Progress Review 1

Cole Gulino

Team C / Column Robotics

Teammates: Job Bedford, Erik Sjoberg, Rohan Thakker

ILR # 2

October 23, 2015

Individual Progress

For this progress review I mainly focused on the Risk Management Table, outlined the PDR tasks left to be completed, and setup the website outline.

The first task that I tackled was to work through the general outline of the Risk Management Table. I looked through each of the requirements that we had worked out before hand. Figure 1 shows the requirements I used to work through the Risk Management Table.

- *3.1 Functional Requirements*
  - 3.1.1 Locate Oil/Gas Wellhead
  - 3.1.2 Autonomously Maneuver to Wellhead
  - 3.1.3 Positively ID as correct wellhead
  - 3.1.4 Align with wellhead dock
  - 3.1.5 Rigidly dock to wellhead
- *3.2 Non-Functional Requirements*
  - 3.2.1 Reduce cost of wellhead intervention
  - 3.2.2 Ease of operation
  - 3.2.3 Simulates Underwater Environment
- *3.3 Performance Requirements*
  - 3.3.1 Sequence completed within 4 hours
  - 3.3.2 Locate specified oil/gas wellhead
  - 3.3.3 Get within visual range
  - 3.3.4 Positive ID of specific wellhead (90% confidence)
  - 3.3.5 Autonomously maneuver to desired wellhead (90% success rate)
  - 3.3.6 Aligns with dock to within a 1 meter radius
  - 3.3.7 Docks with wellhead>50% docking success rate per attempt
  - 3.3.8 Provide feedback
  - 3.3.9 Operable by <= 2 people

Figure 1) Requirements

Once I analyzed the requirements, I outlined a number of risks and filled out the Risk Management Table with them.

Table 1 shows the Risk Management Table.

RIsk #	RIsk	Requirement	Type	Description	Owner	Consequence	Likelihood	Severity	Risk Reduction Plan
5	Cannot get enough precision on docking descent	3.1.5	Programmatic	There is not enough precision on descent due to unstable dynamics of UAVs	Rohan	4	4	16	Design a dock for low landing precision.
1	Cannot get accurate localization of system	3.1.2	Technical	We cannot get accurate localization from our sensors	Rohan	4	3	12	Use external means of global positioning system: tags on the ground or using external visual systems
6	Cannot get the UAV to dock successfully	3.1.5	Programmatic	Dock design and manufacturing does not have the properties needed to successfully dock, or the quadcopters dynamics or structural properties stop the quad from successfully docking.	Job	4	3	12	Prototype multiple dock designs early and often
14	Quadcopter goes wild during run	3.1	Technical	Quadcopter has unexpected motion that can be damaging to the quad or others around it	Job	3	4	12	Create an ABORT button on the computer to take control of quad and land it if it has unsafe motion
8	Drone is damaged	3.1	Schedule + Cost	Drone is damaged while testing and operation	Cole	3	4	12	Use a net while testing Buy multiple backup parts Save budget for a replacement drone ($600)

2	Extra payload on UAV throws off dynamics	3.1.2	Technical	The extra payload on the quadcopter might change the dynamics of the system and will require modification of controller	Rohan	3	4	12	Test the dynamics of the system early
15	Not enough battery life	3.3	Technical	Quadcopter does not have enough power to successfully meet requirements	Job	2	5	10	Keep extra batteries on hand for hot swap and possibly add extra battery power to payload
13	System error while in flight	3.1	Programmatic	There is some system error that occurs while Quad is in flight, resulting in a loss of control	Cole	2	5	10	Every exception must be handled correctly. Develop E-Stop / abort system.
9	Accurate sensing requires expensive sensor	3.1.3 3.1.4	Technical	The inexpensive sensors we have in the lab or get early cannot detect dock and/or obstacles	Erik	3	3	9	Save money on the budget for expensive sensors
4	Not able to detect obstacles and other objects	3.1.2 3.1.3	Technical	There is not enough processing or payload capacity to be able to have a good enough system to detect obstacles and other objects	Erik	3	3	9	Buy multiple processors and test them for speed and low weight. Use methods of visual recognition which require less processing and memory: like tags.
10	Dock does not rigidly connect with UAV	3.1.5	Technical	Dock that is designed does not rigidly dock	Job	3	3	9	Design a mechanism to attach rigidly to quad externally
12	Software packages do not work on ARM architecture	3.1	Technical	Software packages that we need to do certain tasks do	Erik	3	3	9	Buy an extra x86 based OBC and test for compatibility with needed packages

