

Week 2: Website Check, Media, Mover_Node

Individual Lab Report #2

Job Bedford

Team C: Column Robotics

Eric Sjoberg, Rohan Thakker, Cole Gulino

ILR2
10/23/15

Individual Progress:

My main responsibility for this past week was the generation of media and models as well as design and development of the mover_node for our software architecture. I also contributed to the system engineering work of the risk management and the work breakdown structure for the project.

Mover Node:

For the first leg of our project we utilized the ardrone as test bed for our high level code and algorithms. The Ardrone is cheap and ready on hand in the lab. There exist a Ardrone_autonomy library for ros, that enable to the drone to be control via ros over wireless communication. In our system's software architecture, as seen below, we have the mover node be the last leg of communication between our software and the drone. The purpose for this is eventually the state estimate and planer will compose a set of cases and state of the drone to achieve and implement. The mover node takes this information and concatenates it into a sequence of moving commands to send to the drone.

The mover node is the ros node that commands and directs the drone via the ardrone driver. The mover node strictly publishes velocity vector and desired yaw to the defined topics for the ardrone driver. Most of this past week was spent refreshing up on ROS, relearning the proper procedures and implementations. I am disappointed to say I forgot more than I would have liked but more so I didn't know as much as I thought I did about ROS. So, again, most of this week was a learning experience for me.

The Ardrone driver is operational. The mover node has been started, but has yet to compile and test on the actual drone.

Media:

Another responsibility I was tasked this week was the generation of media for the user story, website, visual systems tasks models. For the models I design and modeled a rough proposed environment for our quadroto. The environment is an spacious room with some trivial obstacles, an initial starting pad, and a mock underwater well-head with a docking station, as seen in figure 1.

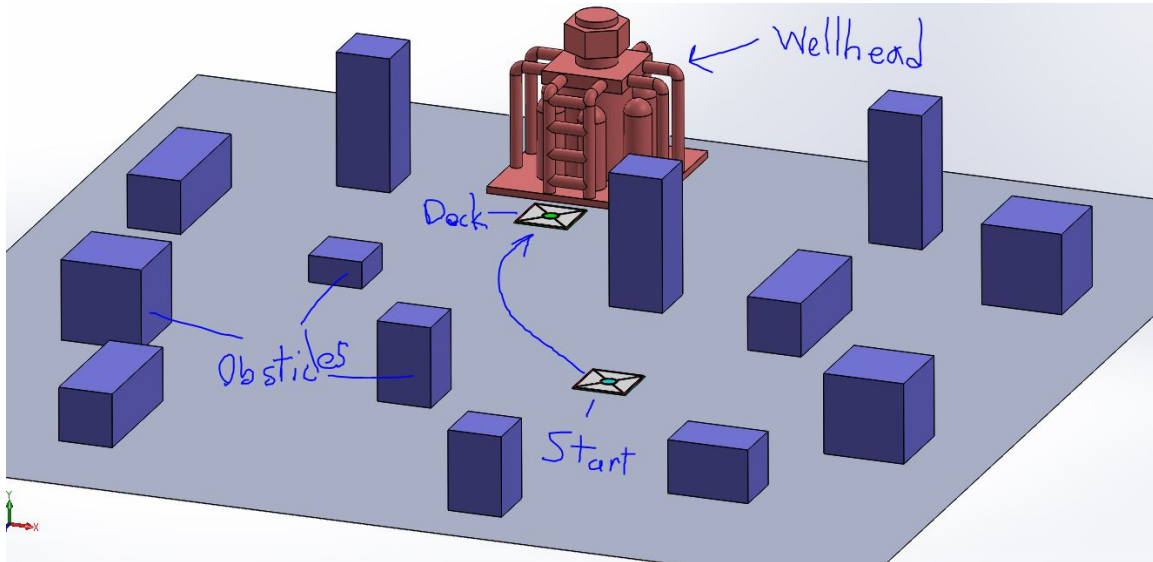


Figure 1: General Environment Layout

I took some creative license on the mock well-head, looking at picture of actual wellhead and relying on my own imagined interpretation, as seen in figure 2.

