

MRO Delay Line

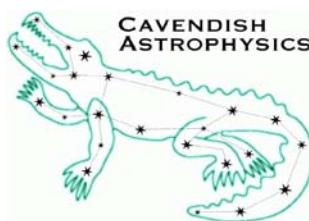
Delay Line Analysis GUI Functional Description

Document No. INT-406-VEN-1006

Martin Fisher

rev 1.0

22 Jan 2010



Cavendish Laboratory
JJ Thomson Avenue
Cambridge CB3 0HE
UK

Change Record

Revision	Date	Authors	Changes
0.1	2010-01-13	MF	First draft
0.2	2010-01-18	MF	Minor changes
0.3	2010-01-20	MF	General revision of section content. Included examples of list box menus, a new GUI figure and a section on time and plot synchronisation.
1.0	2010-01-22	MF	Corrections and re-arrangement of figures

Objective

This document describes the functionality of the Delay Line Analysis GUI and some aspects of the software used in creating and developing it.

Scope

The functionality of the Delay Line Analysis GUI is presented and aspects of the MATLAB implementation are described. Software design is not discussed as the application is developed in and runs under the MATLAB environment. The level of documentation that will support the delivered application is described.

Reference Documents

RD1 Requirement specifications for the MROI “production” delay line software ([INT-406-CON-0101](#))

RD2 MATLAB documentation (available at www.mathworks.com/support/) and specifically ‘Creating Graphical User Interfaces’ at www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/buildgui.pdf

RD3 Prototype Trolley Test Results ([INT-406-VEN-0109](#))

Applicable Documents

AD1 Network Message Protocols and Telemetry/Status/Logs File Format ([INT-406-VEN-1007](#))

Acronyms and Abbreviations

DL Delay lines

FITS Flexible Image Transport System

HDU Header Data Unit

HU Header Unit

GUI Graphical User Interface

ISS Interferometer Supervisory System

MROI Magdalena Ridge Observatory Interferometer

OPD Optical Path Difference

PSD Power Spectral Density

UI User Interface

UTC Universal Time Coordinated

Contents

- Change Record..... 2
- Objective 2
- Scope..... 2
- Reference Documents 2
- Applicable Documents..... 2
- Acronyms and Abbreviations..... 2
- Contents 3
- 1 Introduction..... 4
- 2 Delay Line Analysis GUI..... 4
 - 2.1 The Delay Line FITS log File 6
 - 2.2 The GUI layout 6
 - 2.2.1 File information UI panel..... 6
 - 2.2.2 Quick plot UI panels 6
 - 2.2.3 Signal plotting UI panels..... 7
 - 2.2.4 Analysis UI panel..... 7
 - 2.2.5 Time and synchronization..... 7
 - 2.3 Using the GUI 8
 - 2.4 Examples of contents of GUI list boxes..... 12
 - 2.5 Tag Property Names and Actions 15
 - 2.5.1 Tag Property Names – GUI Areas and controls 15
 - 2.5.2 Tag Property Names – menu functions..... 17
- 3 Documentation 18
 - 3.1 Scripts 18
 - 3.1.1 The GUI components..... 18
 - 3.1.2 Standard Scripts 18
 - 3.1.3 Function Scripts 18
 - 3.2 The Contents file..... 19
 - 3.3 Installation and the MATLAB ‘path’..... 19
 - 3.4 Extending the GUI 19

1 Introduction

The Delay Line FITS log analysis GUI is a stand-alone application that can read the DL FITS log files archived by the delay line workstation and display the data in a number of ways. The GUI is a MATLAB application that generates a single GUI window (in addition to the normal MATLAB windows) and can be run on any computer which has MATLAB installed or which has remote access to run MATLAB.

MATLAB is a widely used high-performance programming language and data visualisation tool for technical computing. It is interactive and has many built-in functions and features as well as being supported by a wide range of specialist *toolboxes*. In particular it has special provisions for building user interfaces which can be used to simplify complex processes or procedures that are suited to a particular application. More information about MATLAB including all the product documentation can be found on-line at www.mathworks.com

The minimum MATLAB requirements to run the GUI are MATLAB plus the Signal Processing Toolbox.

The Delay Line Analysis GUI has already been produced but is only partly functional. It has been used for assessing the performance of the prototype trolley but any special analysis has been done from the MATLAB command line. The delivered GUI will be complete and include the most useful functions so that no special knowledge of MATLAB will be required to carry out analysis of data obtained under the use cases defined in RD1. Provision is made, though, to export chosen data to the MATLAB workspace for further analysis using the full range of functions available in the MATLAB installation..

This document provides a description of the GUI, its functionality and how it is intended to be used. There is some information on how the MATLAB GUI is produced and how the functions are implemented. There is no description of the software itself but there is a section describing what supporting documentation will be provided with the production version of the GUI.

2 Delay Line Analysis GUI

The DL Analysis GUI window, shown in Figure 1, is a MATLAB figure window containing UI panels for areas containing objects such as push buttons, list boxes and graphics display areas called axes. The software is developed using MATLAB's in-built 'GUIDE' facility to provide the GUI figure and an associated script file containing 'call-backs' and custom code for all the objects in the GUI. These are supported by custom written MATLAB scripts which are invoked within the call-back of a control or display provided in the GUI.