				not work on our ISA or our operating system. Reduces effectiveness and creates extra work					
7	High and low level software system dependencies	3.1.1 3.1.2 3.1.3 3.1.5	Schedule	There are high level dependencies on high level and low level designs which is hard to work in parallel	Cole	2	4	8	Use the AR.Drone2 to work on the high level software design, and use the Iris+ for low level software design
11	Group schedules conflict	3.2	Schedule	Group has external schedules that conflicts with group work	Erik	2	4	8	Plan ahead of time work around each other's schedule

Table 1) Risk Management Table Organized by Severity

In the process of building the Risk Management Table, there were about 6 risks that I identified as the most severe. The most severe risk that I identified was that we can not get enough precision on docking descent. This would be incredibly severe, because if we cannot get enough control of the quadcopter, we cannot even do the most basic aspects of our scope. This kind of control is incredibly difficult to attain, so it received the most attention whenever I did the first draft of the Risk Management Table.

The next incredibly severe risk that I identified was that we cannot get accurate localization. This would be very damaging, because it would undermine all of our perception and planning algorithms. It is also incredibly likely, because this kind of localization without using external markers is very difficult.

The third severe risk that I initially identified is that we cannot get the UAV to successfully dock. This is a very serious technical problem that runs through both our hardware and software systems. It even runs into our external infrastructure system with the wellhead and dock. This is a likely risk, because it could have problem through every stage of the project: design, manufacturing, and integration of systems.

After working on the Risk Management Table, I broke down the Preliminary Design Review into sections, outlined the work to be done, commented it with pointers about each section from the Conceptual Design Review provided by Professor Dolan, the TA, and Professor Dimi. I also tentatively put an owner by each section.

Figure 2 shows the PDR outline.

1. Project Description (0.5) (Cole)
    a. **Refined** project description consisting of user needs and your proposed method of meeting them.
    b. Focuses on end results, not details of the technology
    c. Notes from CoDR:
        i. Project description should capture what the team will actually do in the context of the MRSD
2. Use Case (1) (Cole)
    a. Give a brief refined use case coupled with a **graphical representation** of the system in its **use case/mission environment**
    b. Should make it clear what your system will do
    c. Notes from CoDR:
        i. Gets the basic idea across in a cute way, but could use some additional detail
            1. Why is the figure that is apparently connected with it located at the end of the TOC without a caption, figure number, or any other identifying information
            2. Expand on the sentence that begins "The vehicle notices the wellhead, surveys every inch…" by giving an idea of how it is done
            3. Presumably a camera is involved, there is a remote control interface, etc…?
        ii. Should be a lot more on the autonomous ROV than the wellhead itself.
        iii. Should revolve around the functionalities of the ROV
        iv. More sketches
        v. Missing graphical representation of use case
3. System-level requirements (1.5) (Rohan)
    a. Summarize your system requirements
    b. Identify any requirements changes since CoDR with justification for the changes
    c. Notes from CoDR:
        i. Performance requirements give a good quantitative idea of what you are aiming for
        ii. List the sequence before stating its max duration
        iii. Need to enumerate your requirements somehow in order to keep track of them throughout the year
        iv. Describe what are the objectives of the project and what will be done specifically for the MRSD Project Course
        v. Separate the requirements in mandatory and desirable as required
        vi. Separate requirements to be clear and discrete
        vii. Each one will have to be quantifiable
        viii. Restructure some of the requirements to look like individual statements, rather than sub-lists
        ix. Sub-lists make it hard to follow the logic
4. Functional Architecture (1) (Cole)
    a. Block diagram showing your system's major functions and the flow (information, energy, material) between them down to one level below the one presented in the CoDR
    b. Notes from the CoDR:
        i. Some of the process details nicely described here should also be summarized in the use case, as alluded to above

        ii. Drill down into what you mean by 'sense the environment' - which functional requirement are you trying to fulfil?

