**Graduate Projects**

University of Colorado at Boulder

Aerospace Engineering Sciences

ASEN 5018/6028 –Spring 2015

|  |
| --- |
| **Drones Versus Zombies (DVZ)**  **Test Engineering (TE)**  **Summary/Continuity Document** |

|  |  |  |
| --- | --- | --- |
| **Approvers List** | | |
|  | Title | Name |
| Prepared By | TE Lead | Benjamin Zatz |
|  |  |  |
| Approved By | TE Lead | Benjamin Zatz |
| Approved By | Systems Engineer | Edward Scott |
| Approved By | Project Manager | Garrett Brown |

# Introduction & Summary

The focus of the testing subsystem for this project is centered on requirements and data analysis. In stage one of the project, the focus was to work with the systems engineer to create requirements and make sure that each requirement could be verified. Flowing into stage two which was the focus of this second semester was to continue to flow down the requirements into their level three subsystem form. As the project began to take form the main role of the test lead was to analyze the data from individual subsystem tests. This included creating MATLAB scripts to process all of the data from controller output to localization results.

At this stage the requirements have been sorted out down to the subsystem level and extensive testing has taken place for each main subsystem. Very few requirements were able to be verified beyond the baseline customer requirements that are outlined in level one. For the controller subsystem a GUI was created to process the data outputs so that they could be evaluated efficiently. As for localization some of the outputs can be seen in the GUI but others such as the Kalman filter data were processed in a different script.

# Semester Report

## Objectives and Tasks List

Completed:

1. Level one requirements
2. Level two requirements
3. Level three requirements
4. Safety evaluations
5. Verification test equipment
6. Localization processing scripts
7. Kalman filter processing scripts
8. Controller processing scripts
9. Hardware evaluation tests
10. Hallway test procedure
11. Autonomous position hold test

Incomplete:

1. Full requirement verification
2. System validation testing
3. Camera system integration and testing
4. Identification implementation and testing
5. Path planning implementation and testing
6. Full network evaluations
7. Multi drone integration and testing

## Issues

1. Some requirements were decided to be out of scope for this year
2. Hardware was found to not perform as well as expected
3. Network issues caused unforeseen setbacks
4. Controllers had to be reworked to obtain necessary performance
5. Process rates limited many algorithms from reaching their expected performance
6. The integrated system did not meet safety requirements to perform system validation
7. New hardware arrived defective
8. Crashes during testing caused setback due to damaged hardware

## Lessons Learned

1. Clarity is key for testing. When it comes to procedures, the only way to make sure that you have covered every step is to sit down with the hardware and do it. Missing one small step can be the difference between a successful test and creating additional problems.
2. Understanding your audience. Each element of a test should be able to be understood by any person you want to be able to perform the test. This does not always mean that it has to be written for a person who has no association with the project.
3. Always keep in mind why you are performing each test. You can write hundreds of tests, but efficiency is crucial when the project has a set end date.
4. Work with all of your team members so that you understand the progress that is being made. Test expectations and plans must adapt with the project, thus the more informed you are about the project the more effective you are to your team.
5. Learn from others’ mistakes, there are many people tackling similar problems to the ones we are approaching. In many cases their experiences are well documented so that we can expand upon what’s already been done instead of making the same mistakes.
6. Test every piece for hardware when it arrives to make sure that it has the capability to perform as you expect.
   1. The PX4 flow was not able to provide the system with velocity measurements to perform the dead reckoning in the Kalman filter
   2. When the second Hokuyo arrived it was immediately integrated into the system without being tested on its own. It turned out that the readings it was providing were false which created multiple setbacks. This caused us to reevaluate other elements of the project thinking that there were problems with how the data was being used rather than it being the actual data that was wrong.
7. When testing a new piece of code or checking to see if a new approach would work it needed to be designed such that it could be tested piece by piece. This way each element could be evaluated on its own to determine if the results were correct.
8. When analyzing the data make sure that you understand what the data is before throwing it on a plot. Occasionally data was saved that did not actually correlate with the answers that we were looking for.

## Procedures

### Requirements

Before any tests can be created the requirements must be examined and refined to reflect the true goals of the project. The more detail in the requirements the easier it is to make tests and know that the tests will reflect the actual performance of the system. The requirements are broken down into three levels and can be found at the following locations:

* DVZ/Systems Engineering/Requirements Specifications/ DVZ Requirements (L1)
* DVZ/Systems Engineering/Requirements Specifications/ Element Requirements (L2)
* DVZ/Systems Engineering/Requirements Specifications/ Subsystem Requirements (L3)

In these folders you will find each level of clarity for testing and it provides a baseline for how these requirements flow together. While many can be tested together it is critical to make sure that none are overlooked. This is also the guide to make sure that each subsystem is designing a system that can perform to the correct metrics.

