Indoor Navigation and Manipulation using a Segway RMP Platform

Christopher Dunkers (RBE), Brian Hetherman (RBE)
Paul Monahan (RBE), Samuel Naseef (RBE/ME)
Advisor: Gregory Fischer (RBE/ME), Dmitry Berenson (RBE/CS)

Abstract

The goal of this project was to work with a Segway RMP, utilizing it in an assistive-technology manner. This encompassed navigation and manipulation aspects of robotics. First, background research was conducted to develop a blueprint for the robot. The hardware, software, and configuration of the given RMP was updated, and a robotic arm was designed to extend the platform's capabilities. The robot was programmed to accomplish semi-autonomous multi-floor navigation through the use of the navigation stack in ROS (Robot Operating System), image detection, and a user interface. The robot can navigate through the hallways of the building using the elevator to travel between floors. The robotic arm was designed to accomplish basic tasks, such as pressing a button and picking an object up off of a table. The Segway RMP is designed to be utilized and expanded upon as a robotics research platform.

Project Goals/Objectives

- Autonomous navigation in a pre-defined map
- Multi-floor navigation using the elevator
- Allow the user to define a goal floor and pose
- Design and implement a mechanical arm
- Tele-operated control of the arm
- Arm able to hit elevator buttons and pick up small objects
- Tele-presence capabilities in a multi-floored building

Platform

- Updated the Segway to the newest firmware as well as the newest layout. This allowed better communication with Segway
- Designed a power distribution PCB to organize the electronics on the platform.

