|  |  |  |
| --- | --- | --- |
| Project:  WP Name: Ground Control Station Test Report  WP Number: WP-CG-02 | Type of Test:  Inspection and Data Logging | Test Procedure:  Analyse GCS logs and operation |
| Test Article:  Ground Control Station Software | Part Number:  GCS | Serial Number:  - |
| Test Specification:  IP and IK Code Rating | Test Equipment:  Ubuntu 10.04 Laptop | |
| Test Operators:  Tim Molloy | Test Engineer:  Tim Molloy | |
| WP Group Manager:  Michael Hamilton | WP Supervisor:  Luis Mejias | |

**QUT Avionics**

Queensland University of Technology

CRCSS-EESE, GPO Box 2434

Gardens Point Campus

Brisbane, Australia, 4001.

Telephone (+61 7) 3864 1772

Facsimile (+61 7) 3864 1517

e-mail luis.mejias@qut.edu.au

web <http://code.google.com/p/ahns10/>

This document is Copyright 2010 by the QUT. The content of this document, except that information which is in the public domain, is the proprietary property of the QUT and shall not be disclosed or reproduced in part or in whole other than for the purpose for which it has been prepared without the express permission of the QUT

**Test Summary**

The AHNS Ground Control Station (GCS) is required during flight operations to process uplink and telemetry data for the purpose of airborne system monitoring and control. The data processing operations of the GCS are specified in one HLO and seven SRs with accompanying acceptance test procedures. This document describes the use of these methods to undertake acceptance testing of the final GCS design.

The test revolves around data transmission and receipt without the need for an entire airborne software system. Data is transmitted. and received from a test application using the same UDP network library as the flight computer software. Received data from the airborne system is logged and displayed by the GCS. Uplink data originating from the GCS includes control system gains and parameters and be inspected from the test program output.

**Table of Contents**

Paragraph Page No.

1 Test Objectives 3

2 Test Set-up & Equipment 4

3 Procedure 5

4 Results 7

4.1 Connection 7

4.2 AT-19, AT-20 and AT-08 – Helicopter and Flight Computer State Data 7

4.3 AT-02 - Flight Control 8

4.4 AT-09, AT-17 and AT-18 – Data Log Analysis 9

5 Analysis 11

6 Conclusions 12

7 Recommendations 13

**List of Figures**

Figure Page No.

Figure 1 - Remote SSH Connection to Overo and GCS Read for Connection 8

Figure 2 – Helicopter State Data in the Data Plotter Widgets 8

Figure 3 - Flight Computer State in Data Plotter and System Status Widget with changed parameter and gains 9

Figure 4 - Flight Control Widget reflecting change of Active Control Loops and set points 9

Figure 5 – MATLAB Plot of GCS Logged Helicopter State Data from Figure 2 10

Figure 6 - MATLAB Plot of GCS Logged Flight Computer State Data from Figure 3 10

Figure 7 - MATLAB Plot of GCS Logged Autopilot States from Figure 4 11

**List of Tables**

Table Page No.

**No table of figures entries found.**  
In your document, select the words to include in the table of contents, and then on the Home tab, under Styles, click a heading style. Repeat for each heading that you want to include, and then insert the table of contents in your document. To manually create a table of contents, on the Document Elements tab, under Table of Contents, point to a style and then click the down arrow button. Click one of the styles under Manual Table of Contents, and then type the entries manually.

# Test Objectives

The objective of this test is to ensure the GCS GUI software provides the full feature set specified in the system requirements (SR) SR-B-02, SR-B-08, SR-B-09, SR-D-07, SR-D-08, SR-D-09 and SR-D-10. The test is therefore to determine if the GCS meets all acceptance test standards specified with these requirements.

# Test Set-up & Equipment

* Ubuntu 10.04 GNU/Linux Computer with GCS installed (32 bit required for Vicon use)
* Overo Fire or 32-bit computer to run GCS test application
* A WiFi access point to enable the GCS and test application computers to exist in the same local area network.

