Progress Review 4

Individual lab report – 05 || November 24, 2015

Team Daedalus

Pranav Maheshwari

Team Members:

Mohak Bhardwaj

Dorothy Kirlew

Richa Varma

Shivam Gautam

1. Individual Progress

- ❖ Design the mount for IR Proximity sensors
- ❖ Create Decision Unit node responsible for processing and transferring data between various subsystems

Mounts for IR Proximity Sensors

Our platform will have three IR Proximity sensors mounted at front to detect close range obstacles and avoid collision with them. Mounting spots can be seen in Figure 1. Isometric and bottom view of current design of the mount can be seen in Figure 2. The final 3D printed mount can be seen in Figure 3. The mount has a slot in it to let wires pass easily through it. During initial testing, the mount will be attached to the platform using a temporary adhesive but finally it'll be attached to the platform via two 3mm screws. This will require us to drill through the base plate of our platform as well.

These sensors shall be placed in such a way that they are able to detect the presence of an obstacle of width 10 cms within a range of 20 cms from the platform. Once all the mounts are ready, the obstacle detection subsystem will be extensively tested. If required, number of sensors might be increased to increase reliability.

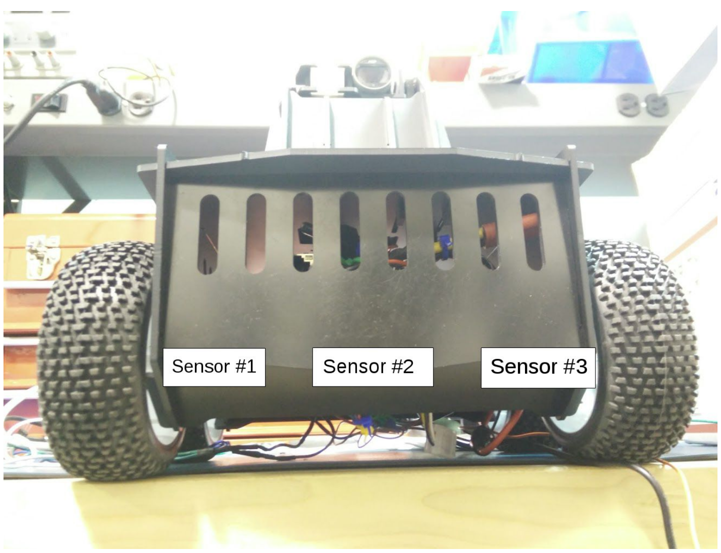

Figure 1: Mounting spots for the IR proximity sensors on Oculus Prime

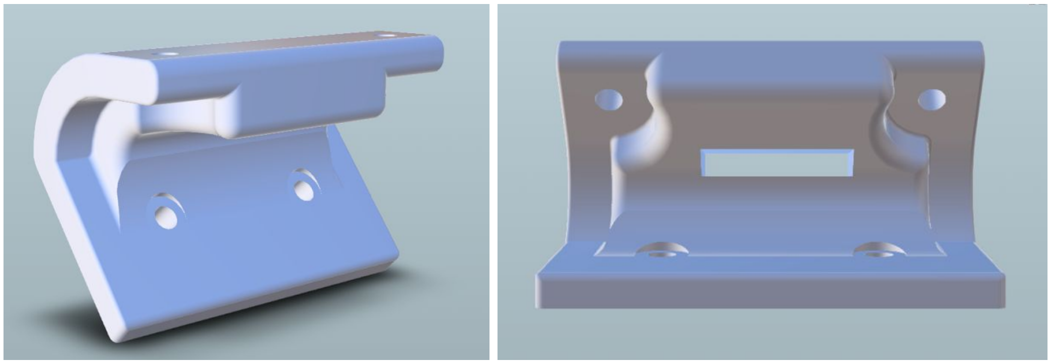

Figure 2: Isometric and Bottom view of the IR sensor mount

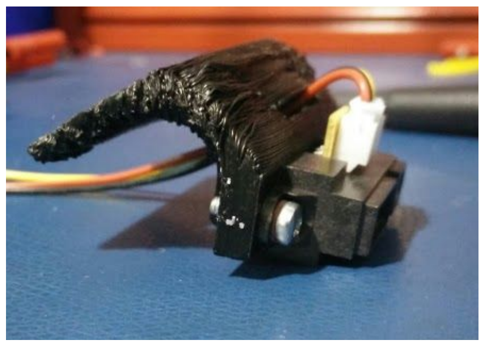

Figure 3: Final 3D printed product

Decision Unit central node

As can be seen in Figure 3, central node will be responsible for managing all the data flowing between various subsystems. It is important to keep a track of current system status, user requests, obstacles in the environment, etc., and central node will be responsible for fetching, processing and take decisions based on all of this data. Commands for locomotion would also be issued by this node.

