Learning Dexterous Manipulation with a Low-Cost Soft Robotic Manipulator

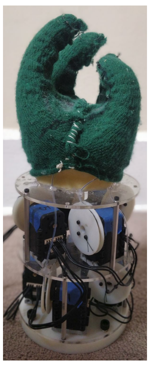
Karmesh Yadav*
Masters in Robotic Systems Development
Carnegie Mellon University

Introduction

With the advent of better learning techniques, the focus on research in robotic manipulation is increasing. However, the robot grippers available in the market currently are either not dexterous enough or they are quite expensive for large scale use. The goal of this project is to help in the development of a robust and versatile gripper which is low cost but also has similar functionalities to the high end robotic manipulators like the "shadowhand", etc. The hand is made out of foam, so it does not pose a risk to a human working alongside a robot and can be easily fabricated in a research lab. We carry out a lot of experiments on the hand to understand its current capabilities and get ideas for future design iterations.

In the coming sections we describe the soft hand in more detail, followed by a description of the training environment that was created during the course of this project for learning manipulation skills. Then, we briefly explain the learning algorithm being used for training the robot in the real world, and finally provide some details of the experiments that were carried out.

The Soft Hand Setup



For our experiments we use a four fingered tendon actuated soft foam hand[1]. The primary structure of such a hand is made of flexible foam and the hand is activated by driving servo mounted winches that contract (or slacken) tendons routed through the robot's textile skin. We use a total of eight tendons, where two fingers are provided with three tendons each while only a single tendon is used on the other two fingers. The three tendons allow those two fingers to move in the forward, left and right direction, while the other two fingers can move only forward from their natural resting state. This tendon configuration was decided after performing a lot of in-hand rotation experiments on the manipulator.

Training Environment

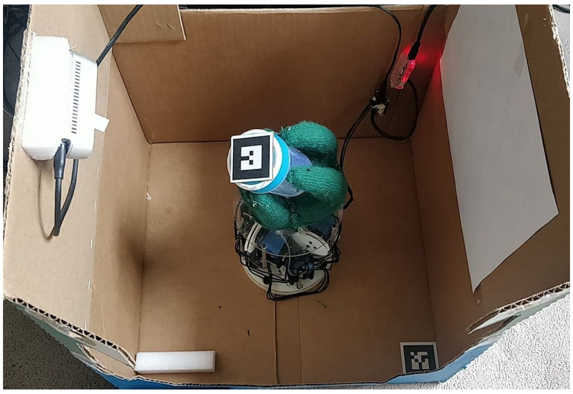
Hardware: The soft hand uses a high torque low velocity servo motor from Dynamixel (AX-12A) with each of the tendons. The tendons are attached to the motors through a pulley which is rotated to pull the tendon, leading to actuation of the finger in one direction. We use the position output from the motor encoders as a proxy for measuring the amount of tendon actuation and this information is passed on as the state of the finger to the policy.

We use a Realsense D435i camera to capture the state data of the object being manipulated. The camera is mounted a few inches above the hand to provide a large enough field of view. Currently, we use AR tags to measure the pose of the object but this could be replaced by more sophisticated point cloud based pose estimation methods in the future.

* This work was done under the guidance of Prof Abhinav Gupta

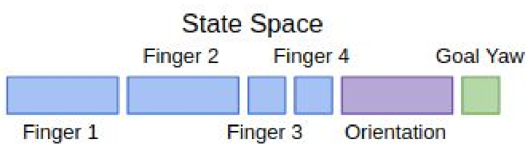
Moreover, to increase the richness of the information available regarding the state of the soft hand, we have ordered a set of SingleTact miniature force sensors which will be mounted on the tip of the soft fingers. These sensors will provide tactile feedback during object manipulation, which can be fed to a reinforcement learning algorithm for learning better policies.

The figure below shows the overall setup of the training environment. Because of the current COVID19 situation, a makeshift rig was used to mount the camera near the hand. The next set of work is to make this environment fully automated by designing a reset mechanism for putting the object back into the hand, if it falls during training.



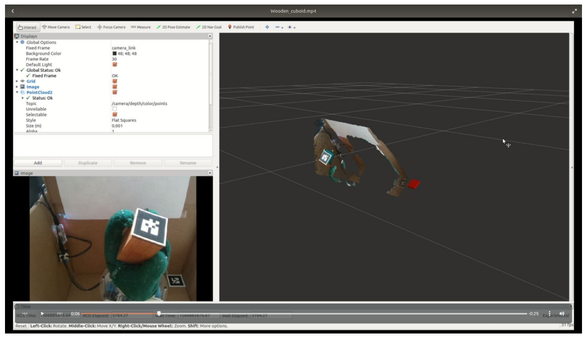
Software: To carry out experiments on the hardware setup, we required a robust and easy to use software API which could be integrated with the robot hardware as well as the deep learning libraries. We used ROS for communication with the camera and motors. The ROS code was integrated with OpenAI's Gym toolkit which is very commonly used for developing and testing environments for reinforcement learning research.

