Individual Lab Report
Weekly Progress

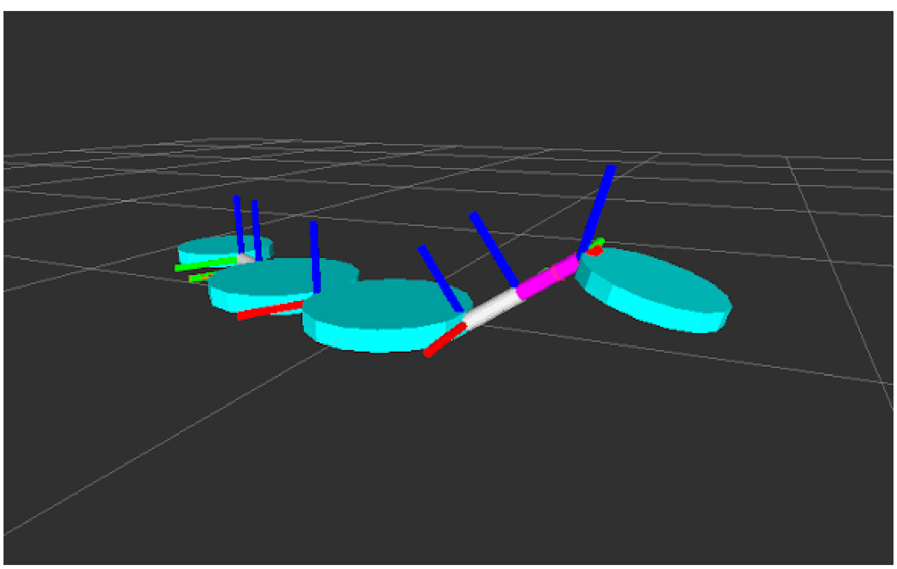
Brian Boyle
Team B: Team Space Jockey
Teammates: Nathaniel Chapman, Ardya Dipta Nandaviri, Songjie Zhong

ILR04
November 5, 2013

1. Individual Progress

My progress during this period was primarily in the integration of ROS into our software suite. The main guide towards this end was our GUI, which acted as a framework for the exact functionalities we desired. As it turned out, many of the book-keeping functionalities we desired are already integrated into RQT, a package for integrating QT-based GUI's into Ros, in the form of plugins. For instance, monitoring debug/info/warning/etc. messages is already handled by the Console plugin, and recording and playback of a ROS state over time can be handled by the Bag plugin. Other plugins, such as Rosgraph and Ros Logger Level, allow us to monitor the ROS system at a high level so we can confirm that nodes are communicating as intended. Many of these functions we'd planned to implement in our custom GUI, but now can be accomplished using these pre-built tools.

In addition to identifying and familiarizing myself with these plugins, I was able to adapt our GUI into an RQT plugin that could run within the ROS environment. Once running in RQT, I was able to attach event handlers to GUI widgets and publish joint position commands to the topic used by RViz to pose the robot model loaded into it. For this functionality, I also implemented a URDF description file of our robot using estimated link lengths. Since RViz is a visualization rather than simulation tool, there was no need for more in-depth information such as mass or inertial qualities. As such, we now have a visualized kinematic model of our robot that we can modify using the custom GUI (See Figure 1). If we later need to build a full simulation in Gazebo once we have the robot fabricated and assembled, the same URDF file can be modified and loaded for that purpose as well.



(Figure 1: RViz model of the Space Jockey Robot)

2. Challenges/Issues

The greatest challenge of this period was simply getting up and running with ROS, RQT, and Rviz. Many interconnected concepts have to be handled with competence at once to get anything accomplished, and learning all of those at once is daunting. Documentation quality is mixed, sometimes very helpful but often outdated, non-existent, or incorrect. While ROS is capable of running both python and C++ modules, many tutorials and documentation apply to one or the other.

In the case of our RQT plugin, tutorials highly recommended using python, but the python-based tutorial seemingly had errors that prevented compilation. I ultimately had to copy and adapt an existing RQT plugin, and even then had to scour forums to discover a little-known cache issue that prevented RQT from recognizing new plugins before my custom GUI plugin could be made functional. That being done, there's seemingly no documentation at all about implementing an Rviz window in a python plugin, only C++.

Similarly, in creating an RViz visualization of our robot, tutorials on the web were not useful and typically assumed I was using a common robot such as the PR2 for which URDF files already existed. Ultimately I had to work from a series of URDF tutorial packages and refactor the code and associated files for our own robot, then trim out the unnecessary functions that were interfering with our custom GUI.

Despite the slow start with ROS, however, I've learned a lot over the past two weeks and feel a lot more competent going forward. Now competent with the basics of using ROS, I'm now faced with the challenge of deciding the best way to divide our desired functions into nodes, and how to structure their intercommunication to fulfill our desired functions.

3. Team Work

The rest of the team's work during this period was towards finalizing our prototype design. Dipta worked with Nate's assistance to finalize our power distribution board and send it in for fabrication. Songjie worked independently to further refine our sticky-foot design and 3D printed one with an attached rigging that we can hang weights from for testing and has been continuously designing our chassis. Simply experimenting with this foot by hand was very encouraging, so much so that I'm now somewhat more concerned about mustering the forces necessary to detach the foot more than wheter or not the dry-adhesive will support the robot's weight.

To facilitate bench testing, Nate also designed a simple board to mount an arduino and two servo motors to. He and I both visited Larry in the Collaborative Machining Shop to laser-print this on Medium Density Fiberboard I acquired from Home Depot, and are both now aware of how to use the laser-cutter for future fabrication needs. While the board in itself won’t be used in the prototype, it was useful as a case study in laser-cutting and will also help with bench testing. Nate also placed what should be our final parts order for the semester to complete the prototype, and has designed a linear actuator that we can build which should be considerably lighter than commercially-available models.

4. Future Work

For our next progress review, I’d like to get as far along as possible to implementing full motions in our robot. Now that I can send commands for single poses, it’s now a matter of finding ways to compose “step” motions for each segment and sequencing intermediate poses into a joint trajectory that can be followed by the prototype. This can be explored in the RViz visualization, and I expect the actual motions involved shouldn’t be terribly complex given the simple kinematic design of our robot. The primary challenge will likely be in moving the center segment, requiring the synchronized motions of at least six servo motors.

The single most pressing issue in our project right now is the fabrication and assembly of the prototype robot. I’ll be diverting my efforts from software as necessary to assist with this however I can, because the usefulness of the software I can develop is limited by not having a physical prototype to interface and test with. To that end, I’ll be volunteering to do as much fabrication and assembly as I can, to free up Nate and Songjie to fabricate and assemble the linear actuators and sticky feet, respectively.