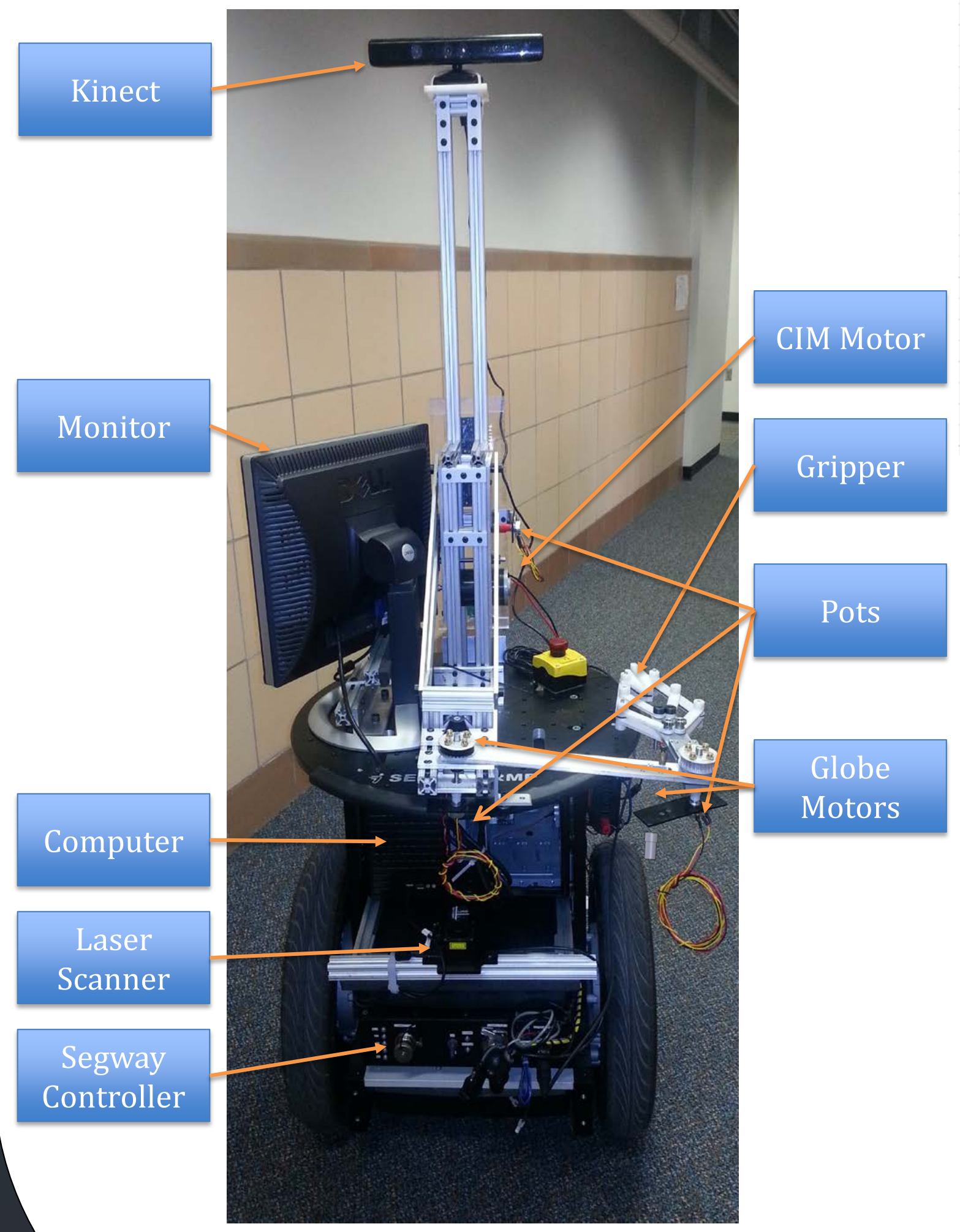

Figure 1: Final Robot

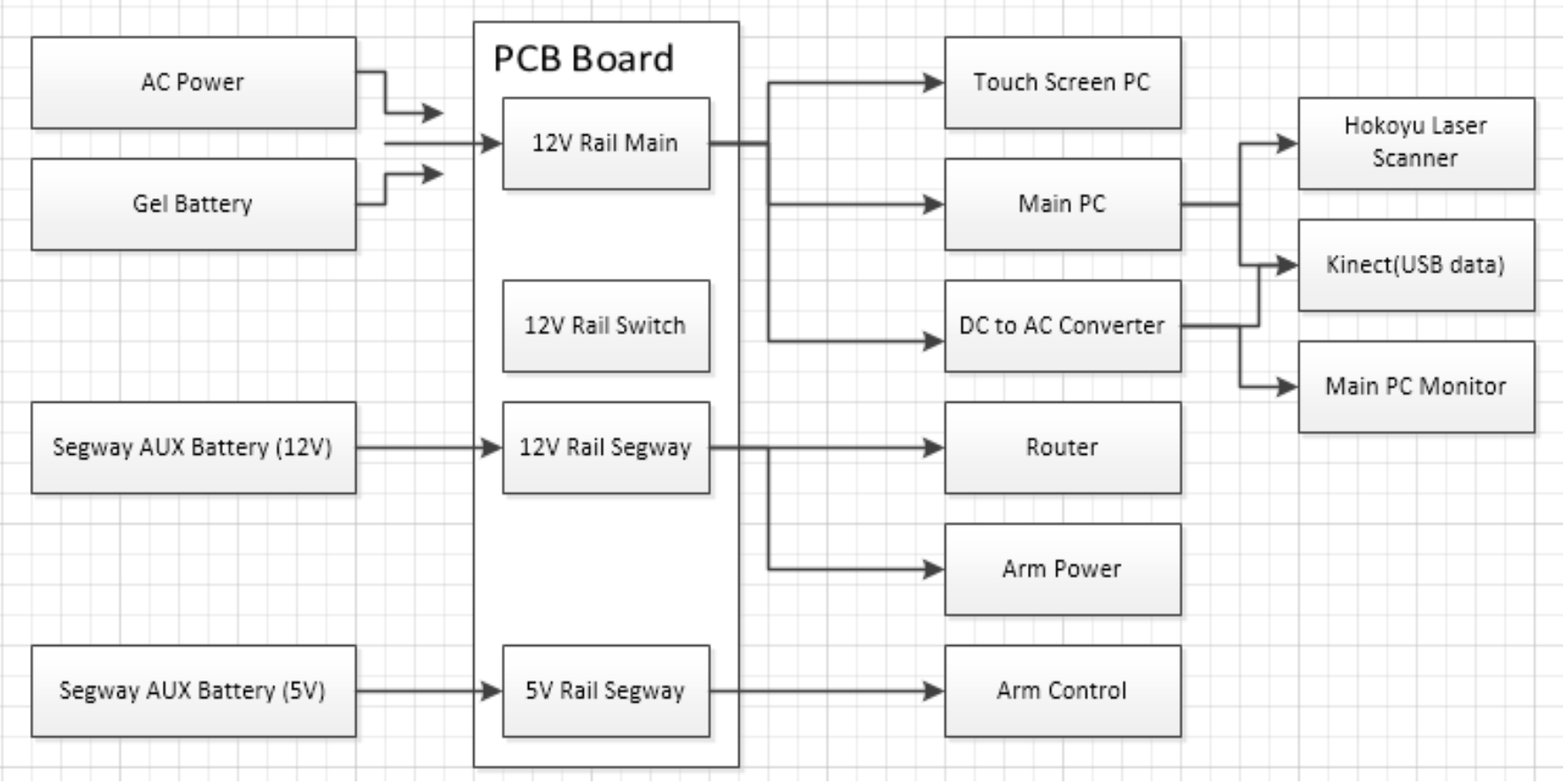

Figure 2: Electrical Schematic

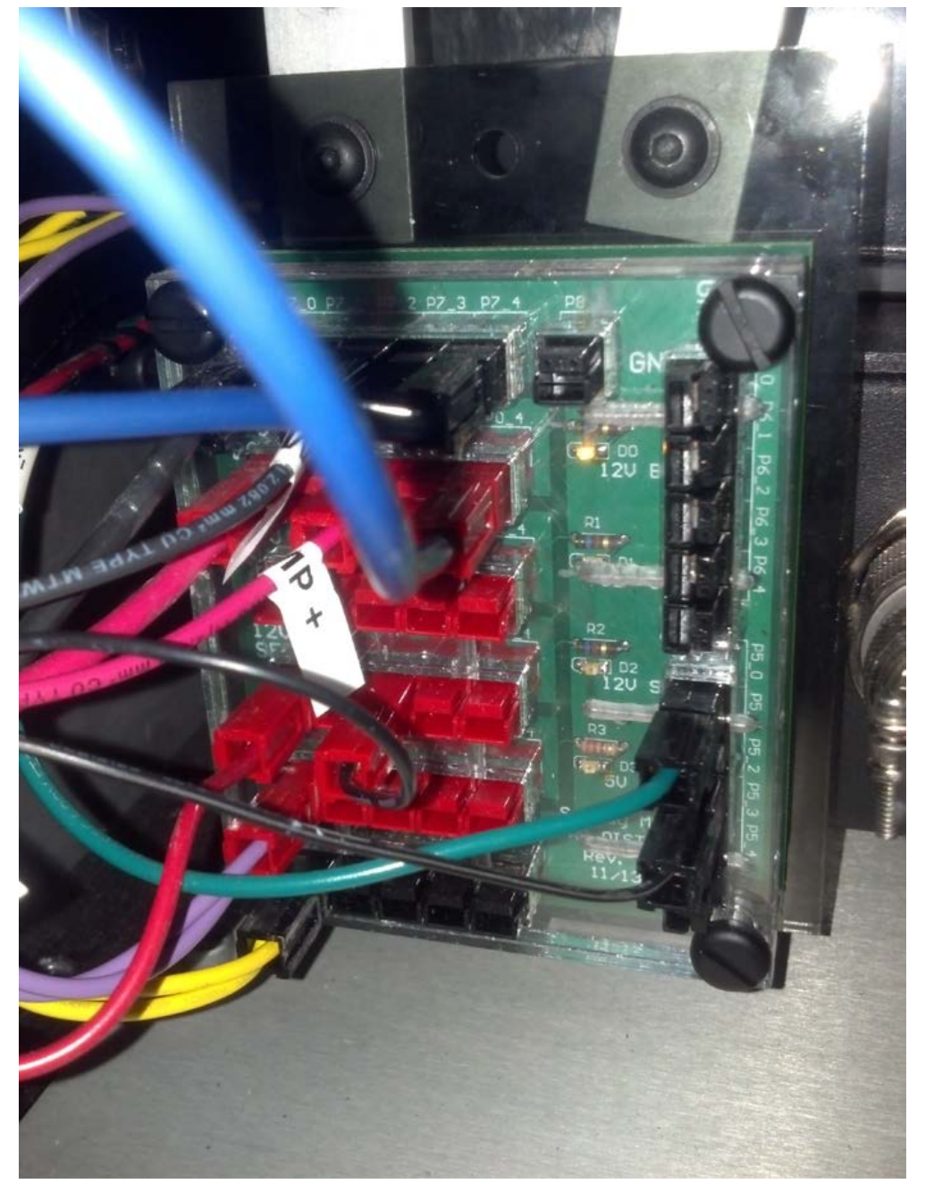

Figure 3: Power distribution PCB

Navigation

- Implemented ROS packages to enable autonomous navigation using a laser scanner, odometry from SegwayRMP and Kinect.