The gym environment gets the state information in the form of ROS messages from the motors and camera. The data is



normalized between 0 and 1 and combined together into a state vector. A goal yaw orientation for the object, which was sampled at the start of each episode, is also appended to the state vector. This data is sent to the policy which computes the optimal action. The action space is 1x8 dimensional as shown in the figure below. Each finger gets a command for forward and side motion, however the side motion command is ignored at the moment for fingers 3 and 4. After performing the desired action, the env returns the next state and the associated reward. We use a dense

reward for penalizing the distance between the current yaw angle and the goal yaw angle. A view of the setup from the Rviz window is given in the figure below



Learning Algorithm

We are using a state of the art, model based reinforcement learning algorithm called PETS (Probabilistic Ensembles with Trajectory Sampling)[3]. PETS tries to learn a model of the environment using high capacity NN models that incorporate uncertainty via an ensemble of bootstrapped models, where each model encodes state distributions. PETS is able to rival the performance of model-free methods on standard benchmark control tasks at a fraction of the sample complexity which makes them enticing for usage on real world rl tasks. After the model is learnt, any off the shelf, sampling based planner (eg random shooting or CEM) could be used for planning optimal actions in an MPC fashion. The main algorithm behind PETS is given below.

Algorithm 1 Our model-based MPC algorithm '*PETS*':

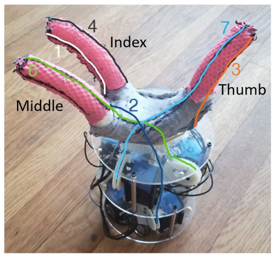
Initialize data $\mathbb{D}$ with a random controller for one trial.
for Trial $k = 1$ to K **do**
  Train a *PE* dynamics model $\widetilde{f}$ given $\mathbb{D}$.
  for Time $t = 0$ to TaskHorizon **do**
    for Actions sampled $\boldsymbol{a}_{t:t+T} \sim \text{CEM}(\cdot)$, 1 to NSamples **do**
      Propagate state particles $\boldsymbol{s}_\tau^p$ using *TS* and $\widetilde{f} | \{\mathbb{D}, \boldsymbol{a}_{t:t+T}\}$.
      Evaluate actions as $\sum_{\tau=t}^{t+T} \frac{1}{P} \sum_{p=1}^{P} r(\boldsymbol{s}_\tau^p, \boldsymbol{a}_\tau)$
      Update $\text{CEM}(\cdot)$ distribution.
    Execute first action $\boldsymbol{a}_t^*$ (only) from optimal actions $\boldsymbol{a}_{t:t+T}^*$.
    Record outcome: $\mathbb{D} \leftarrow \mathbb{D} \cup \{\boldsymbol{s}_t, \boldsymbol{a}_t^*, \boldsymbol{s}_{t+1}\}$.

Experiments

Experiments were carried out with a variety of objects including things found in any household as well as some custom 3D printed objects. Images of some of the objects are given below.

Initial experiments were carried out using a small cube with AR tags on all the faces with the aim of performing full in-hand cube manipulation. However, the cube would get occluded between the fingers and it was impossible to get the state estimate of the AR tags.

In all the following experiments, we focussed on one-directional rotation of tall objects between the fingers. Using household objects for these experiments were challenging because the objects were either too heavy or slippery or large for our current hand. We therefore designed and 3D printed a hexagonal and a decagonal prism for further experiments. We have been successfully able to rotate these objects in the hand using the previously described planning methods. Further work will be done on training models with PETS to obtain robust policies.



This work has also helped us brainstorm better hand designs which have higher dexterity but are cheaper and lighter than before. The three fingered non anthropomorphic hand on the right was designed by a fellow researcher at CMU based on these ideas.

Conclusion

In this work, we presented our approach for learning dexterous manipulation on a tendon actuated soft hand. We start by first introducing the robotic hand and its actuation principles. This is followed by a brief description of the soft hand gym environment which can be used easily with any of the off the shelf reinforcement learning algorithms. We then talked about PETS, a model based reinforcement learning technique which only requires only a handful of trails to perform comparable to model free rl techniques. Finally, we discussed the experiments that were carried out on the hand which were critical for making further improvements on this hand and for the design of the three fingered hand.

In future work, we will be carrying out experiments on the new 3 fingered robotic soft hand. We will keep improving the soft hand design based on the feedback from the dexterity experiments. Another possible research direction is to investigate the performance of this hand for tasks other than table top manipulation.

References

[1] King, Jonathan P., et al. "Design. Fabrication, and Evaluation of Tendon-Driven Multi-Fingered Foam Hands." *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018.

[2] Nagabandi, Anusha, et al. "Deep Dynamics Models for Learning Dexterous Manipulation." *arXiv preprint arXiv:1909.11652* (2019).

[3] Chua, Kurtland, et al. "Deep reinforcement learning in a handful of trials using probabilistic dynamics models." *Advances in Neural Information Processing Systems*. 2018.