A custom pull-down menu bar features in the GUI window from which the DL FITS log file to be examined can be defined and the primary information loaded. The data of interest are then loaded and extracted ready for display according to the options selected in the rest of the GUI. There are also facilities to carry out commonly required operations on the data before display and also to export data to the MATLAB workspace for detailed analysis using MATLAB's in-built functionality.

Also included in the GUI is a standard MATLAB figure toolbar. This allows all the features available in a figure toolbar to be used. This is especially useful for zooming and panning the plots in the four axes areas of the GUI.

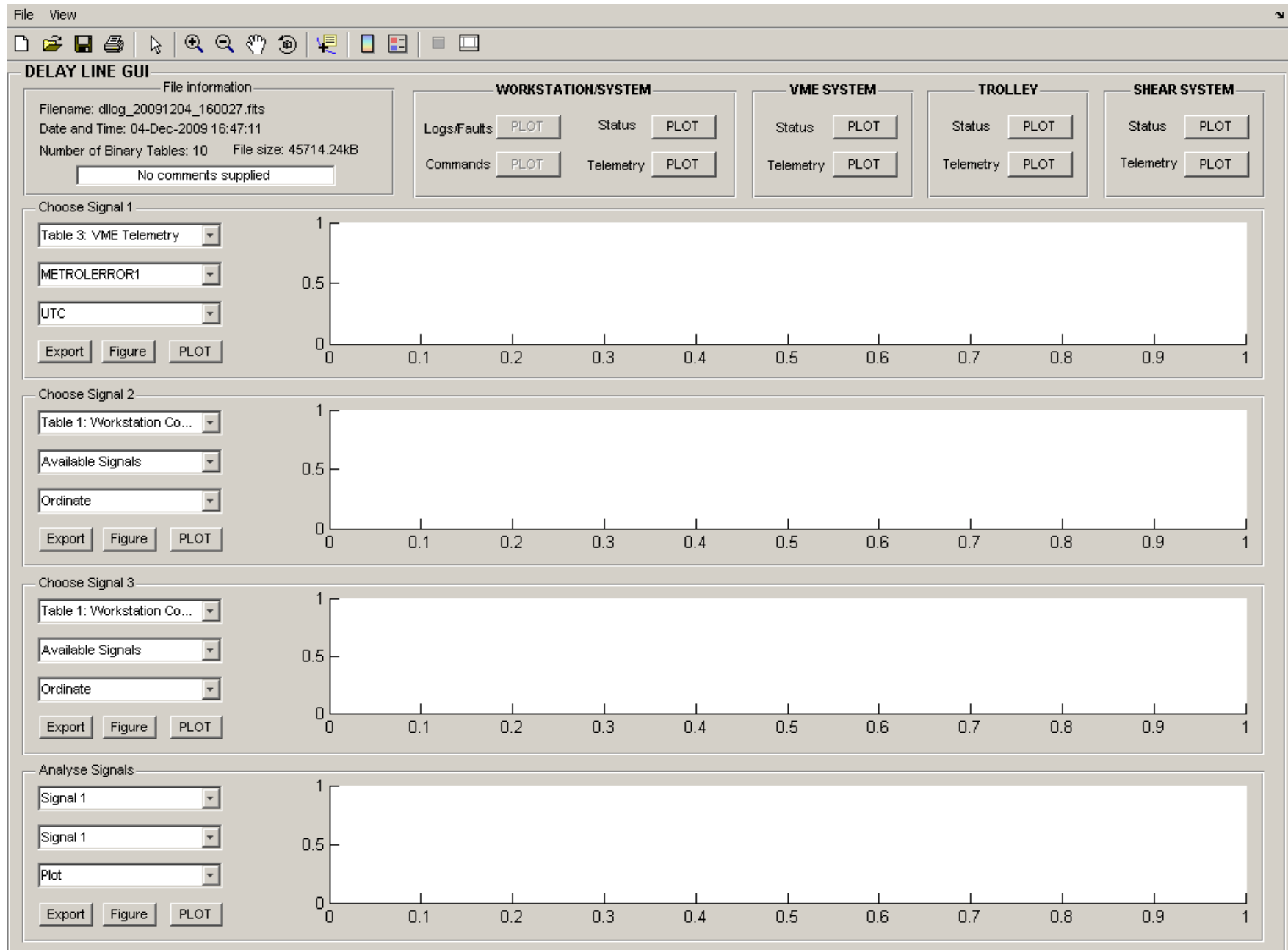


Figure 1 The DelayLineGUI layout

2.1 The Delay Line FITS log File

The DL FITS log files conform to the FITS standard and the header unit (HU) of the Primary HDU contains basic information about the file. The HU of each subsequent HDU comprising the file describe the detailed content and arrangement of the data in that HDU. The actual data is stored in binary tables according to a specially devised protocol, see AD1 for detailed information. The files are read using the built-in MATLAB functions *fitsinfo* and *fitsread* for interrogating the header units of FITS format files and reading in a specified binary table respectively.

