Sensors and Motor Control Lab
Individual Lab Report – 2

Nithin Subbiah Meganathan

Team D: CuBi

<u>Team Members</u>

Jorge Anton Garcia

Paulo Camasmie

Laavanye Bahl

Changsheng Shen

Carnegie Mellon University

20th February 2019

Individual Progress

My progress was achieving the control of a Dynamixel AX-12A motor, two of which we have planned to use for the gripper. A 12V power supply was provided to the motor using ArbotiX-M controller. The motor was then connected to the PC using a USB2Dynamixel connector which eases the process of sending commands from the PC to the motor. The setup was initially tested on Ubuntu environment. Software package was for Dynamixel motors were downloaded and detailed instructions were followed to finish the software setup. But when a program to change the position of the motor was run, the motor did not respond. The correct ID and baud rate were provided but still the motor was not responding. Dynamixel motors can run on two protocols 1.0 and 2.0 but both were unable to make it work. The power supply to the motor was checked and it was working properly.

The non-availability of Dynamixel API for Ubuntu was hurting so the same hardware setup was implemented on Windows. Robotis provides a software to explore all the Dynamixel features on PC called RoboPlus. Using the software firmware update was done on the motor. The motor was then controllable using the software. One potential reason why the motor did not work on Ubuntu might be due to some old firmware running on the motor. The motor which controlled the other end of the gripper was not controlled because it was broken. Thus, CuBi was able to pick objects off ground successfully. The motor provides high amount of torque and could pick heavier objects than expected.

Since we are using Dynamixel motors for all joints, achieving control for other motors (MX-64, MX-106) will be similar. Jorge accompanied me with completing these tasks.

Challenges

The primary challenge was controlling the hardware using software. As discussed in the previous section, the motor was not working while we were on the Linux environment. It was hard to find the root cause and we spend an awful amount of time on this problem. But when we switched to Windows and updated the firmware, we found that it is due to the firmware already installed. Since we borrowed the motors from the inventory, we should have made certain how to reset them to its original condition.

Teamwork

Paulo had finished designing the entire manipulator and had been responsible for 3D printing and assembling everything. Jorge had helped me with the task of controlling the motors and had also created a detailed schedule based on our planning. Bobby had integrated the microprocessor with the Tutlebot base, affixed new batteries, and had also made the control of the base smoother. Laavanye had performed further processing of the obtained point cloud data.
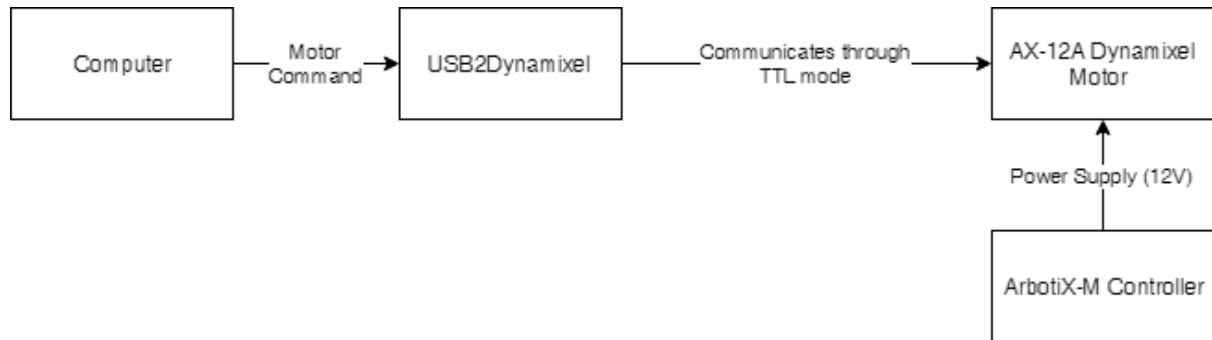
Figure

Figure 1. Connection of Dynamixel Motor

Plans

The next plan of action would be to make sure all the motors in our stock are in proper working condition. Then I should be able to control the motors individually and independently. As soon as this is achieved, it is desirable to control the motors using ROS stack. These would be my next steps. As for the team, we need to prototype and finalize on the manipulator design. We should be done with setting up the electronics to the base and work towards classifying the objects by size.