As per the current routine, decision unit gets triggered by a user command sent via the Android Application. This command gets relayed to the decision unit through the Mobile App node. If a command to park is received, the platform sends a request to nearby platforms, asking for relevant data. This takes place through the collaboration node. Once this data is received, it gets processed and final destination for the platform is calculated. This is then transmitted to the Locomotion node as a waypoint. The locomotion node is connected to the OculusPrime server and is able to control the platform. The locomotion node also keeps a track of whether or not the platform has reached it's destination and keeps on publishing this data back to the decision unit so that it can be sent back to the Android phone.

In parallel to all this, decision unit has a thread running in parallel which accepts data related to the presence of obstacles so that the platform can be stopped if a close range obstacle is detected.

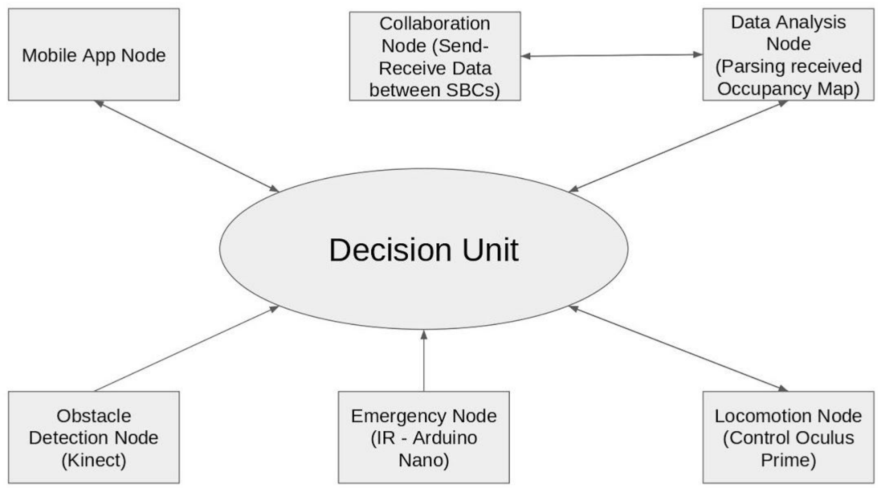

Figure 3: High level subsystem integration architecture

Another interesting aspect of working on the decision unit was using custom ROS messages. Apart from standard message types such as float, int or string which come with standard ROS installation, we can create custom messages which have a structure that suits our application. This helps in avoiding multiple pipelines for different data types. For example, while sending a command for locomotion, we can have a ROS message that has different fields to store multiple waypoints, target velocity etc. This helps in decreasing overall complexity of the internal communication architecture.

2. Challenges

The main challenge with integration lies in being involved with everyone's work. To ensure smooth integration of all the subsystems, it is necessary for me to have a good understanding of what all specific inputs and outputs others will have from their piece of code. My "Decision Unit" needs to be robust enough to handle all of these different messages being published at varied rates and ensure that things stay synchronized. This requires a lot of testing.

Another challenge I faced was interfacing with OculusPrime through my scripts. The documentation provide by Xaxxon for OculusPrime python package is scant in detail. It required some digging around for me to get an understanding of how OculusPrime server works and what sort of commands, state variables and message structure are required to interact with it.

3. Teamwork

As per the tasks distributed by project manager Shivam, everyone had a dedicated subsystem to work on with me being responsible to integrate it all. Richa worked with Shivam to finalize the definition of serial packets and to create a ROS node which sends, receives, and parses all of the information between multiple mobile platforms. This subsystem plays critical role in implementing the collaborative features of our Software Architecture. I assisted her in implementing this in ROS + Python. Mohak and Shivam worked on obstacle detection via Kinect on a laptop. After conducting some tests, Mohak worked on setting up the Kinect environment on Single Board Computer so that all their algorithms can be smoothly transferred. Mohak also setup the SBC to run OculusPrime server which is necessary to interact with the platform. Dorothy worked on the Android Application and the ROS node needed to send/ receive data via Bluetooth between the phone and SBC. Shivam worked on the emergency node which receives data from three IR Proximity sensors via Arduino Nano. This node helps the mobile platform detect close range obstacles and avoid collisions.

4. Plans

The main task ahead for the team is to rigorously test all subsystems and finally the complete integrated system. As we work on testing, we might have to iterate over some of the subsystems to fine tune their performance. Our schedule allows us to do this. Apart from this, one small task that is remaining is that of integrating the PCB with our mobile platform. This PCB will have an Arduino Nano and three IR sensors mounted on it. This task will be mainly handled by Shivam and Richa. Once our baseline goal of achieving 1-D locomotion is completed, Mohak and I will try and spend some time on implementing 2D locomotion. Dorothy, as the risk manager, will actively keep a track of our testing efforts to ensure that nothing gets overlooked and all the components of the system are working well.