2.2 The GUI layout

The GUI is laid out in a top-level UI panel called DELAY LINE GUI (see **Error! Reference source not found.**). Within this panel, at the top, are four ‘quick plot’ UI panels and below these are three identical ‘signal plotting’ UI panels and an ‘analysis’ UI panel. The menu bar of the GUI figure contains ‘file’ and ‘view’ pull-down menu items. The ‘file’ menu item contains options for opening a DL FITS log file, printer setup, printing the GUI and closing the GUI. The ‘view’ menu item currently contains one option which is to close all figures (except the GUI). Below the menu bar is the standard MATLAB figure toolbar which is useful for zooming, panning, editing and annotating plots in the axes areas of the UI panels. Descriptions of the individual UI panels are given in the following subsections.

2.2.1 File information UI panel

At the top of the GUI window, the first UI panel on the left presents information about the file which has been selected. This includes the filename, the date and time it was created, the file size and the number of binary tables it contains. Also, there is an ‘edit text’ box into which a comment (if provided in the file) is pre-loaded. The information in the edit box can be changed by left-clicking in the box with the mouse, allowing a comment describing the test to be entered if required. The information in this box is printed as part of the heading to each figure that is generated by subsequent plotting actions. The comments, if any, in the DL FITS log file are not modified.

2.2.2 Quick plot UI panels

The remaining UI panels from left to right contain ‘quick plot’ functions for the Workstation, VME (metrology) System, Trolley and Shear System. When a file is selected its header contents are examined and the information used to activate the PLOT buttons within these panels to show what subsystem data is available. Actuating a plot button causes the complete set of plots of associated data to be produced in a series of standard MATLAB figure windows. These windows have controls for printing, saving, exporting, editing, zooming and panning the plots contained within them as well as to carry out some basic data analysis functions. These figure windows remain open unless specifically deleted and are not overwritten by subsequent plots.

For each subsystem the current software assumes that there is just one delay line and that a particular set of data is available in a particular order and arranges the plots for that subsystem in sensible groups. The time axes for these plots are not synchronised and start from the first data point available. In practise the start of all data streams should be concurrent to within one second. The production version of this software will handle any number of delay lines and not assume any particular order or presence of data (this complicates plotting data in sensible groups in MATLAB figures with multiple axes and so may not always be possible if data streams are removed or added). However it will attempt to synchronize plots using information available from the FITS header and Binary Table extensions. Section 2.2.5 discusses how time and synchronisation will be handled.

2.2.3 Signal plotting UI panels

The three signal plotting UI panels are used to select and display particular signal data from the DL FITS log file, choosing the abscissa and ordinate as required. Once the signal is selected it is available to be acted on by the three pushbuttons in the panel and also to the analysis UI panel.

Each signal plotting UI panel contains three list boxes which are populated according to the DL FITS log file contents. When a file is selected the first list box is populated with a list of the available subsystems. When a selection is made from this menu the remaining two list boxes in the panel contain the available signals from the subsystem. Examples of list box contents are given in section 2.4. The middle list box is the abscissa for the plot and the lower list box is the ordinate. The plot is produced in the associated axes area of the panel when the **PLOT** pushbutton is actuated. The data is automatically scaled to fit in the axes but there are controls to modify, zoom or pan the plot. It is useful to be able to transfer the plot to a standard MATLAB figure, which has more tools built in, and so release the GUI axis area to be used for another plot. The transfer is accomplished and the new figure produced when the **Figure** pushbutton is actuated.

The **Export** pushbutton is provided to allow the user to transfer the data to the MATLAB workspace for more complicated processing or analysis. The MATLAB user can then, at the command line window use any of the MATLAB functions, tools (such as *Time Series Tool* called by `tstool`), user-written functions, or other toolbox functions.

2.2.4 Analysis UI panel

This panel is very similar to the signal plotting UI panels except that the contents of the list boxes are different. The upper and middle list boxes are populated with the choice of the available signal abscissas of the three signal plotting UI panels and UTC. The upper list box is the abscissa for the new plot and the middle list box is the ordinate. This allows data from one subsystem to be plotted against data from another subsystem or against UTC, whichever is desired. The data can be plotted in the axes area, sent to a new MATLAB figure and/or exported to the MATLAB workspace as described earlier.

The lower list box provides a choice of functions to be carried out on the selected data. The functions currently envisaged are as follows:

Extract: extract a contiguous section of the data and set as the default ordinate and abscissa.

Subtract: subtract the default ordinate and abscissa and plot against UTC.

Stats: perform a set of simple statistics on the default abscissa and display them in the axes area.

Power Spectrum: compute the PSD of the default abscissa and display in the axes area.

Integrated Power: compute the integrated PSD of the default abscissa and display in the axes area.

Auto-correlation: compute the auto-correlation function of the default ordinate.

Cross-correlation: Compute the correlation of the default abscissa and ordinate.

Transfer Function: compute the transfer function of the default abscissa and ordinate then display in the axes area.

