Teleoperation System for Teaching Dexterous Manipulation

Stefan Zahlner, Matthias Hirschmanner, Timothy Patten and Markus Vincze[1]

Abstract— **Dexterous robotic hands are necessary for many tasks requiring the capability of in-hand object manipulation or tool use. Teleoperation remains a common approach for controlling dexterous hands, however, it is very difficult with traditional input methods (joystick, keyboards, etc.) or requires substantial investment in hardware setup (e.g. sensorized gloves). We introduce a teleoperation system based on the consumer-grade Leap Motion sensor to control a simulated model of the Shadow Dexterous Hand. Further, we present two continuous control tasks for the open source simulator PyBullet, integrated with the OpenAI Gym interface. We evaluate our system by using trajectories recorded by human experts as a supervised pre-training step before applying reinforcement learning (RL) algorithms. We show that the use of expert demonstrations accelerates the RL training process. The proposed system is available at: `https://rgit.acin.tuwien.ac.at/matthias.hirschmanner/shadow_teleop/`.**

I. Introduction

Dexterous multi-fingered robotic hands hold great potential to solve various complex anthropocentric control tasks. The continuous and high dimensional action spaces make dexterous manipulation with classical control methods difficult. This is especially true for the use-case of teleoperation in which a human controls the robot hand. Teleoperation can not only be used to operate the robot from a distance, but also to generate datasets of expert generated hand movements. These data can be utilized for imitation learning techniques in which the system is trained to replicate expert trajectories or to automatically generate annotated in-hand manipulation datasets for object tracking and pose estimation.

We introduce a teleoperation system, based on the Leap Motion sensor to control a simulated model of the Shadow Dexterous Hand and a humanoid model based on the CyberGlove [1]. Furthermore, we present two OpenAI Gym environments, integrated with PyBullet [2] as physics engine. The tasks are reaching different target fingertip positions and in-hand object manipulation of a block, similar to [3]. We evaluate the system by recording datasets for the two tasks and pre-train a policy using behavior cloning. We show that with a small number of human expert demonstrations, the RL learning process is sped up for one of the tasks.

II. Related Work

Teleoperation of robotic platforms with a high amount of degrees of freedom is often difficult and cumbersome, especially with traditional input methods (keyboards, mice, joysticks, etc.). Directly imitating the pose of a human teleoperator mitigates this problem. Different approaches using marker-based motion capture systems [4], [5] or motion capture suits [6], [7] are precise and reliable, but are expensive and often require an elaborate setup. In recent years, a variety of consumer-grade, marker-less human motion tracking systems have emerged. The Kinect camera was successfully used for full-body control of humanoid robots [8], [9]. The Leap Motion sensor we utilize in this paper has also been used for a variety of robotic platforms [10]–[12].

A popular approach for controlling robotic hands are various wearable sensorized gloves [13]–[15]. Marker-less deep learning based approaches are applied to successfully teleoperate a robotic hand from only depth images [16], [17] or RGB images [18]. However, these approaches might need substantial computing power and usually produce lower frame rates than dedicated hardware. Zubrycki and Granosik use the Leap Motion sensor to control a three finger gripper similar to our approach [19].

Reinforcement Learning is an approach to teach tasks to complex robotic end-effectors and showed impressive results in recent years (e.g. [20]). However, these methods require substantial training time and are often sample inefficient. Utilizing expert demonstrations during training is a popular approach to reduce training time [21]–[24]. Rajeswaran et al. show that this technique improves the performance for dexterous manipulations with a robotic hand [25].

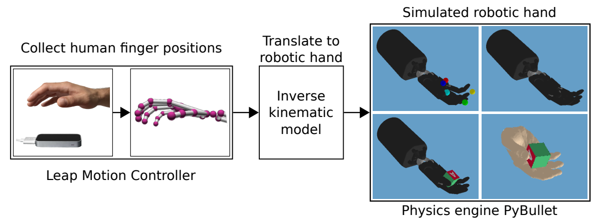

Fig. 1. Teleoperation system: The hand pose is detected by the Leap Motion Controller. We use an inverse kinematic model to control the robotic hand (Shadow Dexterous Hand or CyberGlove) in simulation.

