Failure-mode Analysis of Learned Dexterous In-hand Object Manipulation with Tactile Feedback

AER 1517: Final Project

Abhinav Grover (1001257418)
(UTorID: groverab)
University of Toronto
Institute for Aerospace Studies

***Abstract*—The dexterous hand of primates is an extremely complex system which is a consequence of the hand kinematics and complex sensorimotor nervous system. There has been a lot of work in recent years to mimic this capability on robotic platforms. OpenAI has recently released a few papers that demonstrate state-of-the-art dexterity on the shadow dexterous hand using Deep Reinforcement Learning (deep-RL). Another recent report by Melnik et. al. [1] includes tactile information to the state of the system and demonstrates significant improvement in the dexterous performance. Based on this report, my paper explores the utility that tactile information provides for learning dexterous in-hand object manipulation. After training agents to perform 3 types of dexterous tasks, it is found that the volumetric properties of the manipulated object impact the kind of maneuvers learned by the deep-RL agent, and at the same time effects the utility of including tactile information to the environment. Tactile information offers significant utility for manipulating edged objects like blocks, and none for rounded objects like eggs. The code is available at https://github.com/abhitoronto/shadow_hand_experiments_openai**

I. Introduction

The human hand, combined with an intricate visual system, is an exceptionally complex system that is capable of performing hard tasks like object grasping, precise object motion etc., with ease. Our hand-eye coordination is owed to the multitude of sensory inputs we receive from the environment, which includes optical data, tactile data through the skin and interoceptive data from our motor nerves. Whereas, our dexterity is a consequence of the kinematics of our hand and the presence of an opposable thumb. Thus, for a seemingly simple task like holding an egg, our complex central nervous system combines the sensory data with the hand kinematics and controls the pose of the egg by changing the forces in various skeletal joints.

Our remarkable dexterity and hand-eye coordination has prompted the research community to attempt and mimic similar human capabilities on robotic systems. The researchers at OpenAI have achieved significant advances in this endeavour by approaching the problem from the perspective of model-free Deep Reinforcement Learning. They use the Shadow Dexterous Hand [2] as their robotic platform, which is a 5-fingered anthropomorphic robotic hand with 24 degrees of freedom. The resultant states-space of the system has 68 dimensions while the action space has 20 inputs. This high dimensional

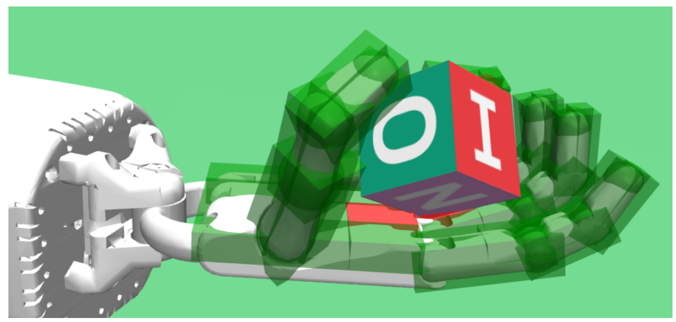

Fig. 1. 92 touch sensors covering the Shadow Dexterous Hand model and a simulated block in a random orientation [1]. Touch sensors are represented by green block and stimulated touch sensors are shown as red boxes.

state and action space renders many control strategies to be intractable, and prompts the use of Deep Reinforcement Learning techniques to achieve high degree of generalization. OpenAI has since published numerous algorithms to learn control policies for the shadow hand to conduct complex tasks like reaching, catching, and object manipulation [3][4][5][6]. Moreover, the company has publicly released the simulation setup for the shadow hand. This simulation setup uses the mujoco simulator [7] combined with the OpenAI gym [8], which is a unified toolkit to facilitate RL experimentation. As mentioned in [3], it is possible to simulate and reproduce the dexterity results using the the OpenAI gym, mujoco_py wrapper, and the OpenAI baselines [9]. OpenAI baselines are the open-source implementations of popular deep-RL algorithms like Proximal Policy Optimization (PPO) [10] or Deep Deterministic Policy Gradient (DDPG) [11] and is compatible with the mujoco_py+gym toolkit.

