] E. Theodorou, J. Buchli, and S. Schaal, "Reinforcement learning of motor skills in high dimensions: A path integral approach," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2397–2403, IEEE, 2010.

[6] J. Kober, E. Öztop, and J. Peters, "Reinforcement learning to adjust robot movements to new situations," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, p. 2650, 2011.

[7] J. Kober, B. Mohler, and J. Peters, "Learning perceptual coupling for motor primitives," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pp. 834–839, IEEE, 2008.

[8] M. P. Deisenroth, C. E. Rasmussen, and D. Fox, "Learning to control a low-cost manipulator using data-efficient reinforcement learning," 2011.

[9] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 763–768, IEEE, 2009.

[10] Y. Chebotar, O. Kroemer, and J. Peters, "Learning robot tactile sensing for object manipulation," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 3368–3375, IEEE, 2014.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[12] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[13] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 3406–3413, IEEE, 2016.

[14] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with large-scale data collection," in *International Symposium on Experimental Robotics*, pp. 173–184, Springer, 2016.

[15] W. H. Montgomery and S. Levine, "Guided policy search via approximate mirror descent," in *Advances in Neural Information Processing Systems*, pp. 4008–4016, 2016.

[16] S. Levine and P. Abbeel, "Learning neural network policies with guided policy search under unknown dynamics," in *Advances in Neural Information Processing Systems*, pp. 1071–1079, 2014.

[17] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 528–535, IEEE, 2016.

[18] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 156–163, IEEE, 2015.

[19] C. Finn, M. Zhang, J. Fu, X. Tan, Z. McCarthy, E. Scharff, and S. Levine, "Guided policy search code implementation," 2016. Software available from rll.berkeley.edu/gps.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255, IEEE, 2009.

[22] J. Fu, S. Levine, and P. Abbeel, "One-shot learning of manipulation skills with online dynamics adaptation and neural network priors," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pp. 4019–4026, IEEE, 2016.

[23] W. Montgomery, A. Ajay, C. Finn, P. Abbeel, and S. Levine, "Reset-free guided policy search: efficient deep reinforcement learning with stochastic initial states," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 3373–3380, IEEE, 2017.