Individual Lab Report 10

Nathaniel Chapman

Team B: Space Jockey

With Brian Boyle, Ardya Dipta Nandaviri, Songjie Zhong

April 4th, 2014

1 Introduction

The last two weeks have been very busy in the software side of our project, unfortunately, my work was fraught with difficulties, and we were not able to generate the results we wanted for this ILR. As more of our teammates begin to focus on software development, these complexities will continue to grow, but I believe we are still making good progress towards our spring system validation, and will be able to generate a compelling demo for our next review.

2 Individual Progress

Most of my efforts in the last two weeks have been in the software space. In an effort to bridge the gap between the waypoint planner previously demoed and the physical robot, I have been working on getting the kinematics, joint trajectory control, and motion primitive sub-planners working. As seen on Thursday, this effort was not as successful as I hoped, but I have learned a great deal and now have a much better idea what needs to happen to get all of our kinematics code working in tandem with our localization, image comparison and inspection tasks.

The first step in this process was refining and adjusting our robot's URDF model to match the physical layout of the robot as built. Brian created the initial URDF, but as I started to take over the inspection tasks, and he was working on the physical integration of the feet. I pulled the correct robot dimensions from Solidworks, and improved the layout of the file and joints to better support kinematics development. The updated model may be seen in Figure 1.

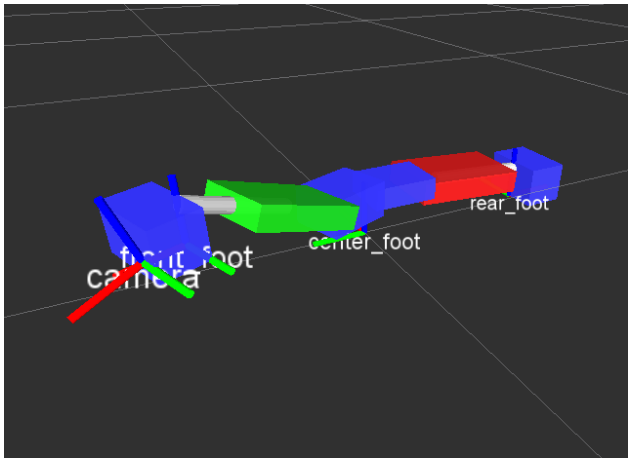

FIGURE 1 - UPDATED URDF MODEL SHOWN IN RVIZ

I have also been working on getting both our simulation and real life systems to integrate with Rviz for easier debugging and systems integrations. Towards this end I have developed several visualization tools for our planning process, and have

incorporated the built in joint state publisher and robot state publisher tools from ROS, to handle robot controls in a much more standardized way.

As more and more of our teammates are developing code projects in tandem, a large chunk of my time these weeks has been spent in performing code reviews and branch merges on our repository codebase. In particular, I spent some time modularizing Dipta's IMU ROS code and compiling it into our robots Arduino firmware. We now have this data available, and should be able to incorporate gear lash compensation into the robot kinematics system before the next progress review.

In addition, I also spent some time updating and rewiring the robot code so as to add support for our foot detachment mechanism. I have also started rolling more of the robot's hardware configuration onto the Computer side of the bridge, to simplify our servo calibration process, and simplify future development.

3 Challenges / Issues

Due to my previous focus on the hardware side of the project, these last two weeks have been my first big foray into our ROS code. In particular, when trying to figure out our inverse kinematics system for our robot, I spent a lot of time trying to figure out what already existed in ROS, and what I could use to solve the problem. In the end, I chose to write my own simple geometric solver (which works due to the low number of links in our robot), because that gives us the simplest, fastest (computation time) solution available, without relying on heavy-weight software packages like OpenRAVE, or ROS KDL library. The downside is that this added a large development task to my week, and I was unable to work all of the bugs out of the Kinematics engine before the demo (which was partially responsible for the 'twitching' problems shown with the robot on Thursday.

In addition we had several mechanical problems with the robot occur this week. One of our foot detachment servos overheated and burned out during testing, unfortunately, I was running the robot from the bench top power supply at the time, and bypassing the power distribution board (with its current filtering and fusing features), which may have propagated bad power spikes and brownouts to the rest of the robot. Around the same time, our onboard radio developed intermittent problems, additional testing is necessary to determine the root cause of the issue, but it may need to be repaired or replaced. I will be ordering spares for both of these parts in the next week, so we can be protected from similar failures during the spring validation experiment.

Finally, I have been having difficulties training my teammates on how to use git properly. As our codebase has grown in size and complexity, and we have had more teammates working in parallel, I had several occurrences where Songjie or Dipta clobbered the master branch with lots of broken code from their private branches, and have had to help them straighten it out. I have also been managing all of our merges and regression testing to protect the stability of the project. At this point, I think I have them pretty well trained now, but it has been a trying couple of weeks in that regard. On the plus side, I am now very good at using git's reversion and amended commit tools to repair accidental deletions or other repository problems.

4 Efforts by Team Members

For the last week, most of my team mates have been working on bringing our previous code parts into ROS for integration. Songjie was working on modifying the April tag and image comparison code he developed to take input and outputs from ROS topics, as ROS image data and tf coordinate transforms. Dipta has developed a URL camera bridge, so that we can receive image data from our Wi-Fi camera in the form of ROS images, which will help with localization and flaw detection for the next progress review. He was also responsible for the IMU prototype code development which I integrated into our firmware.

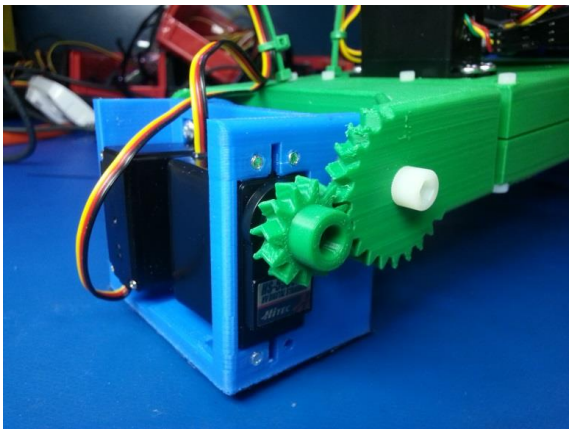

FIGURE 2 - ROBOT END FOOT DESIGN (PHOTO CREDIT: BRIAN BOYLE)

Both Dipta and Songjie spent a lot of time learning how to use the ROS CV bridge package, so we can integrate openCV functionality into our software well now. During these weeks, Brian has been focused on integrating the magnetic foot system into our chassis, and testing them. He did some work with the URDF rewrite as well, but wanted to focus on hardware for the time being, the new foot design he developed may be seen in Figure 2.

5 Future Plans

For the next progress review, I will be focused on rewriting the planning and motion control code I've been working on, as well as integrating my teammates work into our ROS ecosystem to produce a well-functioning system as a whole, as opposed to several very disjoint systems as were demoed in the progress review. I also will be focusing on adding more debugging and calibration tools, so that it is easier to test and verify proper system functionality as our complexity grows.