In [3] [4] the state of the simulated shadow hand includes the position of the finger tips, the hand joint angles, and the pose of the object being manipulated, but tactile information is not available. In real-life experiments, where the pose of an object is estimated using computer vision methods, tactile feedback from the hand may be necessary to perform in-hand manipulation tasks due to object occlusion or variation in lighting conditions [12]. A recent technical report by Melnik et. al. [1] released a new shadow hand mujoco based simulation model that has 92 tactile sensor units simulated along it's fingers and palm (fig 1). The report states that including tactile information as part of the system state significantly improves the learning rate as well as the final cumulative reward, as compared to policies without the tactile modality [4]. Yet the paper doesn't explain why the results are significantly better with the tactile information included.

This paper attempts to investigate why the inclusion of tactile data improves the learnt dexterous control with a deep-rl policy, through visual examination. The main contributions of this paper are listed as follows:

- Verify the reproducibility of the results in [1]
- An intuitive insight into the skills learnt to successfully manipulate an object using a dexterous hand
- A comparison of the skills learnt with and without tactile information, in order to explain the difference performance

II. Related Work

A. In-hand Manipulation

There has been prior work that attempts to solve the the complex problem of in-hand manipulation. Work by Y. Bai et. al. [13] propose a handcrafted technique to produce an object roll on the palm of a shadow hand, while Mordach et. al. [14] attempted to create an analytical control strategies for a generic dexterous hand. Because of the difficulty in modeling a dexterous hand there has also been some work on approaches that only train on a physical robot [15][16] [17], but the learnt behavior is quite limited.

Some of the recent papers have attempted to learn a general control policy using deep-RL algorithms [6]. Another recent paper by Andrychowicz et. al. [5] attempts to transfer simulation trained policies to real robot using domain randomization. This paper has been a break-through in the domain of learnt dexterous control, and there is work being conducted to make it more data efficient. Another paper by Plappert et. al. improves the performance and sample efficiency of deep-rl agent in sparse reward environments [4]. A recent extension to the work by Plappert et. al. mounts simulates touch sensors on the hand and includes that as part of the state [1], which improves the performance of the robot by upto 97%. This paper forms the basis of the implementation and experiments in our paper. Another paper by Rajeswaran et. al. [18] uses human expert demonstrations to accelerate agent learning in dexterous and prehensile tasks like placing objects or opening doors.

B. Tactile Sensors

An important part of manipulating an object, is knowing the state of the object with respect to the end-effector. Tactile sensors are one of the way to receive feedback from the physical environment about the state and properties of the object. There are many modalities of tactile sensing, including pressure or force distribution, temperature, slip, vibration, contact (on/off), and object properties such as texture [19]. In general, providing pressure distribution seems to be the most common modality in use.

Another upcoming tactile modality is detecting material deformation using optical sensors. This modality is especially useful for estimating surface contours and profiles. GelSight is one of most popular vision-based tactile sensor on the market [19].

C. Tactile Simulation

An important part of the problem is the ability to simulate tactile sensor data. There has been a a recent push in the research community for creating tactile simulators and data-sets. Sensenet [20] is a recently released set of tactile simulators and collection of feature rich objects for simulating tactile feedback. Work by Ding et. al. [21] quantifies the sim-to-real gap for an optical touch sensor simulated using the Unity engine. Additionally, researchers at OpenAI released a shadow hand environment with 92 simulated tactile sensors along with a set of manipulation tasks [1] provided with the OpenAI gym.

III. Methodology

This paper attempts to reproduce the results of the technical report by Melnik et. al. [1] and provide an explanation for the improved performance of the dexterous hand with the tactile sensors. Though intuitively it makes sense to us, it is important to explore the skills and maneuvers learnt by the deep-rl policy to achieve dexterous manipulation.

A. Simulation Platform

Initially the Sensenet simulation platform was tested for this purpose, but was soon rejected as an option due to its low maintenance and outdated instructions. Other options included two popular physics simulators: Pybullet and Mujoco. Pybullet is an open-source physics simulator that supports many environments which are used by the deep-rl community for the purpose of benchmarking. Mujoco is a similar physics simulator with numerous environments available, but it is proprietary software and requires a licence.