5. Cyberphysical Architecture (1) (Job)
    a. This is a block diagram showing your system's major cyberphysical (hardware and software) components and the flow (information, energy, material) between them
    b. Notes from the CoDR:
        i. Include the power components of your system and try to highlight direction of flow of information, flow and energy between subsystems, to help you visualize your system at an even finer level
        ii. Capture correlation between the functional and cyberphysical architecture
6. System/Subsystem Descriptions (2.5) (Cole)
    a. Describe and depict the key technical aspects of your system and each major subsystem as your expect it to appear and function in your final system
    b. This should start with an overall system graphical representation and include CAD drawings, photos, or other graphical representations of your subsystems
    c. It should not consist entirely of block diagrams or simply be zoom-ins on components of your functional or cyberphysical architecture block diagrams
    d. Notes from CoDR:
        i. Overall these are good, detailed, and informative
        ii. Why is the search area circular?
            1. State why
        iii. Section is supposed to be a more detailed presentation of the subsystems and key technologies in the cyberphysical architecture
        iv. Detail the electro mechanisms, sensors, software, and user interface
        v. As is, the section describes mostly functionalities as opposed to technical details of the subsystems
        vi. An easier way to think of subsystems would be: electrical, mechanical, vision, planning, etc.
        vii. Designing subsystems in this manner helps you think about the scope of the work for each person responsible in the team and whether that is achievable
7. Current system status (2.5) (Rohan)
    a. Describe and depict what you have developed so far
    b. This should run parallel to the subsystem descriptions by giving the current progress towards the final desired state of each of the major subsystems
    c. Use videos or simulations for moving parts
    d. Fall-semester targeted system requirements
    e. Identify the system requirements and corresponding subsystems and system elements emphasized during the fall semester development
    f. Current system/subsystem descriptions/depictions
    g. Describe and depict the subsystems developed during the fall semester
    h. Start with an overall system depiction
    i. Use design drawings, photographs, schematics, brief video clips, and other visual means to convey the status of your system/subsystems
    j. Modeling, analysis, and testing
        i. Include a summary of any modeling, analysis, and testing you performed in order to design your system to specification and unit-test its components
    k. Performance evaluation against the fall validation experiment.

l. Summarize how well your system performed against the scenario and metrics specified by your fall validation experiment

m. Fall semester website video excerpt
   i. Show at least the beginning of the 2-3 minute fall video that you are required to post on your website
   ii. The beginning of the video should be a "teaser" that shows as much as possible of the system operating performing fall validation experiment-type activities
      1. Refer to the project website guidelines

n. Conclusions
   i. Highlight current system strong and weak points and needed areas of refinement