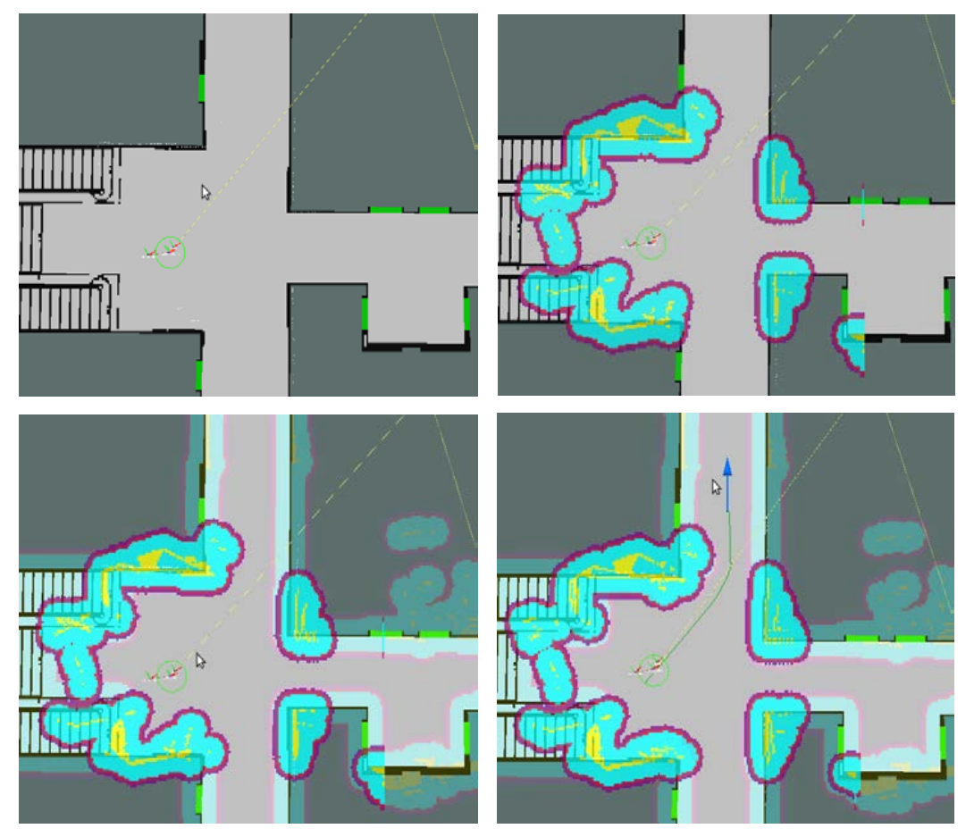

Figure 4: Robot Navigation and Maps in Rviz

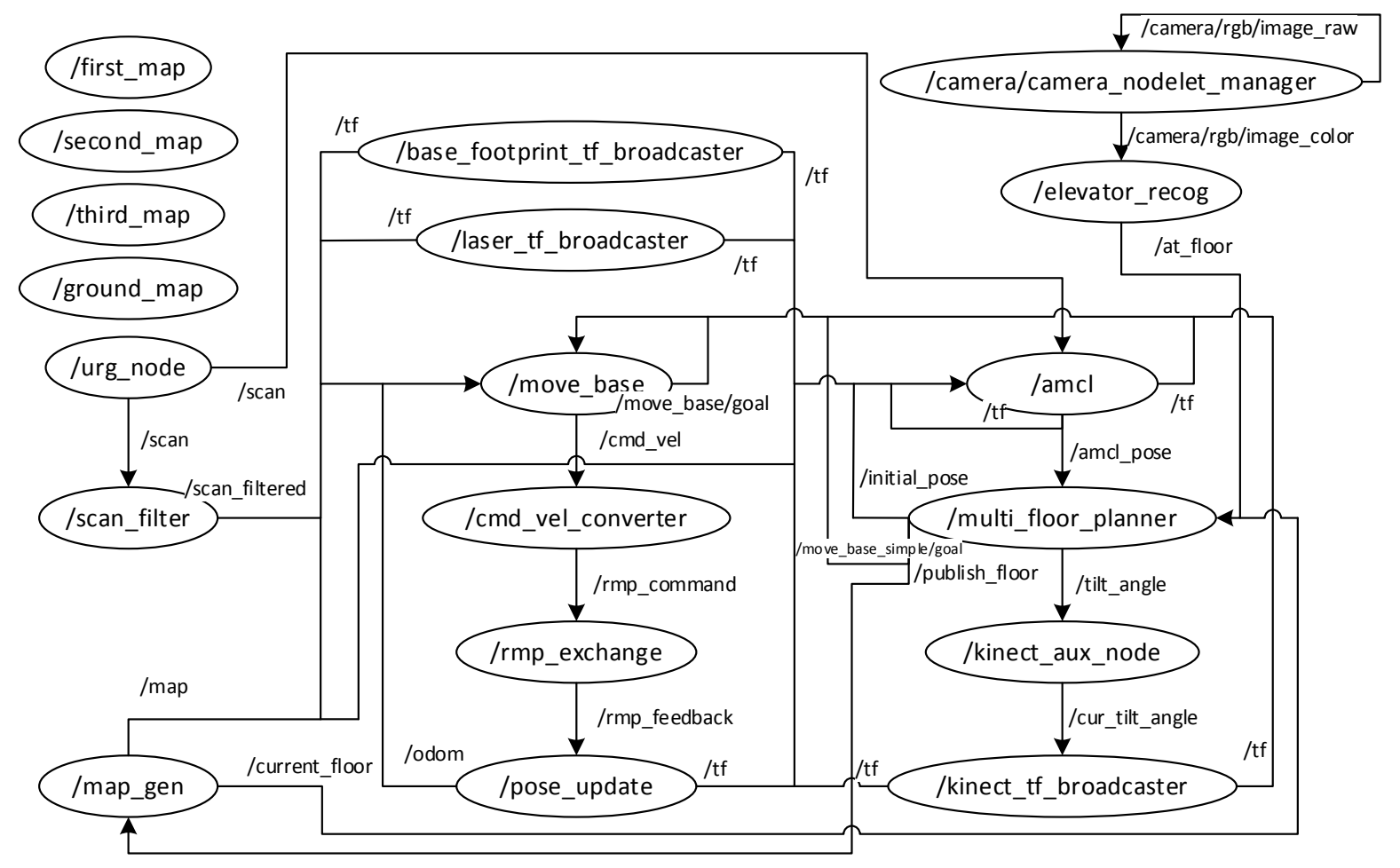

Figure 5: ROS Schematic

- Developed a user interface to set goals on any floor. This interacted with a multi-floor planner to execute the aggregate steps.