# Procedure

This test is reproducible and useful in GCS debugging and UDP communications design. It is therefore presented in instruction format.

1. Connect the GCS computer and the test Linux computer to the same WiFi 802.11 network and ensure they are capable of pinging each other.
2. Launch the GCS (trunk/gcs/) and the GCS Server Test Driver (trunk/airborne/network/gcs\_tester/) programs on their respective computers. The GCS Server Test Driver is program written test transmission and receipt of UDP packets form the GCS. Its uses the same implementation of the flight computer UDP library and functions independent of the control and state estimation code. This enables the GCS to be tested at a component level rather than only at systems integration.
3. Select the test Linux computer IP address in the GCS Communications widget settings for telemetry and attempt to connect. If a connection occurs the test program will display a client added function call. If the test program does not respond the GCS will display an error message.
4. Once connected the test program automates transmission of dummy helicopter state, flight computer state, autopilot state and sensor state data packets. Acceptance test checking can therefore begin.
5. AT-19 for SR-D-09 requires inspection of the state and flight computer state data in the data plotter widget and the received console. If this matches that displayed in the test program the test is passed.
6. AT-20 for SR-D-10 requires inspection of the system status and communications widgets to ensure system health and communications are being monitored. If these are suitably displayed the test is passed.
7. AT-02 and AT-08 for SR-B-02 and SR-B-08 respectively require the Flight Control widget control loop selections to be modified along with the Gains and Parameters. If sent correctly the test application will respond with a successful acknowledgement message which is displayed in the received console widget. If gains, parameters and loop selections are all transmitted successfully then the tests are passed.
8. AT-09, AT-17 and AT-18 for SR-B-09, SR-D-07 and SR-D-08 respectively required inspection of the telemetry logged by the GCS. These logs are found in the logs directory inside the GCS application directory. If the data is comparable time stamped and comparable to that outputted by the test program then the tests are passed.