III. Teleoperation System

This work utilizes the Leap Motion Controller as an optical hand tracking sensor as deemed suitable by [26]. It is an infrared stereo camera that provides the joint positions of the human hand by fitting a hand model to the image stream. Our system can control a simulated model of the Shadow Dexterous Hand [27], which is an anthropomorphic robotic hand with 24 joints. Of those, 20 can be controlled

*This work was partly supported by the Austrian Science Foundation (FWF) under grant agreement No. I3969-N30 InDex.

[1]All authors are with the Automation and Control Institute (ACIN), TU Wien, 1040 Vienna, Austria. stefan.zahlner@student.tuwien.ac.at, [hirschmanner, patten, vm]@acin.tuwien.ac.at

independently and 4 are coupled joints, hence it has 20 degrees of freedom. Additionally, our approach can also be applied to a visualization of the CyberGlove [1] which is similar to a human hand and can be used to generate datasets to investigate human-like hand-object interaction. Fig. 1 illustrates the described teleoperation system.

For the purpose of mapping human finger positions to robotic joint positions, an inverse kinematic model of the Shadow Dexterous Hand is derived based on the Denavit-Hartenberg formalism [28] and the retrievable data from the Leap Motion sensor.

Due to the complexity of a single model, it is subdivided into a total of six smaller models, one for each finger and one for the wrist. To ensure the independence of the hand's pose in the sensor's field of view, the human finger positions are retrieved relatively to the palm position. For details about the derived parameters we refer the reader to `https://rgit.acin.tuwien.ac.at/matthias.hirschmanner/shadow_teleop/`.

IV. Environments

We propose two tasks in the form of OpenAI Gym environments for the Shadow Dexterous Hand. The tasks are reaching a target position for each fingertip in `ShadowHandReach` and the more challenging in-hand manipulation of a block in `ShadowHandBlock`. Both environments are similar to the ones presented in [3]. We use the open source PyBullet simulator instead of the proprietary MuJoCo simulator of the original work.

The actions of both tasks are 20-dimensional regarding to the absolute positions of all non-coupled hand joints. Observations contain the 20 positions and velocities of the hand joints, as well as the Cartesian coordinates of all fingertips (current and target positions) for the reach task. In the object manipulation task, the observations include the object's Cartesian pose, linear and angular velocities, as well as the desired target pose. Both tasks can either operate with dense rewards, or with sparse rewards. This enables the choice between a signal of the negative distance to the goal and a robotic approach of success or failure.

`ShadowHandReach`: Starting from a fixed position, the fingertips need to reach the goal, which is randomly sampled from a set of 5 configurations. The goal is achieved and the episode ends, if the sum of all distances between the fingertips and the desired positions is less than 1 cm.

`ShadowHandBlock`: A randomly orientated block is placed in the hand's grip. The goal is to manipulate the object in-hand to reach and hold a specific target orientation within a predefined threshold (0.1 rad).

V. Results

In this paper, we evaluate the performance of Deep Deterministic Policy Gradient (DDPG [29]) and Hindsight Experience Replay (HER [30]) with and without pre-training on expert demonstrations. The experiments are conducted with both a dense and a sparse reward structure and mainly use the framework provided by [31], [32].

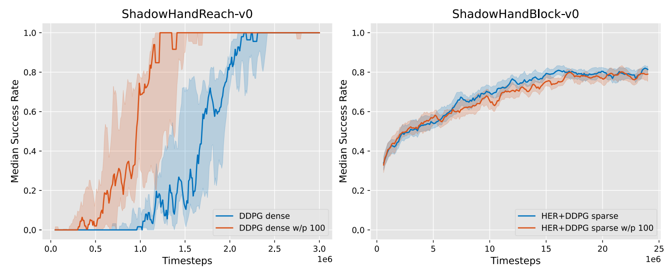

Fig. 2. Median success rate (line) and interquartile range (shaded area) with (orange) and without (blue) pre-training. Left: `ShadowHandReach` pre-trained on 25 trajectories for 100 iterations. Right: `ShadowHandBlock` pre-trained on 150 trajectories for 100 iterations. One training episode has a maximum of 100 timesteps, each timestep consists of 10 simulator steps with a frequency of 240 Hz.

