ILR #1: Sensor and Motors Lab

Adam Yabroudi- Team A

October 15, 2015

Teammates: Tushar Agrawal, Pratik Chatrath, Sean Bryan

Individual Progress

For the Sensors and Motor lab, I was responsible for the integration of the servo and IR sensor as well as the electrical integration of all the components for the final assembly.

Servo and IR Sensor

As a group it was decided early on that we would be using an Arduino as our microcontroller. I was placed on the IR sensor because it is a sensor we are considering for our obstacle avoidance system for our fixed-wing VTOL aircraft.

In terms of writing code, there were Arduino libraries that existed for both the IR sensor and the servo. I did have to fine tune parameters on the IR sensor, though, so as to filter out a lot of the noise. My final calculations ended up using 40 measurements and rejecting consecutive measurements that were not within +/- 5% of the previous measurement. I also ignored measurements that were outside the band of 20 cm to 80 cm. Sensor readings that were too close the device or too far away from the device were super noisy and thus had to be clipped. After filtering the IR input heavily, I then mapped the inputs to the full range of servo outputs (0-180).

While I was writing the code, I also was very sure to make the code modular so as to easily integrate with the rest of the code base. Constants and pins were defined in global variables and code was structured in simple to understand blocks. This allowed Tushar to easily plug it into the code that communicated with the GUI.

Integrating the Electronics

My second major task was to integrate all the electronics. This included 3 separate systems with 3 motors (servo, DC, and stepper) and 3 sensors (ultrasound, IR, and potentiometer).

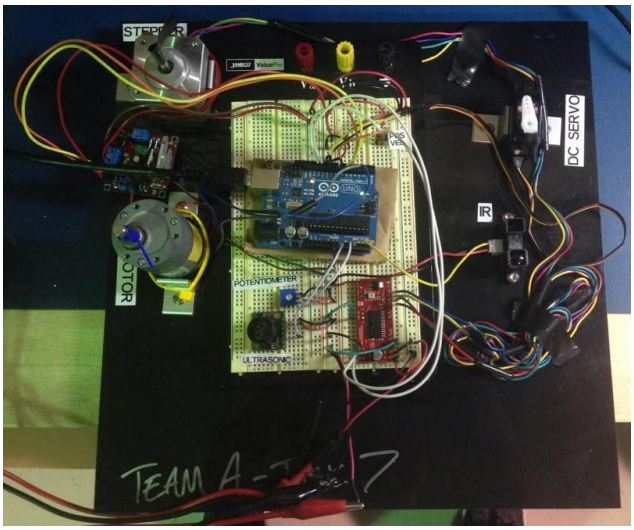

Figure 1: Image of the final integrated system

Figure 1 shows the final system. The stepper and DC motor were placed on the left with the servo motor on the right. The IR sensor is labeled on the right side of the breadboard with the ultrasound and potentiometer labeled on the bottom left corner of the breadboard.

The breadboard was arranged to help with wiring although it is hard to tell. Originally the Polulu stepper driver was placed on the top right corner of the breadboard closer to where the stepper motor was. That had to be altered in the end because the driver didn't work and needed to be changed for the Easy Driver from SparkFun. Since it was larger, the location had to change and wiring became more messy. Also because we didn't want to alter the lengths of wires from the motors and sensors, those wires were generally long and messy looking as well.

The breadboard was generally organized based on voltages as well. The right most rails were for 12V (to power the motors) while the central lines had 5V to power the sensors and drivers. As a result of issues with the Polulu stepper driver, we ended up making the 5V powered off the power supply and not the Arduino to eliminate scenarios of there not being enough current for everything on the bus. Eventually after some debugging, we also added common grounds.

Challenges:

Stepper driver

One of the major issues we faced when integrating all the electronics was with the stepper driver. The stepper driver had worked on Sean's individual board but failed when integrated into the whole circuit. Tushar and I had to debug to see if the issue was code or electronics and we soon realized it was electronics. The driver was really finicky and would sometimes turn on but was not reliable so after a few attempts we decided to make the switch and convert to the Easy Driver.

The Easy Driver was controlled by the same code and just required understanding the new pin layout. The biggest difference between the boards was the addition of the MS1 and MS2 signals which needed to be grounded for Full steps. Originally I had left them floating which is equivalent to $1/8^{th}$ steps and we couldn't figure out why the stepper was moving, but not the complete range. The solution was to ground the two pins which results in full step rotation.

Also while debugging the stepper driver, we realized that the Arduino might not have been supplying enough current for both motor drivers and the sensors. To be extra safe we used the power supply to power the 5V bus as well as the 12V bus. By doing this, we introduced a bug which was that we didn't have common ground now between the Arduino and the sensors. This took some more debugging but eventually we fixed that as well and the system worked.

Teamwork:

The project was split among the four teammates. Sean handled the stepper motor with the potentiometer as well as the mechanical integration of the system. Tushar worked on the DC motor and the ultrasound sensor as well as the integration of all the Arduino code. Pratik worked on the GUI.

In terms of working as a group, we each had to make sure our individually developed code would work with the larger system. The included making sure the ports and pins used on our individual boards wouldn't conflict in the larger system and code base. Also Sean and I

worked together to plan the layout of the system to try and minimize wire lengths and make the design look as clean as possible. Pratik also made sure that we all were able to output sensor and motor variable for easy interfacing from the GUI.

Future Plans

Going forward, we are going to divide up again and try to tackle separate projects. Pratik will be working on the obstacle avoidance system and working to see how the IR and ultrasonic sensors work when in arrays and see at what distances and angles they start to interfere with each other. He will also test these sensors in different settings (outside in sunlight or near a moving motor) to see if that affects noise in the sensor readings.

Sean will help Pratik out in terms of building a model of the aircraft to help test these sensors all over the vehicle at different distances and angles. Sean will also be working with BirdsEyeView Aerobotics to see if we can get the vehicle donated for free as well as working on testing the UGV and seeing if he can get it up and running. He will also work on the website and work with me on getting the gripper ordered and constructed.

Tushar will be working mostly on the vision system and setting up the BeagleBoard-xM. He will get it working with ROS and OpenCV and make sure it is set up with the proper drivers and images for future development.

I will be working on getting the Hokuyo Lidar up and running and will be doing this on a Raspberry Pi. The rationale by the RPI is that it will force us to set up another Linux board with ROS in case the BeagleBoard-xM proves difficult to develop on. It also allows Tushar and me to work in parallel and focus on our respective tasks. Once the Lidar is running, I will add the Lidar to the tests that Pratik is running to again confirm if different settings and parameters affect sensor readings.