# Results

## Connection

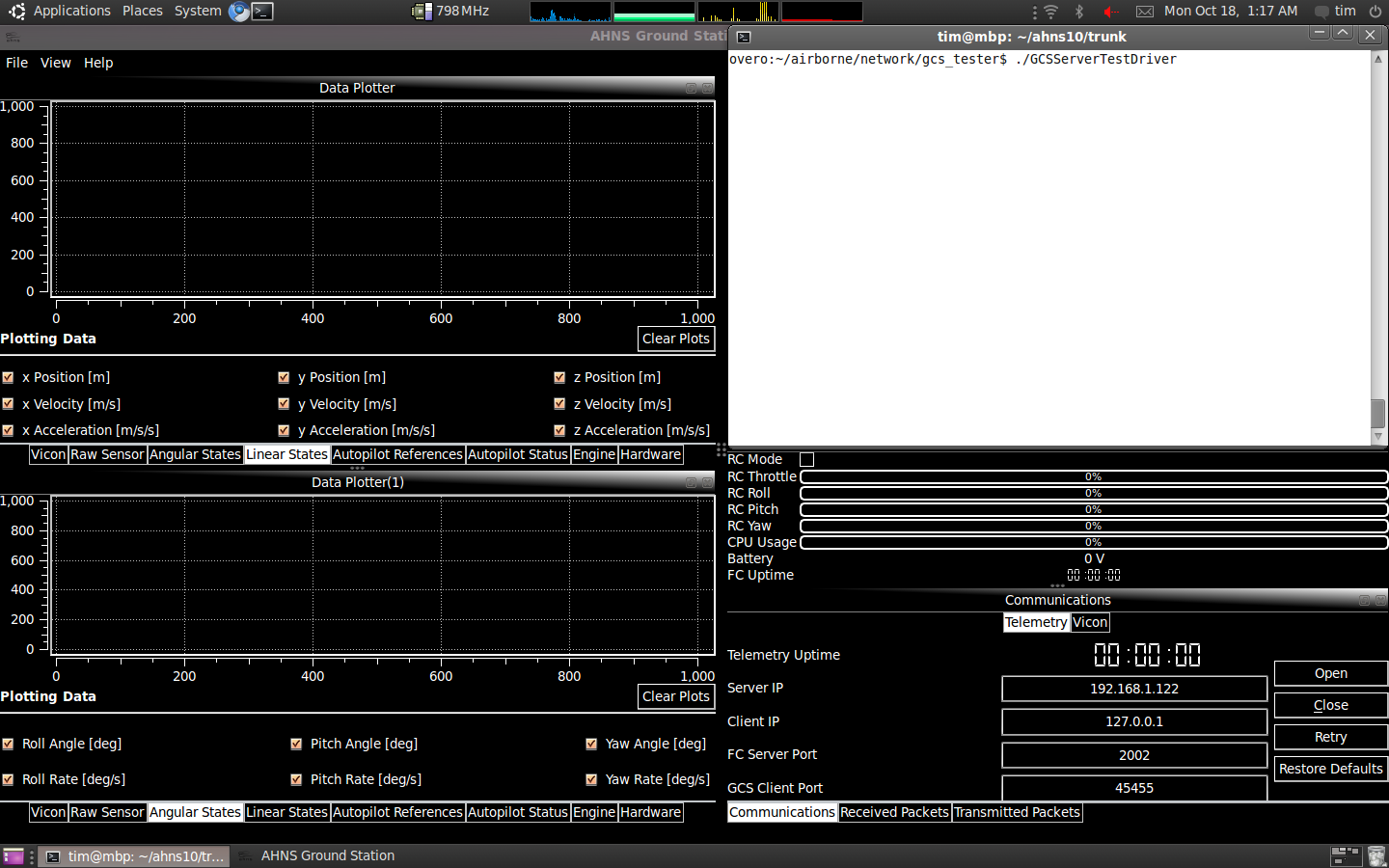


Figure - Remote SSH Connection to Overo and GCS Read for Connection