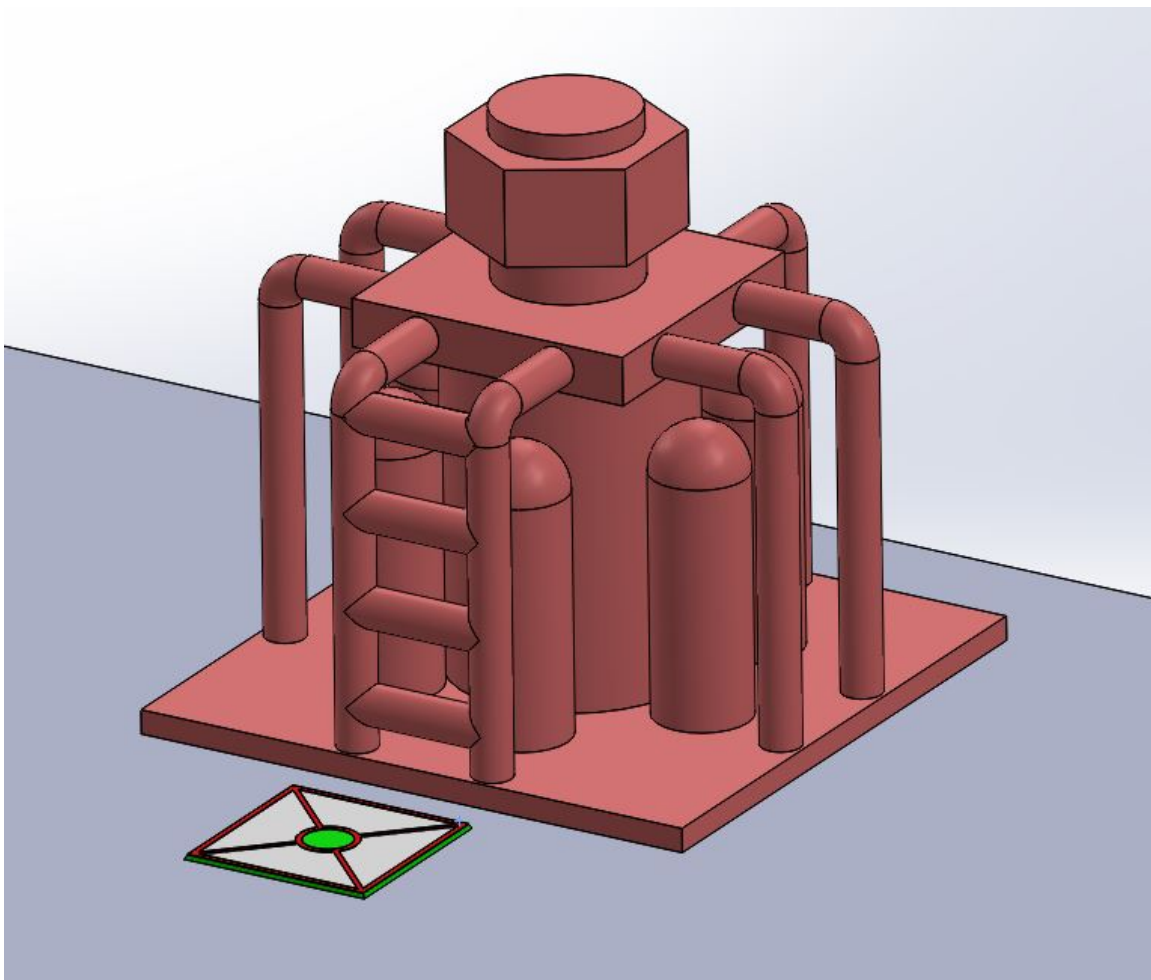


Figure 2: Mock WellHead

The quadrotor, whose model I found online, will start on some initial landing pad, liftoff and then search for the wellhead in the general vicinity as seen in figure 3 and 4.