8. Project management
   a. Work Breakdown Structure (1) (Erik)
      i. Present a high level summary of the three-level work breakdown structure you developed in the sys eng class. Save the WBS details for the Critical Design Review Report due at the end of the semester
      ii. Notes from CoDR:
         1. A hierarchical representation showing the major tasks and their breakdowns would be helpful
         2. A notional work plan and tasks is missing
         3. What is presented as a WBS is more an enumeration of low-level tasks
         4. The work plan needs to be formulated top-down
         5. Great level of granularity
         6. Member responsibilities: assign primary and secondary person responsible to each task/subsystem
         7. More visual
   b. Schedule (0.8) (Erik)
      i. What are the major system development milestones on the remaining schedule?
      ii. Are you behind, ahead of, or on schedule?
      iii. Notes from CoDR:
         1. Way too generic
         2. You need to map tasks identified in the WBS to the Progress Review dates in the fall, and to the ends of months in the spring
         3. The schedule you have is just playing back the course schedule to mecti
         4. The report is missing even a basic schedule on the flow of work
         5. The WBS and analysis on the sequence and precedence of the tasks (work packages) and allow for the development of a schedule
         6. For each spring, detail what you plan to do in that sprint, the milestones achieved at the end of the sprint?
   c. High-level test plan (2) (Job)
      i. Use a table to present your test plan as a set of capability milestones for the three remaining progress review in the fall semester (PR3, 4, and 5-6 since 6 and 6 are FVE and its encore) and for roughly the ends of each of the four months (Jan-Apr in the spring semester). Indicate briefly how these capabilities will be tested
      ii. Fall and Spring Validation Experiments (FVE/SVE). These are the final two (2) Progress Reviews in each semester, to be presented in lab demonstrations lasting no more than 30 minutes per team. Include them in summary form in the table above, but

also describe them separately in somewhat greater detail than the other capability milestones, including these essential elements:

1. The test conditions: location, needed equipment, size and nature of operating area, etc.
2. A list of steps your system will be put through written in a sufficiently clear way for someone with no knowledge of your project to be able to test the robot
3. A set of quantitative performance metrics that your system will be measured against during the validation experiment. Typically these metrics will be written to the list of steps in the previous item
4. Use graphics to the extent possible to illustrate your FVE/SVE

d. Budget (0.7) (Erik)
   i. What is your total budget?
   ii. What are the big-ticket items that comprise the majority of your budget?
   iii. How much/what percentage have you spent to date?

e. Risk Management (1.5) (Cole)
   i. Using techniques learned in the Sys Eng class, present the following
      1. Risk Management table with Risk ID, Risk, Requirement, Type, Likelihood, Consequence, Mitigation
      2. Risk Likelihood-Consequence table

9. Intangibles
   a. Length (1)
      i. 30 min
   b. Intelligibility (2)
      i. format, flow, demeanor, audience connection
   c. Handling of Q&A (1)

Figure 2) PDR Outline

Finally this week, I set up the outline for the website that we all completed on Friday.

Challenges

The main challenge that I faced personally was to think up the large list of possible risks. Risks are hard things to imagine before you begin building. They are oftentimes things that you may not have thought of. That is why they are so important to get out at this stage. Working on this really opened my eyes to the major challenges facing the team, especially schedule wise.

Another challenge we faced as a group was working on our WBS. The team worked hard this week to divvy up tasks for our second sprint. Erik setup a system for us to track progress and hours. We used this to come up with the tasks we were going to divvy up this week. There was serious challenges involved in making the decisions about which aspects to focus on now.

Teamwork

Even though I got the main risks down in the table, the rest of the team went through all of the tasks, made some adjustments, assigned owners, and edited the weights. In this way, I was able to set the groundwork for the team as a whole later.

The website structure was setup by me, but the team worked together to ensure that every part of the website that needed to be completed was. We each took ownership of a page of the website and worked until everything was completed.

As mentioned earlier, we worked together to come up with the tasks we want to show at the Fall Validation experiment, broke them up into subtasks, and chose which tasks we wanted to complete for this two week sprint starting last monday.

Future Plans

During the next week, I will be working on researching pixhawk controller documentation, beginning design on the power distribution circuit, researching global position strategies, and setting up a ROS launch script for our first two nodes that Rohan and Job are writing.

Erik will be working on implementing a 2D x,y map in ROS and detailing our software architecture.

Rohan will be working on integrating and documenting his READER node shown during this week's demo. He will also be ordering his camera and researching existing algorithms for visual state estimation.

Job will be working on completing and documenting the MOVER node along with possibly adding an abort/start button to the program.