## AT-19, AT-20 and AT-08 – Helicopter and Flight Computer State Data

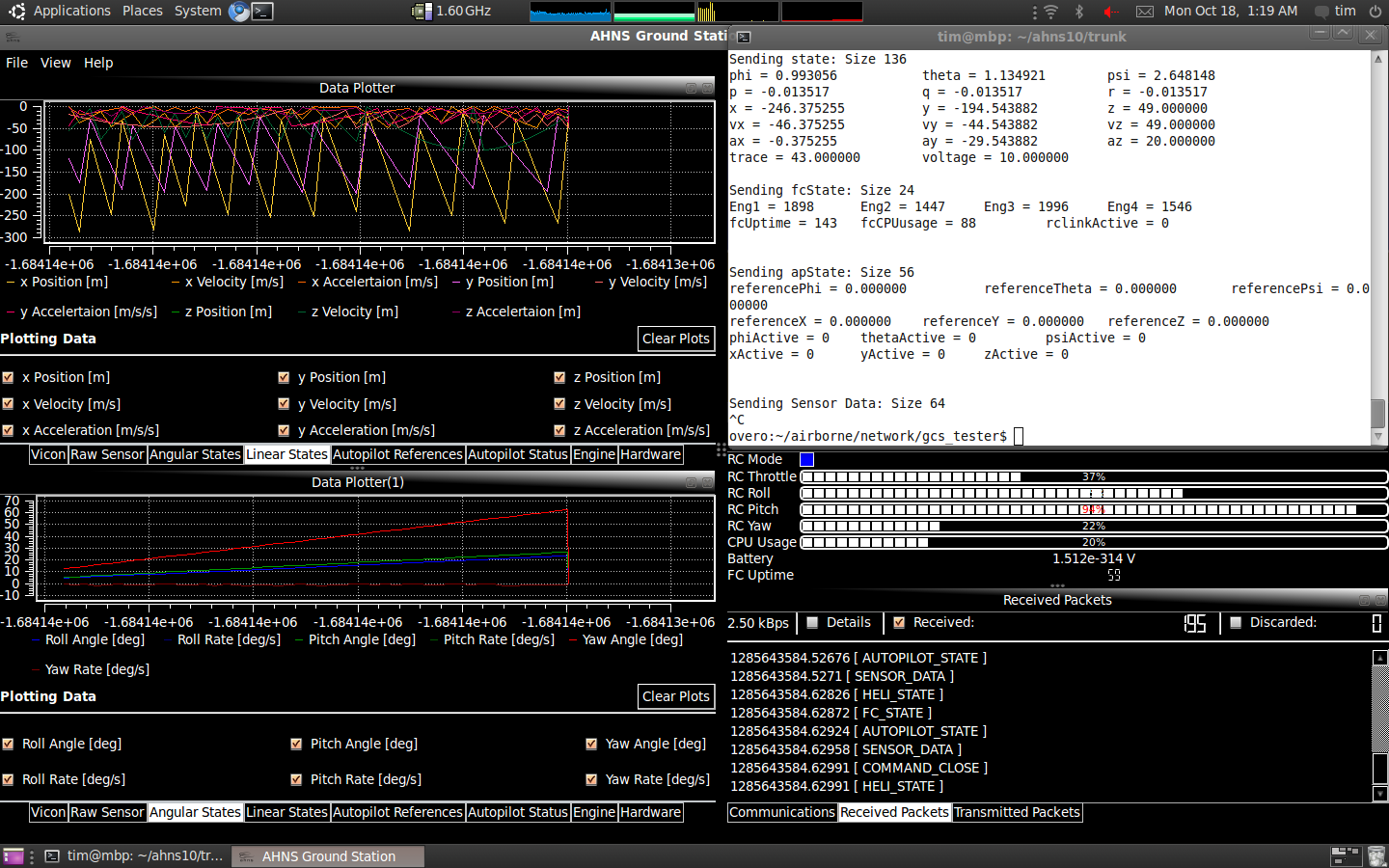


Figure – Helicopter State Data in the Data Plotter Widgets