The RL toolkit provided by OpenAI works with both mujoco and Pybullet, but the models for the shadow dexterous hand are only compatible with the mujoco simulator [1]. Hence, it was a natural choice to work with the mujoco simulator which connects with the RL toolkit through the *mujoco_py* [7] python wrapper. It is important to note that the OpenAI gym at the time of implementation was only functional with mujoco version 1.50.

B. The Environment

The environment being used is based on the Shadow Dexterous hand which is an anthropomorphic hand with 24 degrees of freedom, but only 20 joints are controllable [2]. The state of the hand combined with the pose of the object make it a total of 68 states for this RL environment while the action space is 20 dimensional. Such large state and action space prompt the use of deep-RL to be able to get a tractable solution.

As mentioned in [1], there are two sensor configurations available: the first comprises of 92 touch sensors giving measured force values as floats, and the second is with all

92 sensors providing binary indication of activation as 92 boolean values. It can be seen in figure 1 that an activated touch sensor displays opaque red and stays translucent green otherwise. According to the report by Melnik et. al., there is no notable advantage of providing force values to the learning agent and hence this paper only uses the binary activation signals from the touch sensors; such models are indicated by the "*TouchSensors-v0*" suffix in their name. It is important to note that the state and action space of any shadow hand environment is continuous.

In this environment, the goal of the agent is to manipulate the pose of the object to achieve a goal pose. There are two options available for the reward configuration: the first one being a sparse rewards setup where a 0 reward is given for reaching the goal pose while every other state-action pair receives -1 reward, and the second one is a dense reward setup which, in addition to the sparse rewards, has intermediate rewards as well. Even though the dense reward setup is expected to converge faster and/or perform better, according to [4] the sparse reward setup combined with the DDPG+HER (Deep Deterministic Policy Gradient + Hindsight Experience Replay) gives the best results. It is stated that the agent is able to learn novel maneuvers only with the sparse reward setup which eventually leads to better performance.

C. Learning Dexterous Control

Hindsight Experience Replay (HER) is newly developed technique that enables RL agents to solve problems with large state-spaces and sparse rewards. Environments such as the shadow hand simulator with binary reward is a perfect example. The idea behind HER is simple: after experiencing an episode, the replay buffer stores each transition of the episode not only with the original goal but also with a subset of augmented goals [3]. That is, for each episode in the environment, the goal state is augmented to produce additional replay episodes such that some of the generated episodes reach the augmented goal, thus providing more information to the learning agent. As expected, HER only works with off-policy RL algorithms like DDPG [11] or DQN [22].

As mentioned in [4], the sparse reward setup combined with DDPG+HER (Deep Deterministic Policy Gradient + Hindsight Experience Replay) gives a high reward and improved convergence properties. Moreover, the report by Melnik et. al. also tested the novel shadow hand simulation with this deep-RL algorithm. Thus, the experiments in this report have been conducted in the sparse reward environment setup and the control policy is trained using a combination of DDPG and HER. Their implementation is available in the OpenAI baselines and requires the installation of the OpenAI gym as well as mujoco physics simulator.

D. Experiments

The goal of this paper is to intuitively explain the advantage of using the tactile modality during the dexterous manipulation of objects. There are multiple shadow hand manipulation tasks available in the OpenAI gym where the hand manipulates an object, like a cube, an egg or a pen. For example, figure 2 shows the HandManipulateEgg task and the HandManipulate-Pen task simulated with the mujoco simulator. I experiment with the hard yet comparable task of manipulating the pose of these objects to achieve a goal pose. Thus, for each object the goal state is 7-dimensional (3 cartesian + 4 Quaternion variables). Table I below lists the dexterous manipulation tasks being solved and compared in this paper.