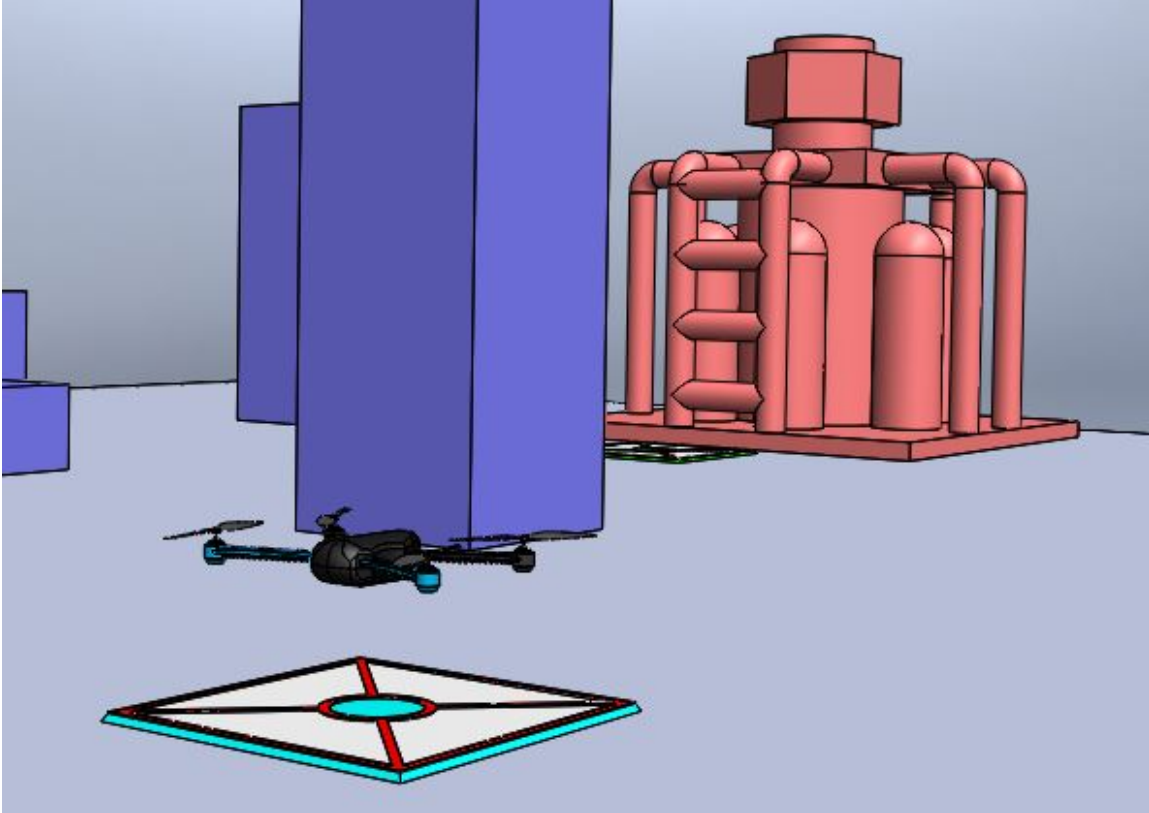


Figure 3: Lift off from initial Landing Pad

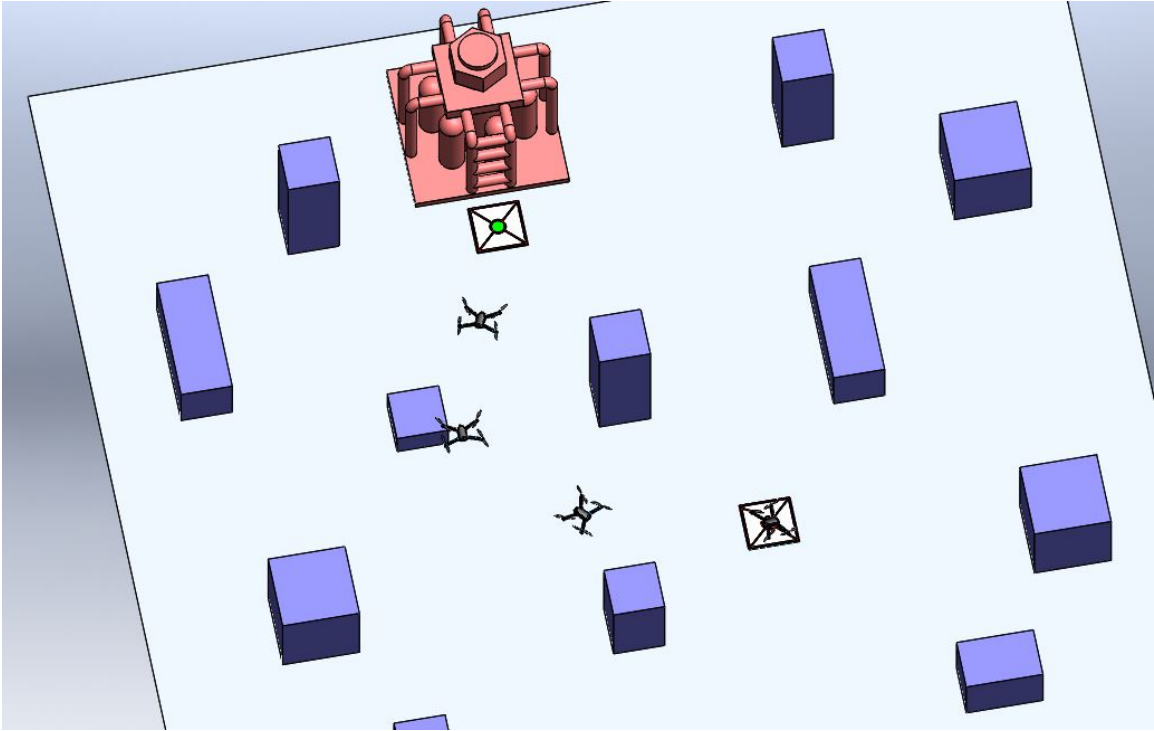


Figure 4: Searching for Wellhead

When the quadrotor encounter the wellhead, it