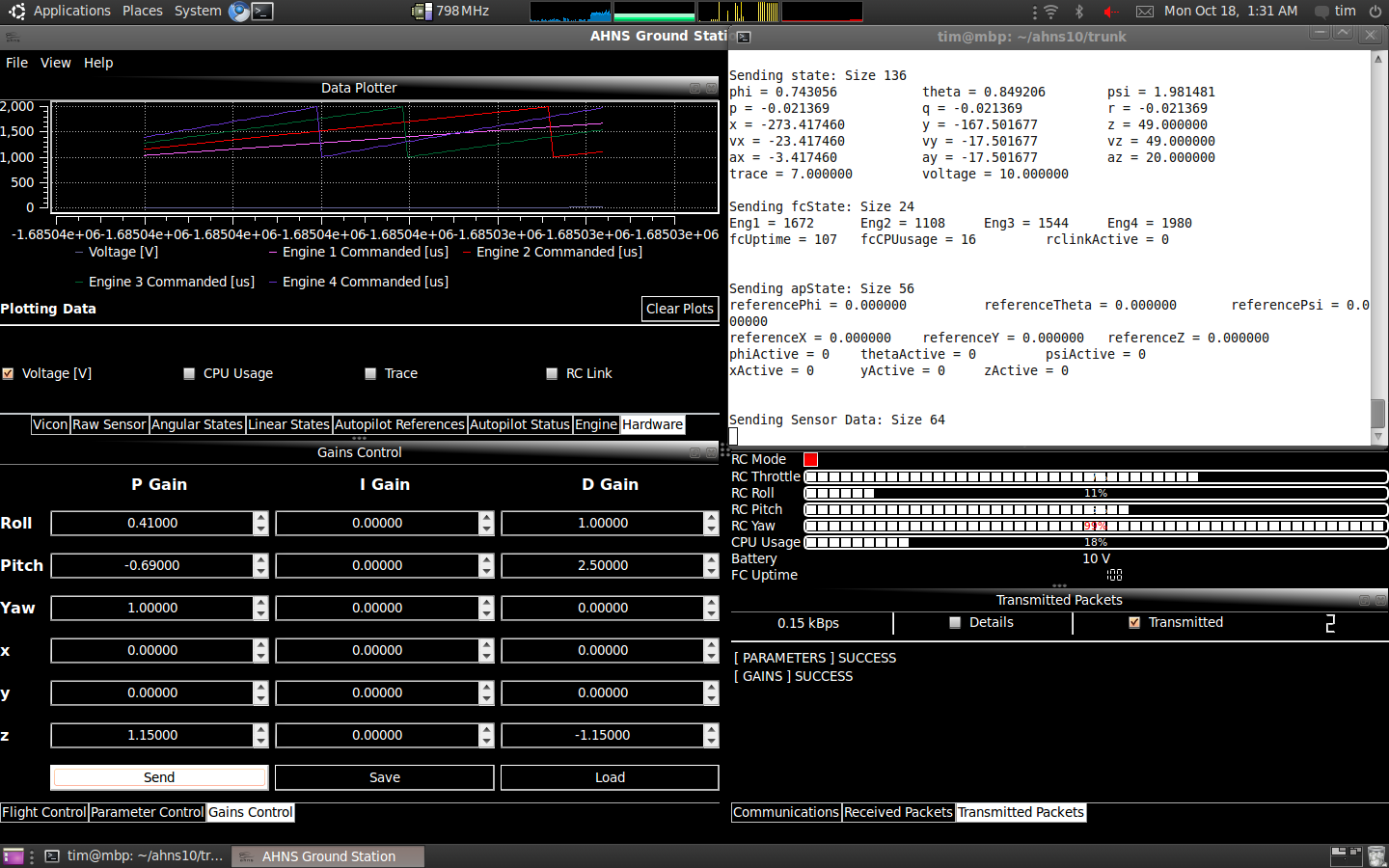


Figure - Flight Computer State in Data Plotter and System Status Widget with changed parameter and gains