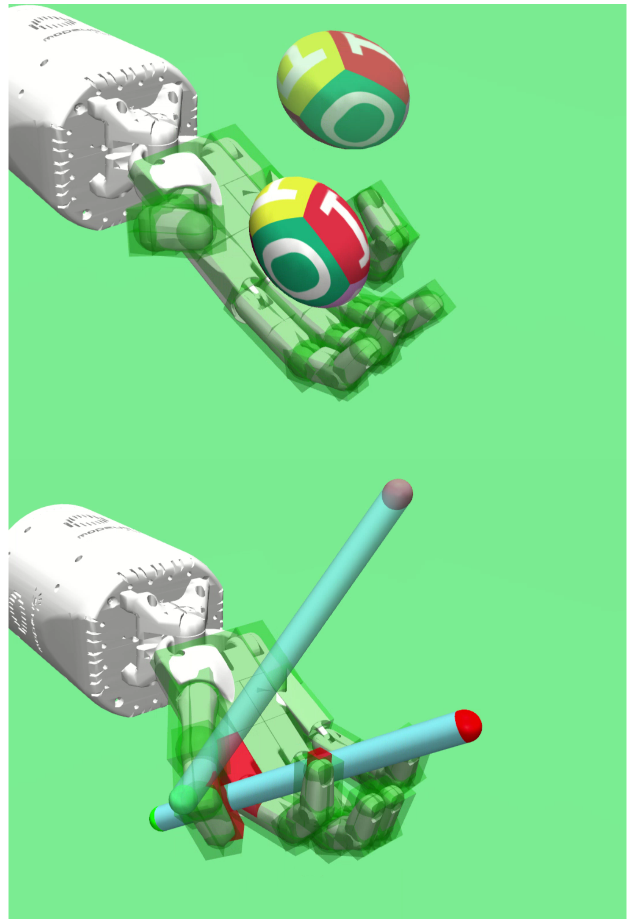

Fig. 2. Shadow dexterous hand manipulation environment with touch sensors. The translucent object is the goal state, while the actual object to be manipulated is present in-hand.

TABLE I
LIST OF SHADOW HAND MANIPULATION ENVIRONMENTS NAMES USED IN THIS PAPER TO COMPARE THE PERFORMANCE OF AGENTS WITH AND WITHOUT THE TACTILE MODALITY

Exp.	No Touch Sensors	With Touch Sensors
1	HandManipulateBlock	HandManipulateBlockTouchSensors
2	HandManipulateEgg	HandManipulateEggTouchSensors
3	HandManipulatePen	HandManipulatePenTouchSensors

The first step of the process is to train the RL agents using the DDPG+HER algorithm. In the report by Melnik et. al., the agents are trained for 200 epochs with 19 parallel runs for regularization. This paper trains the agents for the same 200 epochs, but due to hardware limitations only uses 10 parallel runs for policy updates. Other hyper-parameters have been kept identical to the ones mentioned in the report. The training curves of these agents would be used to verify the claims by Melnik et. al. and would be an exercise in the reproducibility of the HER learning technique.

Once the agents are trained, a visual inspection of the learned behavior will be performed to understand the dexterous skills learnt by the agents. Following the inspection, the failure cases of agents will be examined with special attention given to a comparison between agents with and without touch feedback for manipulating the same object (Table I).

IV. Results

Based on the proposed experiments in section III-D, the learning curves of the agents are provided in figure 3. It is important to note that the block and egg manipulation agents which have access to the tactile data from the environment have done consistently better both in terms of learning rate as well as performance. This validates the claims made by Melnik et. al. [1]. The most significant improvement is visible for the block manipulation task, where the performance improvement is visibly more than a 100%. The performance improvement for the eggs manipulation tasks is smaller but still significant.

If these plot are compared to the plots shown in [1], there is a clear difference in the performance level of the block agents, such that the agents show a poorer performance in comparison. This can be attributed to differences in hyper-parameters, such as number of parallel runs and replay buffer size, which had to be altered due to hardware limitations. It appears that the block agents have not yet converged and would require further training with the given hyperparameters.