`ShadowHandReach`: This simple environment is trained for a total of $3 \cdot 10^6$ timesteps using DDPG with an underlying Multi-Layer Perceptron (MLP) policy. For the pre-training a set of 25 trajectories is used. The experiments are repeated 5 times with different random seeds and the result is reported by computing the median success rate and the interquartile range, as depicted in Fig. 2. The pre-trained policy manages to successfully learn the task in approx. $1.2 \cdot 10^6$ timesteps. This is significantly less, compared to its counterpart without pre-training, which requires approx. $2.2 \cdot 10^6$ timesteps. This task has a long exploration phase without much learning progress in the beginning until it picks up learning at some point. The speed-up from pre-training mostly stems from reducing this phase.

`ShadowHandBlock`: The in-hand object manipulation environment is trained for $24 \cdot 10^6$ timesteps, using HER+DDPG with an MLP policy and 150 demonstrations for the pre-training. Fig. 2 depicts the results. For this task, pre-training does not improve training time. Contrary to the reach task, this task does not show a long exploration phase in the beginning. Instead, it has a significant success rate early on, because the randomly orientated block sometimes already starts in the goal orientation. Additionally, the block demonstrations lack quality, since it is difficult to operate an object in simulation without haptic feedback. We believe that these two factors are the main reasons for the lack of improvement using pre-training.

VI. Conclusion

We propose a teleoperation system to control a robotic hand in simulation. The system utilizes the Leap Motion Controller to capture human finger positions and converts them to robot joint positions. Additionally, we introduce two robotic control tasks in the PyBullet simulator. We use our system to record human expert demonstrations for these environments. The demonstrations are utilized to pre-train a policy using behavioral cloning before applying state-of-the-art RL algorithms. We show that pre-training significantly speeds up the training process for one of the tasks. In the future, we want to integrate better approaches to combine the demonstrations with the RL process to accelerate the training for more complex tasks as well.

References

[1] F. Heller, R. Jäkel, A. Kasper, P. Meißner, T. Nguyen, and X. Yi, "Asr cyberglove visualization," https://github.com/asr-ros/asr_cyberglove_visualization, 2016.

[2] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019.

[3] 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," *CoRR*, vol. abs/1802.09464, 2018.

[4] N. Pollard, J. Hodgins, M. Riley, and C. Atkeson, "Adapting human motion for the control of a humanoid robot," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2. Washington, DC, USA: IEEE, 2002, pp. 1390–1397.

[5] D. Matsui, T. Minato, K. F. MacDorman, and H. Ishiguro, "Generating natural motion in an android by mapping human motion," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3301–3308.

[6] J. Koenemann, F. Burget, and M. Bennewitz, "Real-time imitation of human whole-body motions by humanoids," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China: IEEE, May 2014, pp. 2806–2812.

[7] C. Stanton, A. Bogdanovych, and E. Ratanasena, "Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning," *New Zealand.*, p. 10, 2012.

[8] S. Filiatrault and A.-M. Cretu, "Human arm motion imitation by a humanoid robot," in *2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings*, 2014, pp. 31–36.

[9] Y. Ou, J. Hu, Z. Wang, Y. Fu, X. Wu, and X. Li, "A real-time human imitation system using Kinect," *International Journal of Social Robotics*, vol. 7, no. 5, pp. 587–600, 2015.

[10] D. Bassily, C. Georgoulas, J. Guettler, T. Linner, and T. Bock, "Intuitive and adaptive robotic arm manipulation using the Leap Motion controller," in *ISR/Robotik 2014; 41st International Symposium on Robotics; Proceedings of*. VDE, 2014, pp. 1–7.