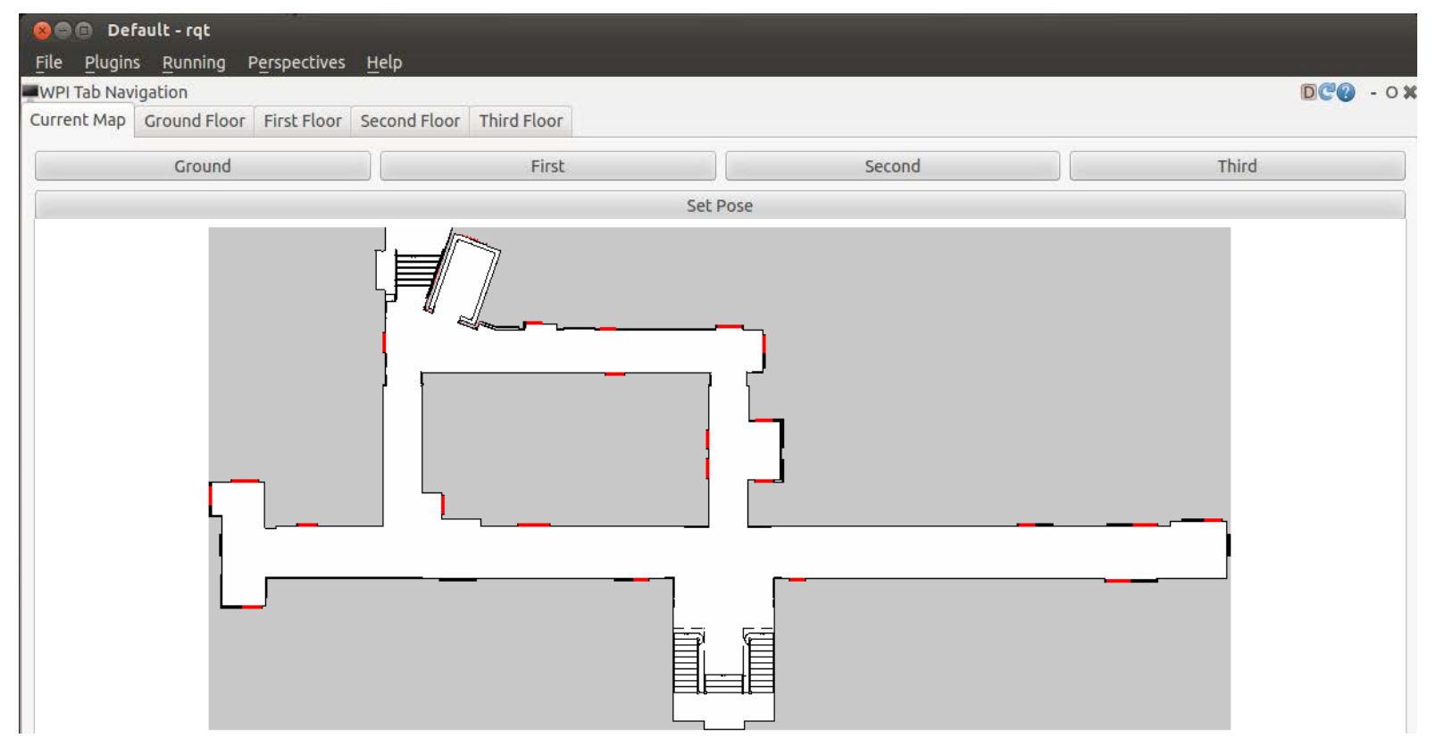

Figure 6: RQT Plugin developed

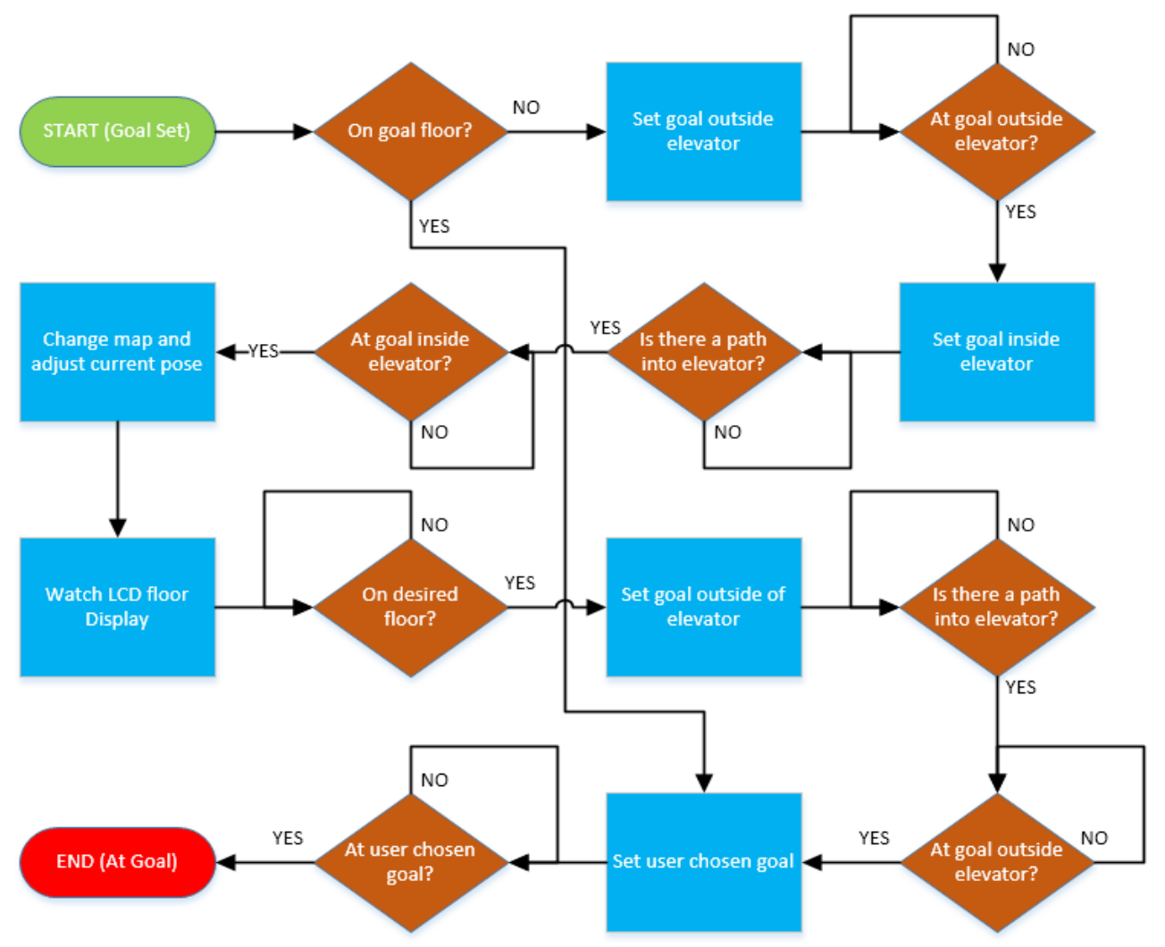
START (Goal Set)
On goal floor?
NO
YES
Set goal outside elevator
At goal outside elevator?
Set goal inside elevator
Is there a path into elevator?
At goal inside elevator?
Change map and adjust current pose
Watch LCD floor Display
On desired floor?
Set goal outside of elevator
Is there a path into elevator?
At goal outside elevator?
Set user chosen goal
At user chosen goal?
END (At Goal)

Figure 7: Multi-Floor Planner Flow Diagram

- Used image matching to identify floor before exiting elevator.
- Developed a map generator to read in bitmaps and publish occupancy grids of the current floor.

