Alex Brinkman

Team D: Project HARP (Human Assistive Robotic Picker)

Teammates: Abhishek Bhatia, Lekha Mohan, Feroze Naina, Abhishek Bhatia

ILR 10: Progress Report

Submitted 3/31/2016

1. Individual progress

The last two weeks I completed a lot of small tasks that built toward the single autonomous testing. I added a trajectory playback feature to the system that stores trajectory plans, completed the turntable to autonomously collect and segment images, and conducted the Failure-mode effect analysis on each subsystem.

The Trajectory playback feature records plans generated by any Moveit planner and stores the start configuration and end goal state. The c++ class saves the trajectories to a file for later retrieval. When we pass a new goal to the arm planner we first search the library to see if we have a cached plan and execute it before spending the time to generate a single query plan. We have a good understanding of the configuration space so we can safely do this without colliding with the environment or mounting frame. The trajectory recall is almost instant and saves 75% of the total execution time. Additionally, this feature allows us to mitigate several failure modes. For example, we can guarantee that any plan saved in the library does not cause collisions or tangle the hoses and wires attached to the end effector.

Next, I worked on the turntable system to autonomously segment and generate a large data set of objects and scenes. We obtained a green screen to help segment the background of the scene. I was unsure of the best way to segment the item out of the background and started by implementing a simple HSV threshold filter and created a simple GUI to manually tune the filter. The GUI can be seen in Figure 1. I found that discriminating the background based on HSV color was good enough to begin generating the first dataset. The results are shown in Figure 2. The full system took longer than I had anticipated to integrate the individual components and debug. The dynamixel powered turntable has a lot of inertia from the wooden disk used as the turntable platform that would cause a lot of overshoot and oscillation when trying to reach a commanded setpoint. Using a delay to allow the platform to settle but would require a long time to ensure the platform was not moving but since we need to acquire approximately 4000 images, this is undesired. Instead I read the error signal of the dynamixel and ensure both position and velocity error is below a small epsilon before continuing to capture images. This approach will also have problems with green objects which can be addressed by using a different background color and then retuning the HSV threshold filter.

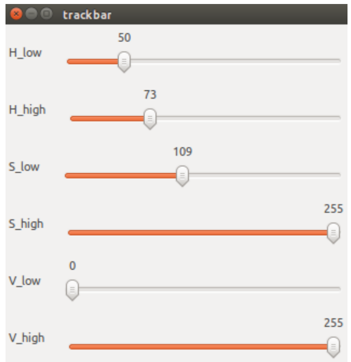

Figure 1: HSV Threshold GUI Showing Tuned Values

Figure 2: HSV Color Segmentation Result

Finally, I worked to mitigate all the known failure modes we had identified on our system to date. Table 1 enumerates the failure modes and how they are being mitigated.

Table 1: Identified Failure Modes

Subsystem	Failure Mode	Mitigation
Suction	Arduino Disconnects	Set respawn flag to reconnect to Arduino
Suction	Pressure sensor signal has failed	Look for erratic pressure signal and perform simplified pick and place
Hoses and Wiring	Hoses or wires pull on EE and cause arm to disable against safety stop	Limit robot joint limits to prevent FM Execute pre-computed trajectories that do not result in problematic plans
Arm Control	Arm impacts shelf or objects with sufficient force to disable arm	Execute pre-computed trajectories that do not result in problematic plans Reset after disable events
Arm Control	Executive continues to plan for excessively long time	executive logic monitors planning and limits max planning time
Perception	Empty Pointclouds cause server failure	Add a check for size zero point clouds
Perception	Kinect device goes down	Has not been shown to be a problem during runtime, only during development activities
Perception	Perception returns poor results	This problem is approach-dependent, on hold

Since we were having problems with false positives and negatives from the pressure sensor I started there. I revised our detection logic to use a state machine to accommodate debugging and better understand performance. The state flow first triggers when we command the vacuum on. The logic allows the pressure to rise before continuing. Next, the pressure is monitored for 30 samples to determine the mean and standard deviation of the pressure signal and then it calculates the threshold pressure values as -2.5 sigma of the measured standard deviation. The sigma band was tuned to be as sensitive as possible while ensuring false positives would not be seen. This approach works really well and should be robust to bias in the pressure sensor, environmental factors like elevation, and electrical noise.

One of the more difficult failure modes is restoring the arm after a disabling event. If the arm contacts the environment with moderate force the UR5 controller will disable the arm. To re-enable the arm, we have to disconnect the arm from ROS, re-enable the arm using the touchscreen interface, and then reconnect from ROS. I contacted Tom and asked him if there was a way to re-enable the arm from the computer and he informed me a socket called the Dashboard Server (socket 2999) that can clear popups and re-enable the arm. I created a simple script to communicate with the Dashboard Server and showed we can re-enable the arm from the computer. We have not made the reset part of the logic of the ROS software yet but will implement this feature before the SVE.

2. Teamwork

Rick and Abhishek have been working closely on perception tasks to create the trained neural nets and implement and train the classifiers. Feroze and I have been working toward the single-item autonomous demonstration, debugging problems in logic added features as needed. Lekha and Rick are working to find the best approach to the perception problem. Rick is working with Venkat from SBPL to integrate his PERCH perception algorithm.

3. Plans

My plans for the next few weeks are driven by the large amount of perception tasks we need to complete in the next few days. My main tasks will be the following:

- Create the image dataset on the turntable for the Identification CNN
- Create the image dataset on the Kiva shelf for the filtering CNN
- Help manually label the training data
- Evaluate stereo vision to address poor depth data from specular objects
- Get multi-object grasping demo working