will scan it to identify it, as seen in figure 5.

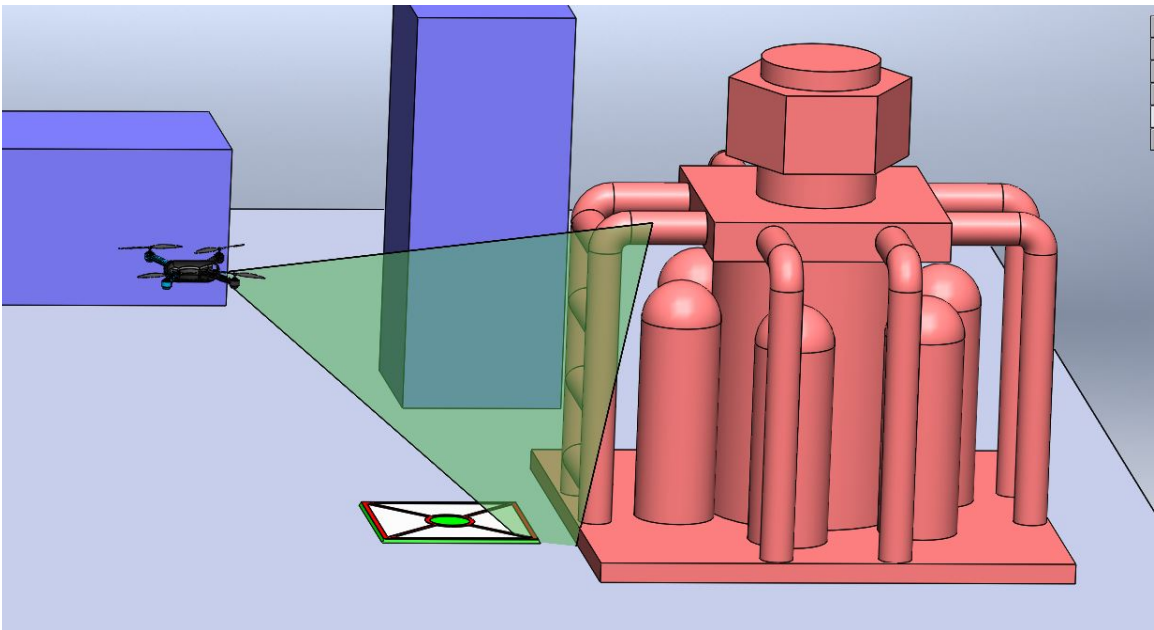


Figure 5: Scanning WellHead

Then the quadrotor will orient itself to the pre-docking position, as seen in figure 6.