### Safety

Due to the location of the final demonstration being in the Engineering Center, there were many safety considerations that had to be made. Starting with the tests that were planned to be performed this semester in the first basement of the Aerospace Engineering wing, the extent of the safety precautions had to be fully evaluated. In the Verification and Validation Test folder under Lockheed Martin Hallway are photos that showcase the areas of the hallway in which possible hazards were identified. These were then looked at and evaluated for their potential severity level which is further defined in the Preliminary Safety plan document. The Preliminary Safety Plan outlines how each issue will be approached when the tests take place. A key test task during phase one was getting the safety procedures approved by the necessary parties. A meeting took place with JoAnn Zelasko, who is the assistant dean of admissions to determine the necessary documentation before testing was carried out. Ultimately the safety plans were approved and Matt Rhode was determined to be the appropriate point of contact for specific test safety concerns.

Beyond test location safety, the safety of the team must also be taken into consideration. The VICON space provides netting around the test area which acts a preliminary barrier, but this is not sufficient. Safety glasses must be worn by anyone who is in the test area whenever a flight test is being performed. This includes both manned and unmanned tests as there is always a chance that something may go wrong. In many cases the blades would break if the octo-quad flew into a wall or the netting.

### Test Procedure Documents

For this project the procedural documents include ones for manned and unmanned flight tests as well as safety procedures. The safety plan focuses on what is required to fly in the Lockheed Martin Hallway, but many practices should also be implemented in the RECUV flying lab. These documents can be found at the following location:

* DVZ/Verification Validation Test/ Test Plans

These tests provide an outline of how a test of this nature should be performed. DVZ tests are primarily not requirements-focused due to the fact that they are meant to be adapted slightly based on which requirements are being verified.

The bulk of the testing that has been performed has been focused on data analysis and characterization. All of the data analysis for this project was done with MATLAB in a combination of the GUI and individual scripts for processing.

* DVZ/Verification Validation Test/ Test Data

In this folder you will find a whole list of folders which correlate to different systems that were tested throughout the life of the project. As the project has progressed, individual test files have been retired as the system came to be integrated. The Full System Tests folder is the current focus of the testing. This is in part due to how the log files have changed throughout the project from only containing partial information to now containing all of the necessary information. In this folder you will find folders labeled with dates that correlate to when the data was taken. Opening plot\_interface.m and running the script will produce a GUI which can be used to process results from any number of tests. This is the main tool for looking at data as it allows you to easily access different results and compare them with system performance. The other main items is plot\_Acceleration.m, this file is the post processing for the Kalman filter. This will plot all of the necessary information for understanding the performance of the Kalman filter as compared to both VICON and the scanmatcher on its own.

Table 2.: Software list

|  |  |  |
| --- | --- | --- |
| Program Name | Version | Purpose |
| MATLAB | Any beyond R2013b | MATLAB was used to process all of the results from testing each subsystem |

# Next Semester/Future Expectations

## Prioritized List of Tasks and Objectives

1. Finalize the level 3 requirements and specifically the performance metrics
2. Update test procedures
3. Correlate different tests with requirements
4. Define final verification and validation approach
5. Consolidate data analysis scripts
6. Define procedures for using multiple drones
7. Test new Hokuyo
8. Create path planning tests
9. Create identification tests

## Starting Points

1. The first task should be to review the requirements with the systems engineer. The file path for this can be found in procedure, requirements above. This is one of the most important parts about testing and provides the backbone for all the tests that should be completed. Making sure that each requirement has a verification rationale will allow you to easily decide if multiple requirements will be able to be verified in the same test.
2. Test documents will flow according to how the requirements are defined. A starting point for these documents has been created, but work will need to be put in to correlate test documents with requirements. A key thing to note is which requirements can be verified without flying as this will allow for higher efficiency.
3. Sitting down with the customer and redefining what the final validation scenario will need to take place. Getting this on paper is crucial so that when it does take place it will encompass everything that the customer is looking for in the project.
4. The final aspect is data analysis for each of the different subsystems. These scripts are found in the test data folder along with the actual data. There are two main methods for processing at this point which are using the GUI and individual scripts. Best practice would be to run these scripts with old data to understand how they work and what information they provide.

## Improvement, Updates, Verification

1. Many of the assumptions that were made center around the progress that is expected from the system. The full system tests require that each aspect of the project works to its expected potential and most completely fall apart if one element is not fully functional. Testing should be better designed to accommodate a system with components at differing levels of maturity.
2. To be able to test in the Engineering Center, a confidence level of at least 95% in system functionality is desired to ensure that the system consistently performs as expected. Due to the hazards that are involved with testing indoors, the possible mishaps must be fully characterized, and safety procedures improved and validated before testing to allow for any mishaps that occur.