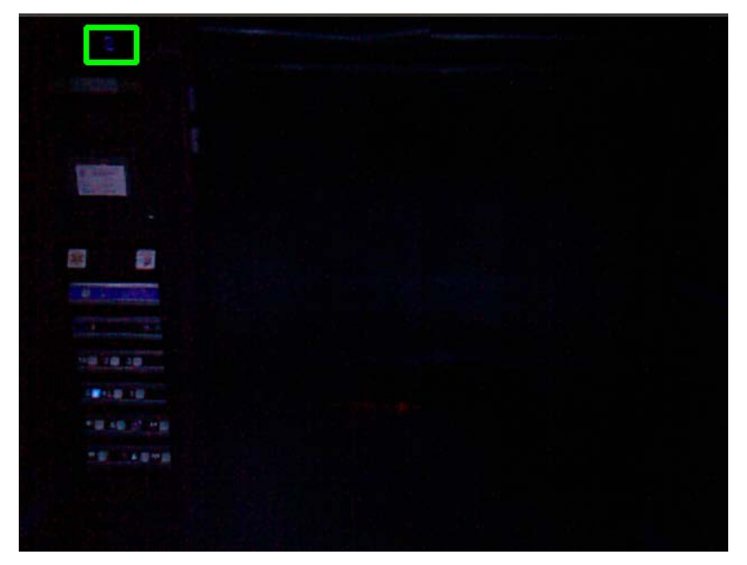

Figure 8: Image Matching in the Elevator

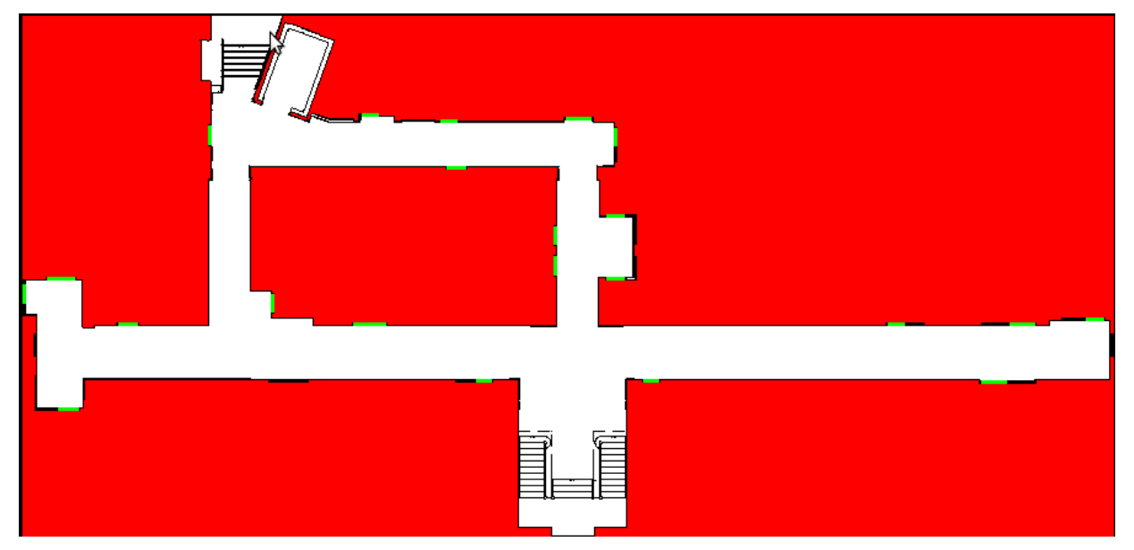

Figure 9: Initial Floor Plan Bitmap

Arm

Goals:

- 5 lbs objects the size of a Nalgene water bottle
- 30" height range; 24" horizontal reach
- Press 1" by 1" square buttons

Design:

- 4-bar linkage, 2-link arm, 1-DOF gripper
- SparkFun/Polulu motor drivers
- Microchip PIC32 microcontroller