2.2.5 Time and synchronization

Subsystem time is set using NTP and therefore subsystems can only be synchronized to about one millisecond. This means that, while data from the same subsystem can be synchronized when plotting, data from different subsystems cannot be synchronised to better than one millisecond. In order to

'align' plots of data from different subsystems to the millisecond level of synchronisation time is obtained from the FITS header and binary table extension information which is provided by the MATLAB function '*fitsinfo*'. This avoids having to read in every binary table just to find the UTC of the first data point. The keyword DATE_OBS in the binary table extension information provides the date and UTC (to three decimal places of second) of the start of measurement for that particular binary table. The start time chosen from one of the binary tables can be defined as the zero point or reference time to which the start time from all the other tables is adjusted when plotted (or exported).

The GPS time module resides in the VME system and so the most accurate time will be that supplied with data from that system. Because the VME system is likely to be operational for most DL FITS logs, the reference time will be derived from the DATE_OBS keyword of the VME Telemetry data. In the event that this is not available then the reference time will be derived from the earliest DATE_OBS time in the DL FITS log.

2.3 Using the GUI

To select a DL FITS log the file menu at the top of the GUI window is pulled down and the OPEN choice is selected. This opens a standard file selection window from which the required file can be selected. The information from the FITS header and binary table extensions is imported into MATLAB and displayed in the *File Information* UI panel and also, in more detail, at the MATLAB command window. The UI panel presents information about the file which has been selected, including the filename, the date and time it was created, the file size and the number of binary tables it contains. The 'edit text' box will contain a comment (if one was provided in the file) or some default text. The information in the edit text box can be changed at any time by left-clicking in the box with the mouse and entering/deleting text. The information in this box is printed as part of the heading to each figure that is generated by subsequent plotting actions and so comments should be short but sufficient to identify the nature of the test.

It is possible that DL FITS logs may contain more than one binary table of a particular type because another recording was made without changing the filename. This is detected automatically when a DL FITS log is selected, appearing in the Source Selection list box and also in the information in the MATLAB command window. It is not evident as far as the quick plot button are concerned and so if more than one table of a particular type exists then whenever a quick plot is requested a dialog box will appear asking the user to choose which table should be loaded. Plots are then produced using the data from the table that has been loaded. A similar dialog box will appear if there is more than one of a type of subsystem to choose from.

The GUI can be used to quickly assess the performance of the whole system by importing the DL FITS log of a particular test and then using the quick plot buttons to generate the plots of interest. As these plots are MATLAB figures then all the standard figure tools are available to manipulate, analyse, print and save the plot. Plots remain open unless specifically closed so it is possible to have one particular plot from a series of DL FITS log files open for direct comparison.

For a more specific analysis e.g. to examine the PSD of some portion of the metrology error one would load the DL FITS log file, get the appropriate signal into one of the signal plotting UI panels and then select that signal in the analysis UI panel. The 'Extract' function is then selected and the portion of the signal required is defined and plotted in the axes area. The 'Power Spectrum' function is then selected and the result replaces the plot in the axes area. Pressing the 'Figure' button plots the same result in a MATLAB figure, which is easier to examine, and also allows the analysis UI panel to be used for

another plot, e.g. the Integrated Power function, or to operate on a different set of data.

For overall performance assessment the most appropriate plot, an example of which is shown in **Error! Reference source not found.**, is available from the VME metrology telemetry subset. This provides an assessment of metrology error (converted to OPD) against performance criteria derived from the DL requirements and used as part of the acceptance test procedures (see appendix to RD3). Also, apart from the plots of each telemetry item there are also plots of the power spectral density and integrated power of the metrology error which inform about periodic disturbances and what contributes most to the metrology error. An example output from a trolley test is shown in **Error! Reference source not found.**

The following is an example of the output which appears in the MATLAB command window when a DL FITS log file is selected in the GUI window.

File information.....

```

Filename: 'C:\Matlab Files\Cavendish\Log Files\dllog_20091215_112110.fits'
FileModDate: '15-Dec-2009 11:54:51'
FileSize: 28290240
Contents: {1x11 cell}
PrimaryData: [1x1 struct]
BinaryTable: [1x10 struct]

```

There are 10 Binary Tables with Name and Source/Client:

```

Table 1 DL_CMD WKSTN
Table 2 DL_LOG N/A
Table 3 DL_TELEMETRY VME
Table 4 DL_TELEMETRY SHEAR1
Table 5 DL_TELEMETRY TRLY1
Table 6 DL_TELEMETRY SYSTEM1
Table 7 DL_STATUS SHEAR1
Table 8 DL_STATUS VME
Table 9 DL_STATUS TRLY1
Table 10 DL_STATUS SYSTEM1

```

FITS header user-supplied comments (if any):

```
'looking for diff pos interference'
```

If a second recording had been made before the file was closed then another set of status and telemetry tables would have been added and in this case the following lines would appear after Table 10 line above:

```

Table 11 DL_TELEMETRY VME
Table 12 DL_TELEMETRY SHEAR1
Table 13 DL_TELEMETRY TRLY1
Table 14 DL_TELEMETRY SYSTEM1
Table 15 DL_STATUS SHEAR1
Table 16 DL_STATUS VME
Table 17 DL_STATUS TRLY1
Table 18 DL_STATUS SYSTEM1

