Sensors and Motor Control Lab

Individual Lab Report – 4

Nithin Subbiah Meganathan

Team D: CuBi

Team Members

Jorge Anton Garcia

Paulo Camasmie

Laavanye Bahl

Changsheng Shen

Carnegie Mellon University

28th March 2019

Individual Progress

Dynamixel motor issue was finally resolved during this period. The problem was most likely due to the faulty hardware and the connection. We were using USB2Dynamixel to communicate to the motors and ArbotiX controller to power the motors. Bobby got help from some PhD students at RI to establish that the motors were not at fault. This was made possible by just using the non-faulty hardware from their lab.

Jorge and I worked on the following together. A U2D2 Robotis USB to Dynamixel adapter was used to both communicate and power the motor. Each motor was connected individually to set ID and baud rate. This was made possible through Mixcell package on Linux.

Now that the ID's and baud rates were set, I tried to use the ROS Dynamixel package in order to control the motors. After configuring the launch file with the motor's properties, a node was run to detect the motors. All motors were detected on ROS. But when I tried to publish some value the motor did not respond. Later, I found that the baud rates were set low and that was the reason why the motors did not respond.

The motors were then configured to a higher baud rate. A yaml file was created with four joint controllers, one for each motor. A launch file was created which calls the yaml file and controls are established. The ROS Dyamixel package publishes a bunch of topics among which one is used to rotate the motors.

A node was created that subscribes to the topics that publishes the desired position and velocity to the motors. A publisher node then publishes these to the ROS Dynamixel package topics which controls the motors.

Challenges

With respect to the Dynamixel motors, I did not face any unforeseen challenge as I did before. One huge lesson learnt was that I must test every hardware before putting it into use. Also, the ROS packages for Dynamixel does not have good community or forum support so it was a bit hard to resolve some issues. Another major challenge that I face is failing to think of the system as a whole. I got stuck with a small task that consumed a lot of time when I should have spent time on developing other subsystems.

Teamwork

Laavanye created a URDF for the robot. He also made an initial attempt at creating bounding boxes with orientation around the objects. Paulo created a different adapter to replace a MX-106 motor in the wrist with a MX-28T. He also modified the length of the adapter of the first motor close to the base such that the arm can be closer to the robot with a compact configuration. He also designed and fabricated a mount for the

camera and assembled the manipulator on to the robot. Bobby integrated the control of Dynamixel motors using joystick. He also made initial attempts at transferring SLAM that he had performed already on a better LiDAR system. Jorge worked on the Dynamixel motors alongside with me.

Figures

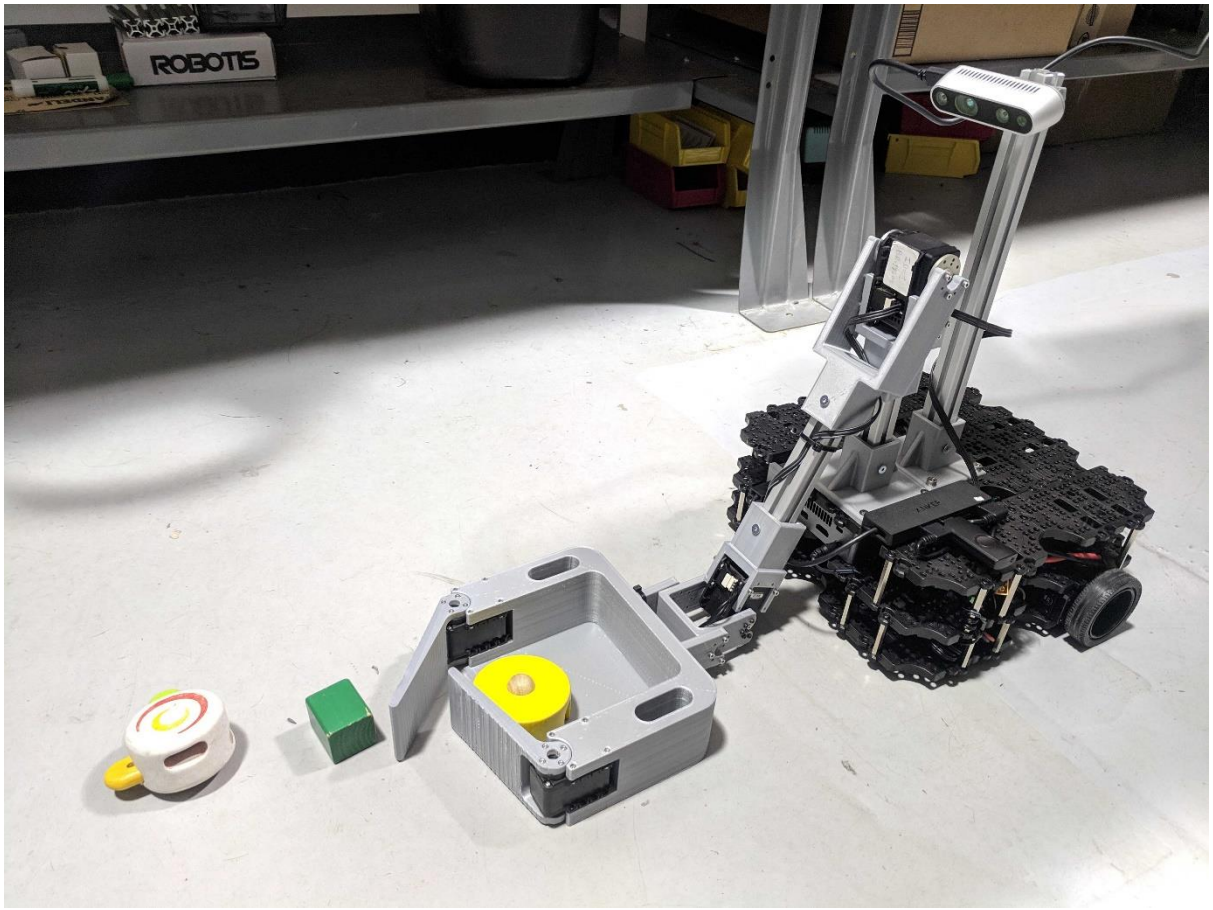

Fig 1. CuBi with the assembled manipulator setup and Dynamixel motors

Plans

Now that the task of controlling the Dynamixel is over, I will be shifting towards Machine Learning and Computer Vision subsystem. Laavanye is attempting a traditional Computer Vision approach to detect objects. My plan is to use Deep Learning techniques for object detection. For that, a pipeline that takes in frames of images from the camera should be obtained. I am thinking of using a single-stage detector like RetinaNet or YOLO since the system has to be real time. Another important task is to plan our final month's work keeping the spring validation tasks in mind. It is essential to plan our work with the final system in mind so that we don't lose track.