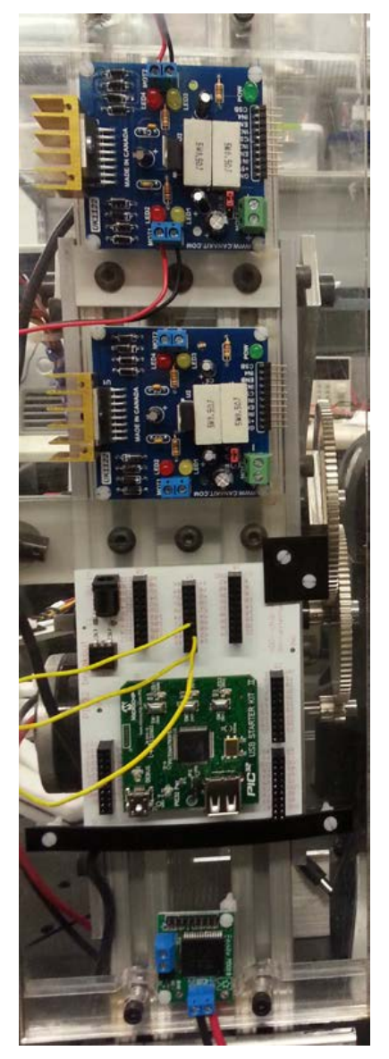

Figure 12: Motor drivers & PIC32

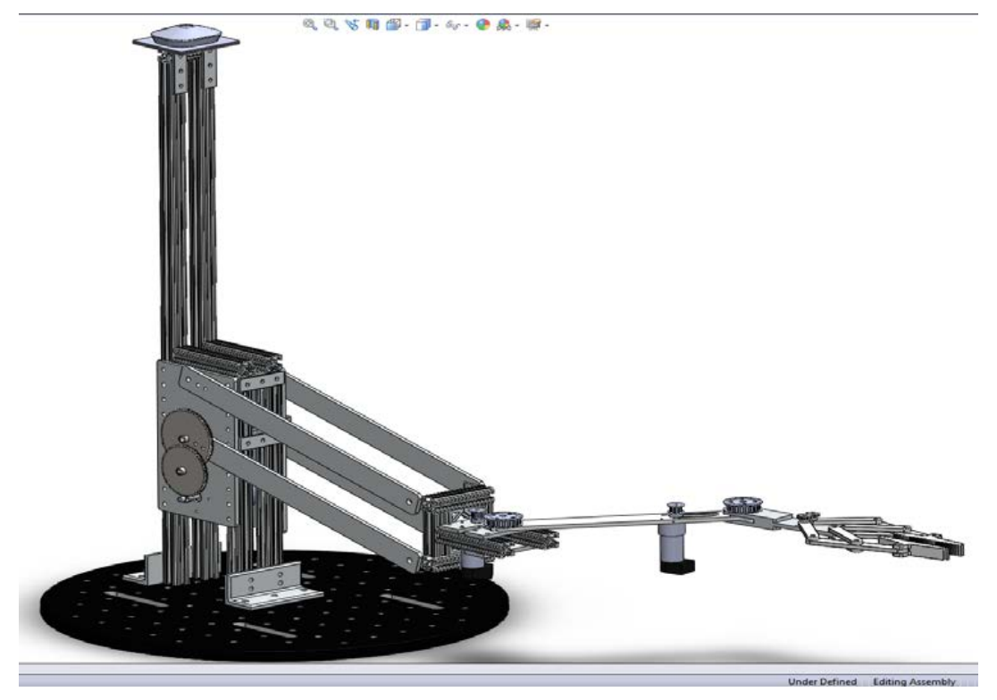

Figure 10: SolidWorks model of arm

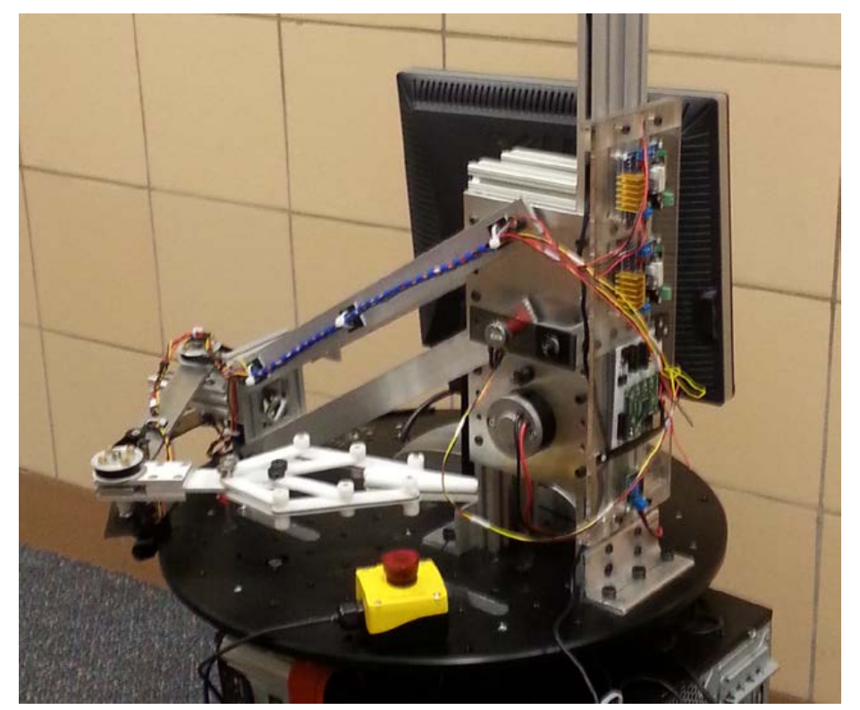

Figure 11: Constructed arm

Results/Outcomes

- The robot was reconfigured to allow easy modifications/updates.
- Navigation of a multi-floor building was successful, with the assistance of a person to press the elevator buttons.
- The running processes were not as reliable as desired due to the fact that the computer could not handle the processing load.
- Basic object detection was implemented using template matching to accurately determine which floor the elevator was currently on.
- Delivered ROS software packages to Segway Inc. that created an interface to the RMP controller.
- A mechanical arm was designed, built, and mounted to the robot.

Recommendations

- **More processing power:** Although the navigation functioned, it was not as consistent as desired. The localization did not happen at a high enough rate du to limiting issues of the CPU . Therefore, our first recommendation is to update the computer, as the four CPU cores were running at full capacity during navigation.
- **Distribute processing over wired ethernet or faster Wi-Fi connection:** When trying to distribute the processing load between laptops and the onboard computer, it was determined that the wireless speed was the limiting factor. Our second recommendation is therefore to use a wired connection or upgrade the router to support faster wireless speeds.
- **Multi-Threaded communication with Segway RMP:** Segway provided feedback to our method of communicating, suggesting that a threaded communication protocol with the Segway RMP could be used.
- **Full implementation of the arm:** Control of the arm was not able to be fully integrated into the ROS architecture.
- **Redesign of the gripper:** To support greater loads, we recommend the gripper be redesigned and constructed from a stronger material.

Acknowledgments

Chris Crimmins and Segway for the donation and continued support of the Segway RMP.
Professor Gregory Fischer for being an advisor and donating AIM Lab space and resources to our project.
Joe St. Germain for his continued support and advice, and for lending the LIDAR sensor to our project.

References

Segway RMP - http://rmp.segway.com/
ROS Hydro - http://wiki.ros.org/hydro/
Microchip - http://www.microchip.com/