## AT-02 - Flight Control

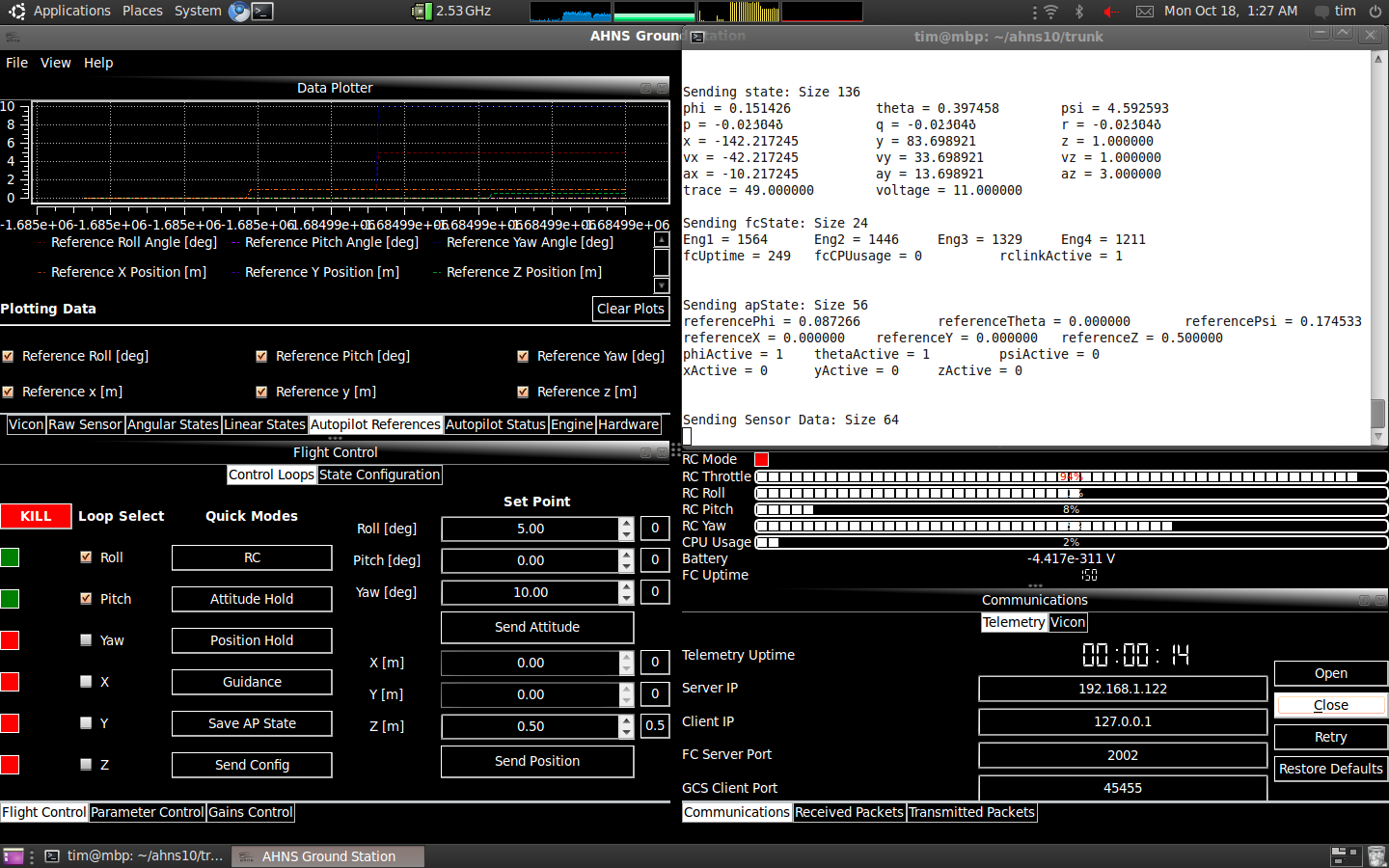


Figure - Flight Control Widget reflecting change of Active Control Loops and set points