```

dlog-20080806-144929.fits - OPD error. Test description: Holding station (in vacuum). **(Test Run passed)**

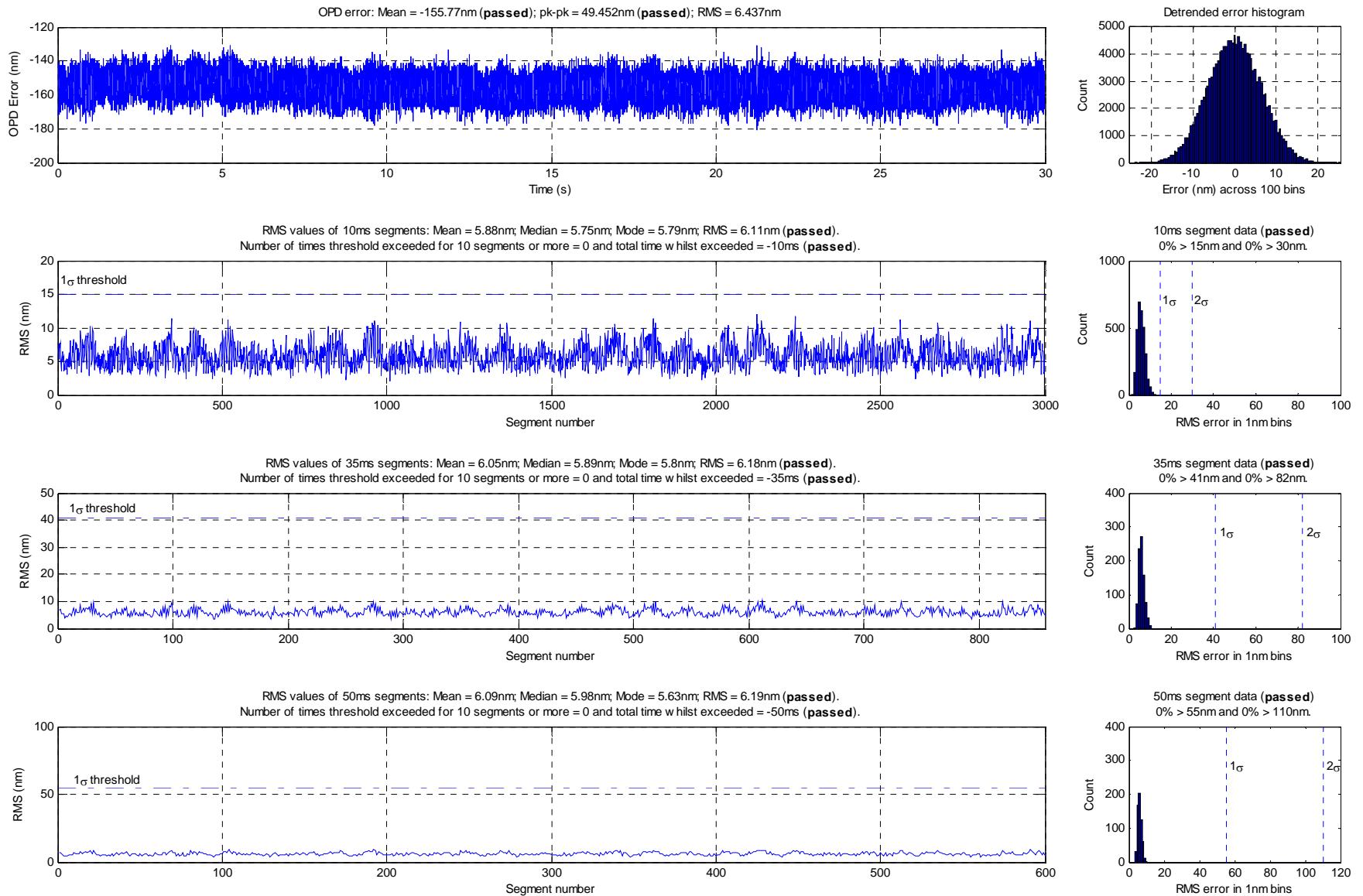


Figure 2 Example output of test criteria plot generated from metrology error data

dlog-20080806-144929.fits - VME telemetry data. Test description: Holding station (in vacuum)