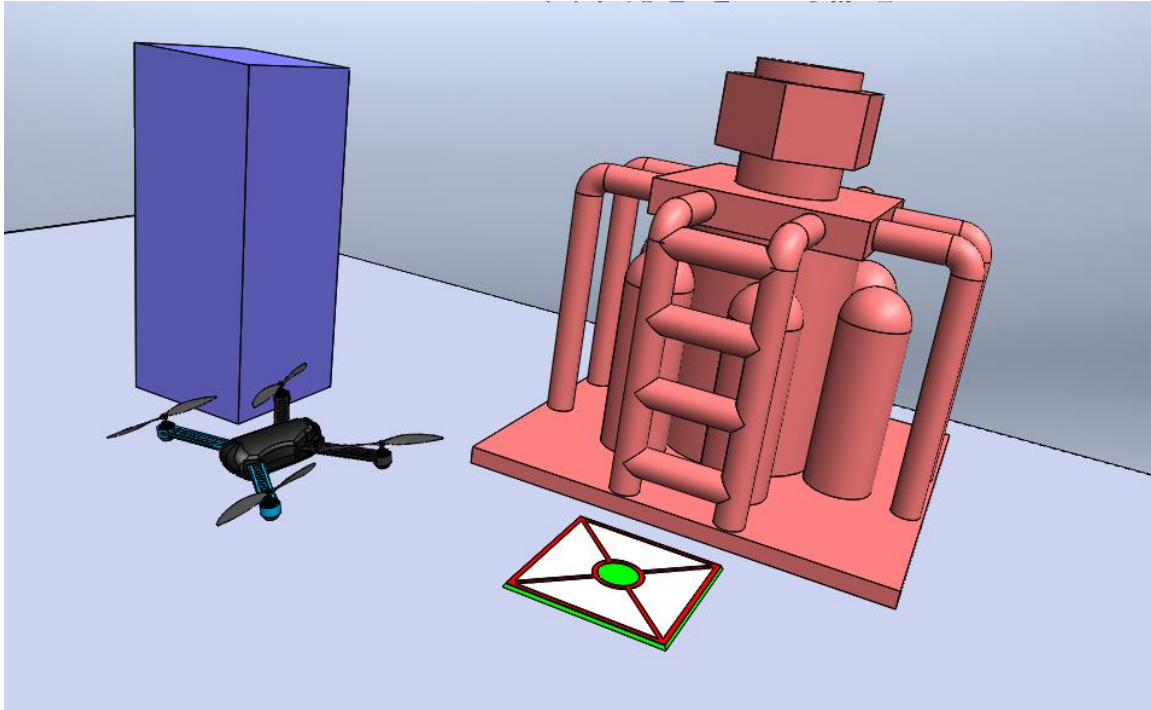


Figure 6: Orientation before Docking

The models are important for visual conveying information about our project. These models will be useful in future presentations and documentations. The models are also expandable to addition of new parts, understanding geometry constraints, and adding animations to create compelling videos for our mission.

I also created a quick sketch to use in our user story.



Figure 7: Quick User Story Sketch

Challenges:

I am grateful to have the opportunity to work outside my normal element and code from scratch on a challenging project. One of the two major holdups was realizing that running two versions of ROS simultaneously present problems. I originally download the latest version of ROS, Jade, right after dual booting with Ubuntu. This appeared to work splendidly for the tutorials. When it came time to git clone the Ardrone Library, I learned it only supported up to Indigo, the previous version of ROS. So I downloaded the Indigo Version, but soon after ran into Source setup.bash problems. The two version called reference to certain scripts and ROS tools the conflicted with paths and references (it's still a little unclear the actually problem). After uninstalling Jade, the ROS system functioned properly again.

The other major hold up was tackling the syntax and object oriented coding of C++, which I have little no previous experience in.

On the other hand the media and modeling are within my element and little to no challenges were present in their fabrications and development.

Teamwork

As a team, we all worked and the risk management, website, and work breakdown structure together. We are working under a scrum management system and are currently in sprint 2. In sprint 2, our core objectives are to get the Ardrone

flying under ROS control as soon as possible, purchase the necessary parts and finalize our detail work breakdown structure.

This week Erik was responsible management and spearheading the work breakdown structure, which he did marvelously almost to the point of OCD. We are now way more organized and have a strong horizon view of the mission we need to accomplish the hour required for each simple task, a strategic way of delegating the work and burnout chart of our weekly progress. He contributed on the research of high-level state estimation techniques and finalize scouting of the processors for our project.

The week Rohan was responsible for the development and implementation of the ROS Reader Node, which routes the sensor and telemetry data from the Ardrone using the driver. His mission was successful, and the reader node is operational and running. He also put in the purchasing request for the IRIS Quadrotor, and the early-on basic sensors.

This week Cole took the lead on the initial risk management documentation, initial outline for the Preliminary Design Review (PDR) and the framework for the Website. Cole's initial draft of the risk management structure expedited the progress during the team meeting. His framework for the website, allow the rest of the team easily delegate remaining sections making its completion efficient.

Upcoming Week

This upcoming week, I will continue establish a ROS mover_node to communicate with the AR Drone. In addition, I will attempt to have the node initiate a take off and land instruction once it is operational. I also hope to have tele-operated key-press input commands to the system, similar to the TurtleSim ROS Tutorial.