## AT-09, AT-17 and AT-18 – Data Log Analysis

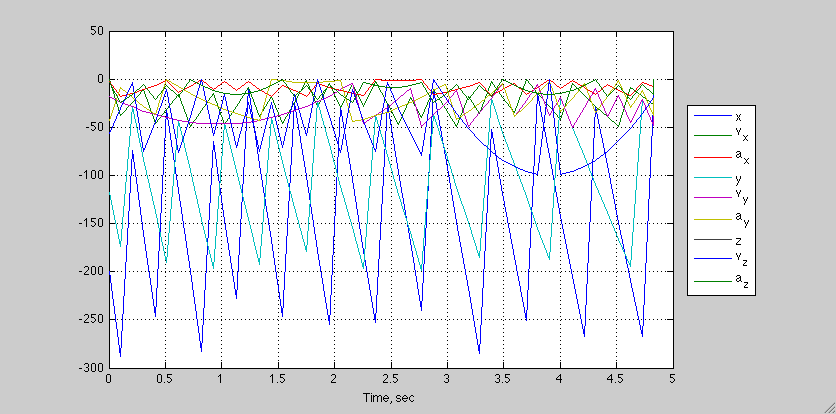


Figure – MATLAB Plot of GCS Logged Helicopter State Data from Figure 2

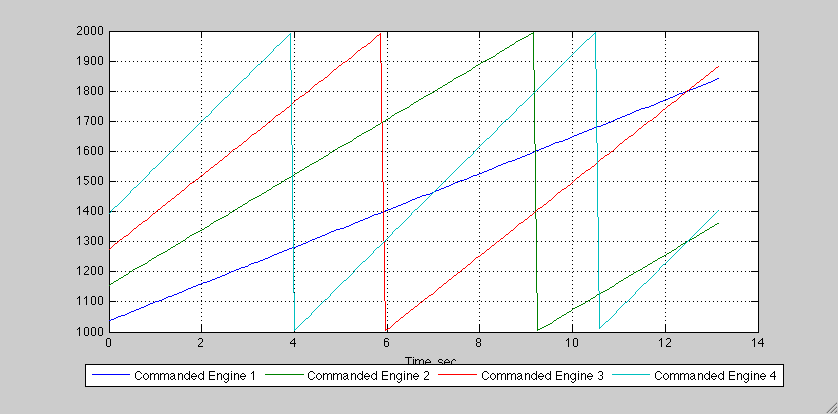


Figure - MATLAB Plot of GCS Logged Flight Computer State Data from Figure 3

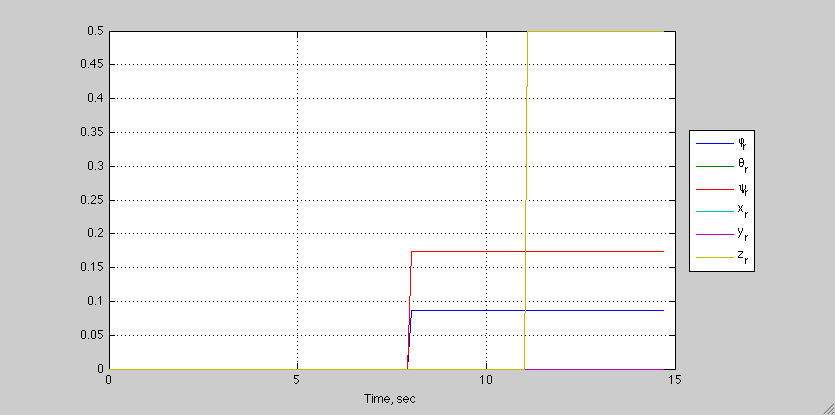


Figure - MATLAB Plot of GCS Logged Autopilot States from Figure 4

# Analysis

It is important to consider the conditions under which the GCS screenshots in the results section were taken and what these represent. Figure 1 shows the GCS running on the local Ubuntu computer with a terminal showing the secure shell (SSH) connection to the GCS test application about to be executed. Note the selection of the Overo’s IP address and server port in the GCS Communications widget. UDP data transmission was initiated when the test application was started and the GCS “Start” button in the communications widget was pressed.

Figure 2 is a screenshot comparing the GCS and GCS test application outputs for linear and angular state data. The received console is also updating in time to display the type and number of packets received. There are no discarded packets and the data can be seen to match that in the SSH connection terminal.

Figure 3 and Figure 4 are similar demonstrations of the GCS’s ability to receive autopilot status data and RC commands using UDP. The data in the data plotter, system status and flight control widgets can be seen to agree with that being send from the test application. It is also important to the note the successful transmission messages in the Transmitted Packets terminal. The GCS considers the gains and parameters properly received as the UDP library has implemented acknowledgement messages. Autopilot mode configuration can also be seen to have occurred as the GCS Flight Control widget is displaying active lights on the loops ticked as active. The lights are green based only on the periodic autopilot state UDP packets, which are clearly being received.

Figure 5, Figure 6 and Figure 7 represent the same data that has been previously displayed in the GCS software of Figure 2, Figure 3 and Figure 4 respectively. It has been established that the received data is consistent with that originally sent. The logged data is identical to the received data thus demonstrating the GCS is logging airborne system data.

# Conclusions

Analysis of the test results has concluded that the GCS is capable of configuring the autopilot and receiving and logging helicopter, autopilot and flight computer state dat. Selecting active control loops, updating controller gains and parameters has been demonstrated thus the GCS passes AT-02 and AT-08 and meets requirements SR-B-02 and SR-B-08. Successful display of the system state data fulfils AT-09, AT-17 and AT-18 thus the GCS meets SR-B-09, SR-D-07 and SR-D-08. Inspection of the log data also leads to the GCS passing AT-02 and AT-08 and therefore meeting SR-B-02 and SR-B-08.

# Recommendations

It is recommended the acceptance test results be made available as a final deliverable to the customer. If major changes are made to the GCS or network code these tests will need to be repeated.