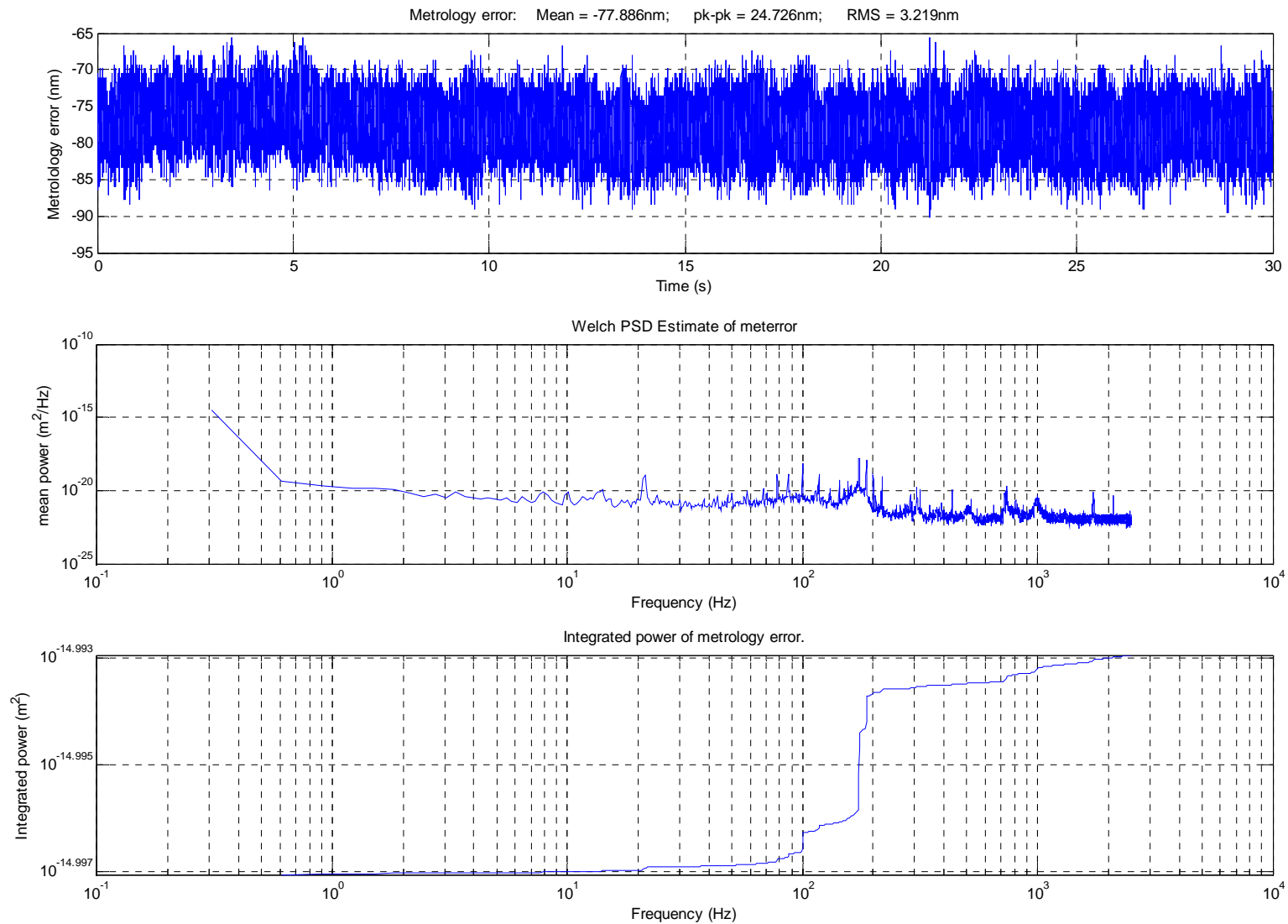
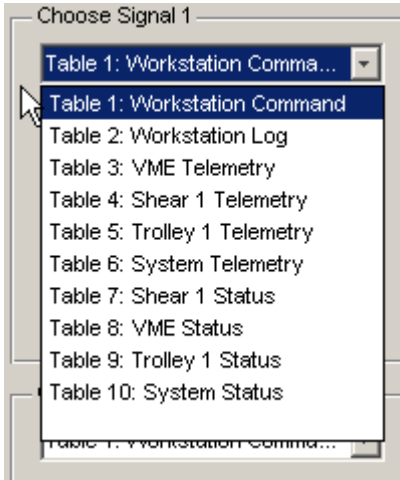


Figure 3 Example output from one of the VME telemetry figures produced by the quick plot button.

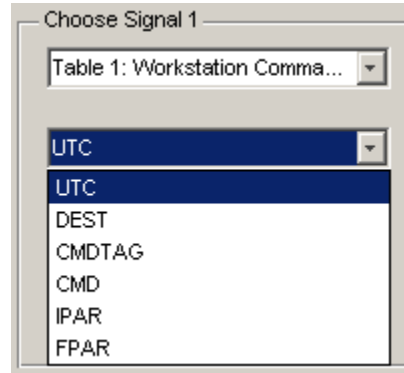
2.4 Examples of contents of GUI list boxes.

The following examples show the contents of list boxes for each subsystem. Note that the contents of the Source Selection list box depend on which binary tables are present in the DL FITS log while the contents of the other list boxes depend on what data is available in the binary tables.