A. Learnt Dexterous Skills

In an overview paper by Ma et. al. [23], in-hand manipulation has been classified into 6 categories: *Regrasping* (including manipulation using the inertial forces of the object), *in-grasp manipulation* (while maintaining consistent contact with the object), *finger gaiting* (repositioning fingers through object motion), *finger pivoting/tracking* (object rotation with two-point contact), *object rolling*, and *controlled slipping*. These classes form the fundamental techniques used by dexterous individual to manipulate any object in their hand, be it prehensile or non-prehensile manipulation. Here is a description of how the dexterous hand agents solve the three control tasks using the above mentioned maneuvers:

- *Block Manipulation* - This is a unique object (fig. 1) because of its corners which can act as lever points to apply moments for rotation. In order to match the correct block orientation, the agent uses a combination of rolling, regrasping, and finger pivoting depending on the axis of required block rotation. In order to match the position of the block, the agent uses a combination regrasping and in-grasp manipulation. This whole process is quite aggressive and seldom leads to the block falling down. This behavior makes intuitive sense though finger-gaiting can be used for finer position control.

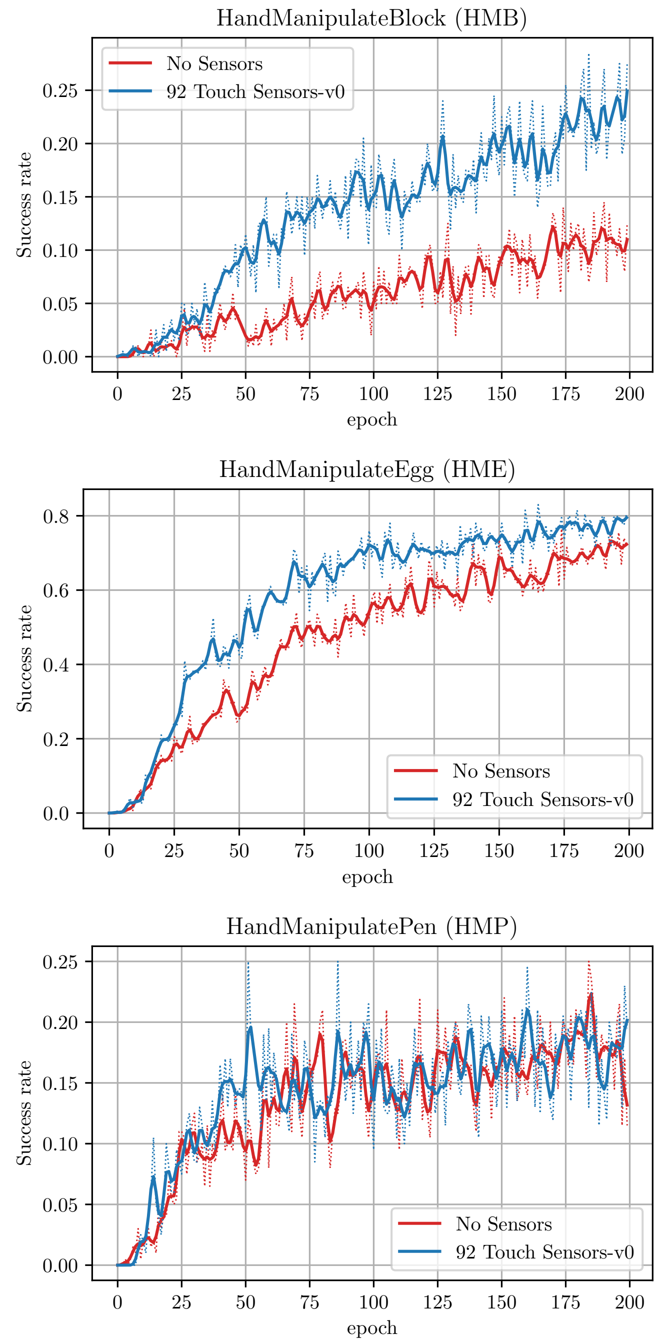

Fig. 3. Success rate learning curves comparing the performance of the shadow dexterous hand with and without touch sensors. The plots show the training curve for manipulating a box (top), an egg (middle) and a pen-like object (bottom). Success rate represents the average percentage of episode length that the manipulated object is held in the goal position. The curves have been smoothened with a gaussian filter, while the underlying learning curves is visible as a dotted curve.

TABLE II
PERFORMANCE OF THE RL AGENT FOR VARIOUS MANIPULATION TASKS. THIS TABLE ALSO INCLUDES THE FAILURE DISTRIBUTION OF THE AGENTS. EACH AGENT WAS RUN FOR AT LEAST 100 EPISODES.

