CSCI 5525
Project Proposal

Haoyi Shi

I. Problem Description and Motivation

RECENTresearch shows success in using deep reinforcement learning (DRL) which uses a neural network to simulate value function train control policy for robots in a virtual environment and do trajectory planning for robots in the real world. Moreover, the recent DRL algorithm Proximal Policy Optimization (PPO) and Twin delayed deep deterministic policy gradients (TD3) showed greater sample efficiency and reduced approximation error. In this project, I want to use DRL techniques to train a control policy for a 7-degree-of-freedom kinova manipulator that perform object-reaching task with obstacles avoiding.

A. Why use machine learning?

By comparing with classic control methods such as PID control, using machine learning methods such as DRL can find better solutions for highly dynamic or non-linear environments. Also, traditional control methods do not involve exploration since they rely on predetermined control rules or models. On the other hand, DRL can learn control policies by interacting with an environment.

B. Initial solution idea and Approach

First, I will use a sim-to-real strategy, which training a control policy in the virtual environment. In the virtual environment, I need to represent the robot using the URDF robot model which has the same action space as the real robot, and an object as a target for the robot end-effector to reach. Moreover, I need to map the action space and observation space into the DRL algorithm which can used during policy training. For complex tasks, I may need to use the experience copy technique which trains a policy for easy tasks, and use transfer learning to re-train the policy for complex tasks.

II. List of Required data/tools/resources

Hardware: GPU for accelerated machine learning computations. Ubuntu operating system for compatibility with machine learning libraries and ROS software.
Software: ROS (Robot Operating System) for communication and control of the kinova manipulator. Virtual environment simulation platforms such as Mujoco or pybullet for training and testing the control policy in a virtual environment.
Robot Models: URDF model containing the Kinova Gen3 robot and the target object for simulation in the virtual environment.
Reward Function: Development of a customized reward function tailored to the object-reaching task with obstacle avoidance. This function will serve as a critical component in training the reinforcement learning agent.
Deep Reinforcement Learning (DRL) Libraries: Utilization of DRL algorithm libraries, including Proximal Policy Optimization (PPO) and Twin delayed deep deterministic policy gradients (TD3), to implement and train the control policy.
Simulation Environment Setup: Configuration and setup of the virtual environment, including the creation of a robot model with the same action space as the real robot and the inclusion of the target object.
Experience Copy Technique: Implementation of the experience copy technique, involving the training of a policy for simpler tasks and the subsequent transfer learning to adapt the policy to more complex tasks.
Robot-ROS Bridge: Establishment of a communication bridge between real world robot and the control policy, enabling real-time interaction and control of the Kinova manipulator in the physical environment.

In the conclusion, by leveraging these resources and utilizing deep reinforcement learning techniques, this project aims to address the problem of training a control policy for a 7-degree-of-freedom Kinova manipulator, enabling it to perform object-reaching tasks while autonomously avoiding obstacles. The combination of hardware, software, simulation environments, and custom reward functions will facilitate the development of an effective and adaptive control policy. This project represents a modern and data-driven approach to solving complex robotics challenges in dynamic and non-linear environments.