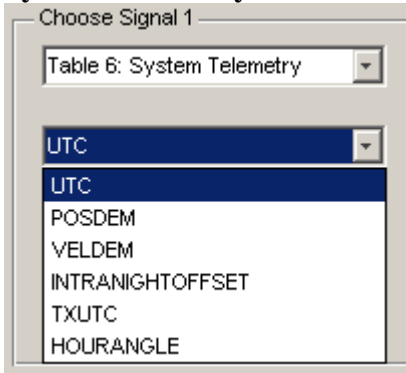
Source Selection List Box



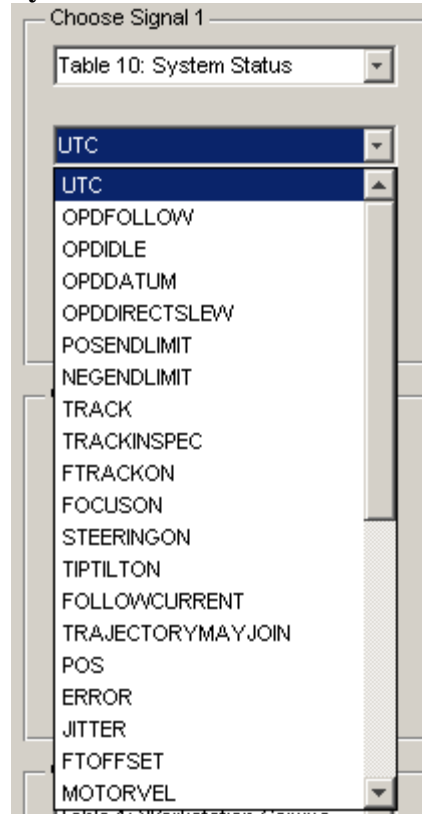
Workstation Commands List



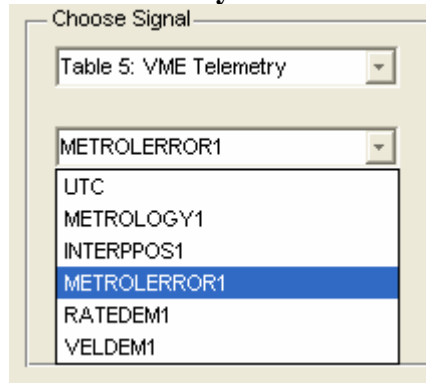
System Telemetry List



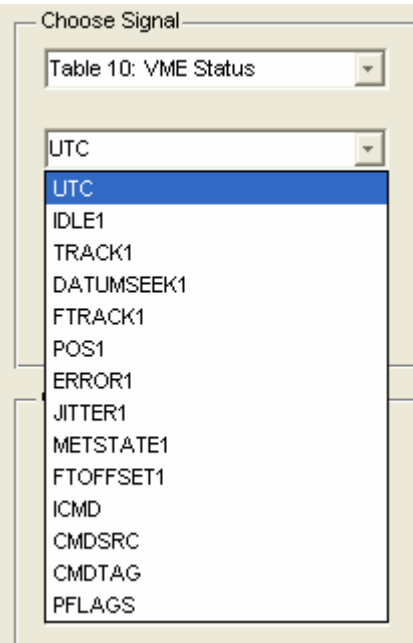
System Status List



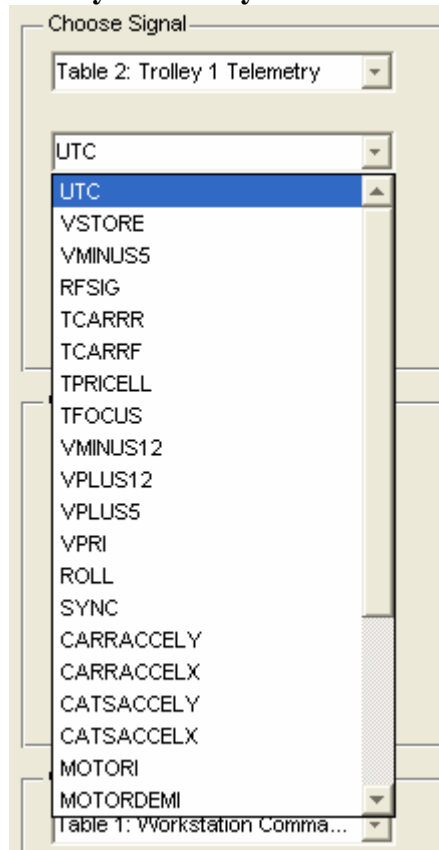
VME Telemetry List



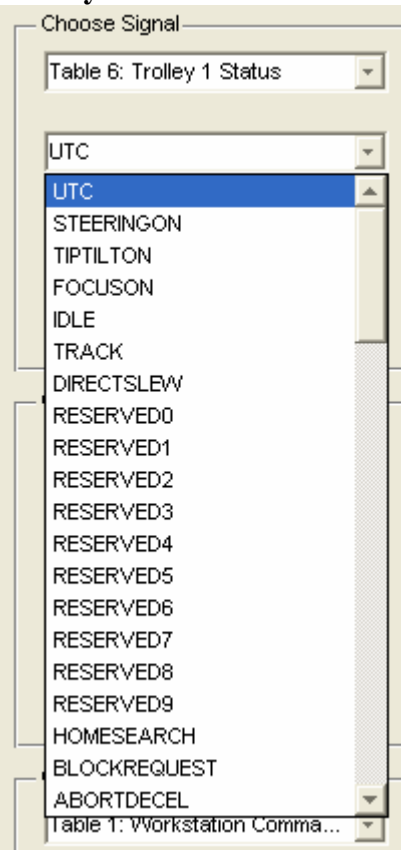
VME Status List



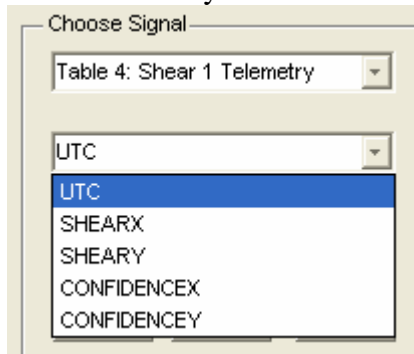
Trolley Telemetry List



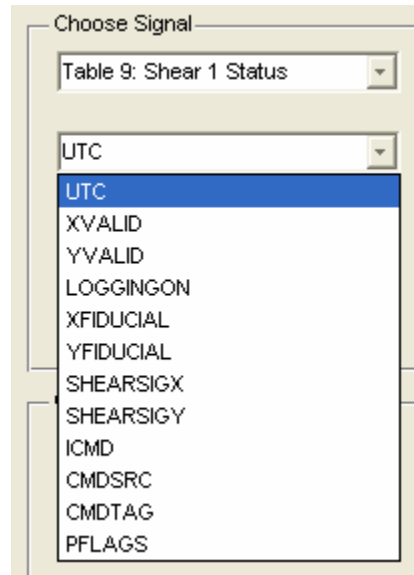
Trolley Status List



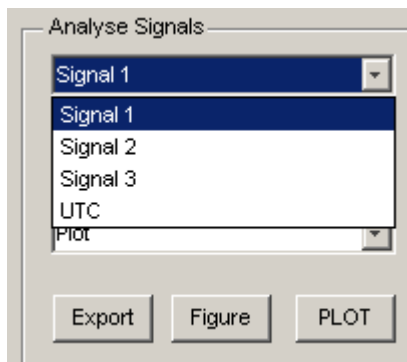
Shear Telemetry List



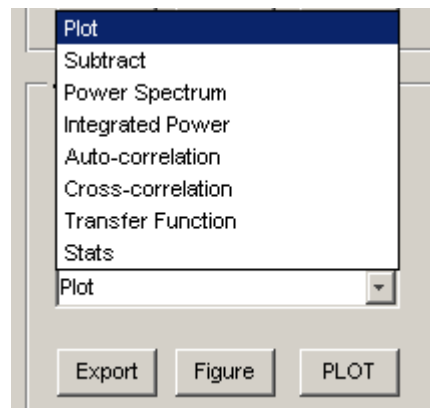
Shear Status List



Analyse Signals List



Plot List



2.5 Tag Property Names and Actions

The following tables list the component objects of the GUI panel and the actions initiated when a control is operated.

2.5.1 Tag Property Names – GUI Areas and controls

GUI Area	Object	Component	Tag	Actions
File information	Test description	Edit box	edit1	Presents default text or text gathered from file comment. Box is editable by left clicking. The content of the box is incorporated into plot headings.

GUI Area	Object	Component	Tag	Actions
Workstation/System quick plot	Logs/Faults Plot	Pushbutton	pushbutton1	Opens an interactive figure showing log and fault events from subsystems to the workstation plotted against time. Details of nearest event appear when mouse cursor moved over plotted data. Can also spawn a scrollable window listing all events.