Tasks	Task Performance		Failure Distribution		
	Success rate	Failure rate	Out-of-Reach	Out-of-Time	Unrecoverable
HandManipulateBlock with Touch (200 epochs)	**80%**	20%	10%	7%	3%
HandManipulateBlock w/o Touch (200 epochs)	58%	**42%**	**22%**	10%	**10%**
HandManipulateEgg with Touch (**50** epochs)	78%	**22%**	**16%**	2%	4%
HandManipulateEgg w/o Touch (**50** epochs)	**87%**	13%	7%	4%	2%
HandManipulatePen with Touch (200 epochs)	27%	**73%**	22%	1%	**50%**
HandManipulatePen w/o Touch (200 epochs)	**29%**	71%	**26%**	0%	45%

- *Egg Manipulation* - The smooth surfaces of a egg make it much easier to manipulate than a block (fig. 2). This fact is reflected in the learning performance of the manipulations agent to solve the two tasks (figure 3). The agent behavior in this case is quite similar to that of the block; the agent uses successive rolling and regrasping maneuvers to achieve the appropriate orientation, followed by regrasping and in-grasp manipulation maneuvers to place the egg at the appropriate position. The success of the agent lies in it ability to roll the eggs and subsequently stabilize it for for precise positioning.
- *Pen Manipulation* - The unique shape and weight distribution of this object makes is especially hard to manipulate (fig. 2). Though the exact maneuvers users to achieve the goal vary every time, maneuvers like regrasping, finger pivoting and sliding are the dominant manipulation techniques in the case of a pen. The use of opposable thumb becomes even more important here when the pen needs to be oriented vertically upright. Unfortunately, the agent is unable to learn the maneuvers required to do the horizontal planar rotation of the pen. This manipulation procedure requires careful finger pivoting and regrasping, which even humans find it hard to do with a single hand. Additional failure modes, including this one, will be discussed in the next section.

A video showcasing different maneuvers for each manipulation task is available here.

B. Failure Mode Comparison

This section goes over the failure cases for the shadow hand agents and attempts to give a heuristic explanation for performance difference between the two types modalities. These failures have been visually inspected and are defined as an inability to maintain the pose of the object with a certain tolerance of the goal state by the end of the episode. It is important to note that these limits are more lose than the limits defined in the environment and are based on the intuitive judgement of the author, and thus are prone to bias.

On doing a visual inspection of the agents episodes with and without touch feedback, for each object type, three major reasons for failure were recognised. The most common one is the object falling out of reach, which develops from successive mishandling of the object. The second most common failure mode is reaching an unrecoverable state; this is the most common failure mode in the case of pen manipulation. The last one, which is not as common as the other, is running out of time; this mode is more common when manipulating the block. Table II provides a distribution of these failure modes from various task and environment combinations. It is important to note that the majority of failure in the case of pen manipulation were due to the inability of the agent to horizontally invert the pen orientation, thus never reaching the goal state. It is also important to note that in order to get a reasonable distribution of failures for the egg manipulation agents, poorer performing agents, which were only trained for 50 epochs, were used. A video showcasing the various failure-modes is available here.

The performance distribution of the table II varies from that of the training plots in figure 3. Block and pen manipulation, which are expected to have similar results, have polarized performances in reality. Egg manipulation is expected to perform better in the presence of the touch sensor, but the trend is reversed on visual inspection of the policy. Moreover, the block manipulation agent appears to be performing significantly better, whereas it was expected to perform as badly as the pen manipulation agents.

These experiment give us an insight into why agents with access to the tactile modality would perform differently. The most significant difference is visible in the case of the block. It is worth nothing that majority of the falls, in the case of block manipulation, happen during a rolling and/or regrasping maneuver. In order explain the observed results, I speculate that the availability of touch modality aids in successful rolling and regrasping of an object with edges (to draw a distinction from the egg example). This also prevent the block from falling into unrecoverable states, thus reducing the percentage of that failure-mode. It is hard to explain the rise in the out-of-reach failure in the case of egg manipulation when touch modality in included, but one explanation could be that the agent is overconfident about the frictional force acting on the egg being held between fingers which is correlated with the touch feedback; this makes the agent often drop the egg.