[11] S. Chen, H. Ma, C. Yang, and M. Fu, "Hand gesture based robot control system using leap motion," in *International Conference on Intelligent Robotics and Applications*. Springer, 2015, pp. 581–591.

[12] M. Hirschmanner, C. Tsiourti, T. Patten, and M. Vincze, "Virtual reality teleoperation of a humanoid robot using markerless human upper body pose imitation," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 259–265.

[13] R. N. Rohling, J. M. Hollerbach, and S. C. Jacobsen, "Optimized Fingertip Mapping:A General Algorithm for Robotic Hand Teleoperation," *Presence: Teleoperators and Virtual Environments*, vol. 2, no. 3, pp. 203–220, Jan. 1993.

[14] F. Kobayashi, K. Kitabayashi, H. Nakamoto, F. Kojima, W. Fukui, N. Imamura, and T. Maeda, "Multiple joints reference for robot finger control in robot hand teleoperation," in *2012 IEEE/SICE International Symposium on System Integration (SII)*. Fukuoka, Japan: IEEE, Dec. 2012, pp. 577–582.

[15] C. Meeker, T. Rasmussen, and M. Ciocarlie, "Intuitive Hand Teleoperation by Novice Operators Using a Continuous Teleoperation Subspace," in *IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, May 2018, pp. 5821–5827.

[16] S. Li, X. Ma, H. Liang, M. Gorner, P. Ruppel, B. Fang, F. Sun, and J. Zhang, "Vision-based Teleoperation of Shadow Dexterous Hand using End-to-End Deep Neural Network," in *2019 International Conference on Robotics and Automation (ICRA)*. Montreal, QC, Canada: IEEE, May 2019, pp. 416–422.

[17] D. Antotsiou, G. Garcia-Hernando, and T.-K. Kim, "Task-Oriented Hand Motion Retargeting for Dexterous Manipulation Imitation," in *Proceedings of the Fourth International Workshop on Observing and Understanding Hands in Action at ECCV2018*, Oct. 2018.

[18] F. Gomez-Donoso, S. Orts-Escolano, and M. Cazorla, "Accurate and efficient 3D hand pose regression for robot hand teleoperation using a monocular RGB camera," *Expert Systems with Applications*, vol. 136, pp. 327–337, Dec. 2019.

[19] I. Zubrycki and G. Granosik, "Using Integrated Vision Systems: Three Gears and Leap Motion, to Control a 3-finger Dexterous Gripper," in *Recent Advances in Automation, Robotics and Measuring Techniques*, Cham, 2014, vol. 267, pp. 553–564.

[20] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving rubik's cube with a robot hand," *arXiv preprint*, 2019.

[21] M. Vecerík, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. A. Riedmiller, "Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards," *CoRR*, vol. abs/1707.08817, 2017.

[22] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 6292–6299.

[23] H. Zhu, A. Gupta, A. Rajeswaran, S. Levine, and V. Kumar, "Dexterous Manipulation with Deep Reinforcement Learning: Efficient, General, and Low-Cost," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 3651–3657.

[24] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. d. Freitas, and N. Heess, "Reinforcement and Imitation Learning for Diverse Visuomotor Skills," in *Robotics: Science and Systems XIV (RSS2018)*, 2018.

[25] 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," in *Robotics: Science and Systems (RSS2018)*, 2018.

[26] C. Mizera, T. Delrieu, V. Weistroffer, C. Andriot, A. Decatoire, and J.-P. Gazeau, "Evaluation of Hand-Tracking Systems in Teleoperation and Virtual Dexterous Manipulation," *IEEE Sensors Journal*, vol. 20, no. 3, pp. 1642–1655, Feb. 2020.

[27] Tadinu, "My arm," https://github.com/Tadinu/my_arm, 2019.

[28] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower pair mechanisms based on matrices," *Journal of applied mechanics*, vol. 77, no. 2, pp. 215–221, 1955.

[29] 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," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[30] 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.

[31] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines," https://github.com/hill-a/stable-baselines, 2018.

[32] A. Raffin, "Rl baselines zoo," https://github.com/araffin/rl-baselines-zoo, 2018.