Individual Lab Report
February 16, 2017
Matt Lauer
Team E
Mike Beck, Akshay Bhagat, Leo Lu, Jin Zhu

Individual Progress

For this PR I focused on integrating the linear actuator into the planning software of the UR10. I started by designing and building a small circuit board that supplied power for the motor controller and level shifting so that an Arduino Uno could communicate with the motor controller, which took 24V logic. The board itself has proved fairly reliable. Figure 1 includes an image of the circuit board attached to the Arduino. At a glance, the board is fairly messy, but all relevant signals are labeled and there is a basic color coding scheme that has helped me keep things straight.

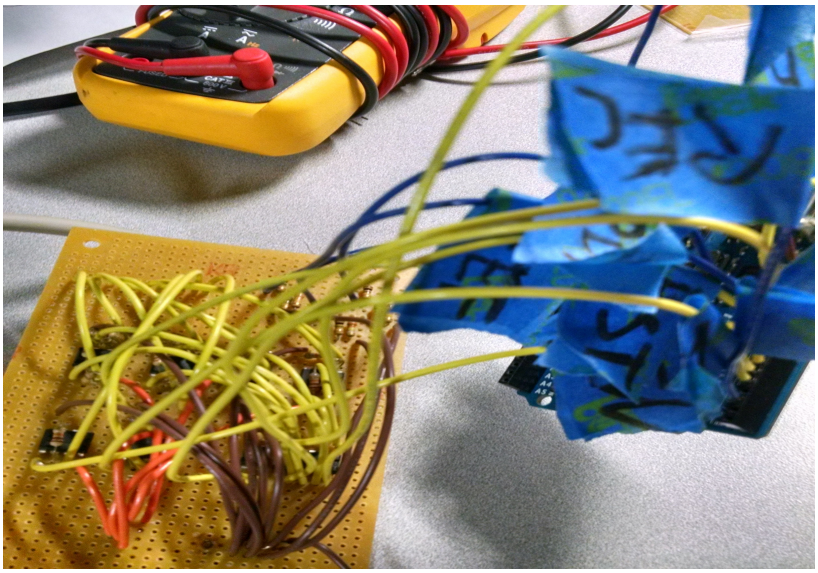

Figure 1. The completed circuit board connected to the Arduino.

Once the board was constructed I powered it up and measured the output voltage and to determine it would not fry the board. I wrote a very short program for the Uno to determine that signals were being sent and received correctly. I verified this manually by using a multimeter. Once that was complete I began writing a larger script that would allow the linear actuator to move to arbitrary positions at a resolution of 1mm. The feedback on the motor controller is spotty (or perhaps not fully understood), so the current implementation simply waits for a fairly large amount of times between moves so as not to accidentally make a movement command during motion.

I attempted to integrate the linear slider into the planning packages I was using, but I found quickly that the goal was beyond my reach for the time frame given. I decided to give control of the slider to the state machine of the robot, which would be a less fine control than the planners I was simulating with. I modified some scripts that published planning scene transforms to reflect the new way the linear actuator was handled.

I also worked on creating another iteration of the planning scene to reflect the new connect mounts and the minor deviations of the bins in the real world. Since then Mike and I taped floor positions of the bins so that we could easily determine when the bins were not in alignment.

After that I reviewed the old grasping code and to get it ready for the demonstration. I modified the way that centroid data was interpreted. Since the centroid is now reported in pixels I determined the pixel to cm conversion for each of the mounted Kinects. I also determined the transform of each Kinect so that the centroid estimation could be converted to a point in space. I made some new stored poses to help the arm with the new picking scene and verified an entire test run in simulation (gathering image data with Kinects, but no other hardware).

I continued to perform tests with all hardware enabled, but there were many communication issues due to the heavy load on the serial busses of the computer. We had purchased an additional bus in anticipation of this, but we had still underestimated the issue. After a series of incomplete hardware tests I believe that the system is viable, but the communication issue is completely crippling.

I also did some maintenance on the computers and installed some new hardware.

Challenges

One challenge I faced was getting the planners operational for the 6DOF arm. I had been planning in 7DOF for about a month and it seems that I altered something fundamental in the 6DOF code. I had only intend to use the base UR10 for a couple of days a little over a month ago, so I had difficulty tracking down the change that had rendered the code inoperable. I ultimately fixed this by reverting my universal robot package back to the default package produced by Universal Robotics. It was a pain tracking down all the decencies related to my homemade version, but I believe that was the ultimately easier solution.

The second challenge was the communication issue when all hardware was connected. This prevented our demo from working consistently. This issue remains unresolved and while we tried multiple fixes there were other small compounding problems that ate time or made possible solutions unworkable. These include the failure of our second computer and strange driver problems on the Kinect.

Personally my largest challenge was the amount of work I was given to complete. I was responsible for the entirety of the demo sans vision. I thought I would be receiving more help from my teammates with things that getting Kinect transforms or the resolution/characterization of known issues like the Kinect drivers, the master slave status of our PC's, or even just determing the Kinect transforms. Things like this should not have been my job, but it was too late to negotiate with people who had already left for home the night before the demo when I was made aware of these problems. This made me very unhappy. Mike is aware of this and gave me a few days off to get back up to speed with my other courses. Maybe this doesn't belong in an ILR, but sometimes I feel like some teammates don't care.

Teamwork

Jin has continued working on a neural net for image classification with faster RCNN as well as some work on localization.

Akshay made some wooden mounts for the Kinects.

Leo integrated the new vision code into the devel branch of plaid_apc. Leo modified the state machine and grapsing code to reflect the new changes in the vision code.

Mike helped me make minor adjustments to the physical robot workspace. He also helped me to debug the linear actuator code and

Future Work

I need to make control of the linear actuator more robust. Occasionally the node loses sync and currently it does not publish it position.

I would like to make modifications to the ur_driver so that I can plan with 7DOF in a more robust way.

I will make a plan to consolidate our electronics, which will only become more cumbersome as the shelf actuators are added in the coming months.

To contradict the previous ILR, I will not be working on grasping int the future.