V. CONCLUSION

This paper attempts to explain the utility of including tactile modality for an in-hand dexterous manipulation task learnt by a deep-RL agent. The reproducibility and the claims of the experiments by Melnik et. al. [1] are verified.

A visual inspection of the agent performance reveals the skills and maneuvers learnt by the agents to manipulate different kinds of objects. It also reveals a disconnect in the visually acceptable success rate and the calculated success rate of the agents.

It is discovered that inclusion of the tactile modality aids the agent in certain maneuvers like roll and regrasping, but only in the case of block like object with edges. The presence of tactile modality is slightly derogatory when manipulating an egg like object, and it makes little to no difference when manipulating a pen like object.

In order to understand the true advantage of the tactile modality, it is suggested to run experiment on a real robot where the state of the object is estimated using computer vision methods. Presence of the tactile modality would provide additional information about the pose of the manipulated object with respect to the hand.

References

[1] A. Melnik, L. Lach, M. Plappert, T. Korthals, R. Haschke, and H. Ritter, "Tactile Sensing and Deep Reinforcement Learning for In-Hand Manipulation Tasks," in *Different Aproaches, the Same Goal: Autonomous Object Manipulation, IEEE/RAS IROS2019 Workshop*, 2019.

[2] "Shadow dexterous hand - website," https://www.shadowrobot.com/products/dexterous-hand/, accessed: 26 April, 2020.

[3] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Advances in neural information processing systems*, 2017, pp. 5048–5058.

[4] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, V. Kumar, and W. Zaremba, "Multi-Goal Reinforcement Learning: Challenging Robotics Environments and Request for Research," 2018. [Online]. Available: http://arxiv.org/abs/1802.09464

[5] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *CoRR*, 2018. [Online]. Available: http://arxiv.org/abs/1808.00177

[6] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. TB, A. Muldal, N. Heess, and T. Lillicrap, "Distributed distributional deterministic policy gradients," 2018.

[7] "Openai python wrapper for mujoco physics simulator," https://github.com/openai/mujoco-py.

[8] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

[9] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov, "Openai baselines," https://github.com/openai/baselines, 2017.

[11] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015.

[12] T. Korthals, A. Melnik, J. Leitner, and M. Hesse, "Multisensory assisted in-hand manipulation of objects with a dexterous hand," *juxi. net*, 2019.

[13] Y. Bai and C. K. Liu, "Dexterous manipulation using both palm and fingers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1560–1565.

[14] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*. Eurographics Association, 2012, pp. 137–144.

[15] V. Kumar, E. Todorov, and S. Levine, "Optimal control with learned local models: Application to dexterous manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016.

[16] V. Kumar, A. Gupta, E. Todorov, and S. Levine, "Learning dexterous manipulation policies from experience and imitation," *arXiv preprint arXiv:1611.05095*, 2016.

[17] H. Van Hoof, T. Hermans, G. Neumann, and J. Peters, "Learning robot in-hand manipulation with tactile features," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 121–127.

[18] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," *Robotics: Science and Systems XIV*, Jun 2018. [Online]. Available: http://dx.doi.org/10.15607/RSS.2018.XIV.049

[19] A. Yamaguchi and C. G. Atkeson, "Recent progress in tactile sensing and sensors for robotic manipulation: can we turn tactile sensing into vision?" *Advanced Robotics*, vol. 33, no. 14, pp. 661–673, July 2019.

[20] J. Toy, "SenseNet: 3D Objects Database and Tactile Simulator http://sensenet.ai," Tech. Rep., 2017. [Online]. Available: http://sensenet.ai

[21] Z. Ding, N. F. Lepora, and E. Johns, "Sim-to-Real Transfer for Optical Tactile Sensing," 2020. [Online]. Available: www.robot-learning.uk/sim-to-real-tactile-icra-2020. http://arxiv.org/abs/2004.00136

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[23] R. R. Ma and A. M. Dollar, "On dexterity and dexterous manipulation," in *2011 15th International Conference on Advanced Robotics (ICAR)*, 2011, pp. 1–7.