	Commands Plot	Pushbutton	pushbutton2	Opens an interactive figure showing command events to subsystems from the workstation plotted against time. Details of nearest event appear when mouse cursor moved over plotted data. Can also spawn a scrollable window listing all events.
	System Status Plot	Pushbutton	pushbutton3	Loads in the appropriate Binary Table Calls the appropriate workstation status plotting scripts: PlotWkstnStatus1_gui PlotWkstnStatus2_gui
	System Telemetry Plot	Pushbutton	pushbutton4	Loads in the appropriate Binary Table Calls the appropriate workstation telemetry plotting script: PlotWkstnTelem_gui

GUI Area	Object	Component	Tag	Actions
VME System quick plot	Status Plot	Pushbutton	pushbutton5	Loads in the appropriate Binary Table Calls the appropriate VME status plotting script: PlotVmeStatus_gui
	Telemetry Plot	Pushbutton	pushbutton6	Loads in the appropriate Binary Table PlotVmeTelem_gui

GUI Area	Object	Component	Tag	Actions
Trolley quick plot	Status Plot	Pushbutton	pushbutton7	Loads in the appropriate Binary Table Calls the appropriate Trolley status plotting script: PlotTrlyStatus_gui
	Telemetry Plot	Pushbutton	pushbutton8	Loads in the appropriate Binary Table Calls the appropriate Trolley telemetry plotting script: PlotTrlyTelem_gui

GUI Area	Object	Component	Tag	Actions
Shear System quick plot	Status Plot	Pushbutton	pushbutton9	Loads in the appropriate Binary Table Calls the appropriate Shear System status plotting script: PlotShearStatus_gui
	Telemetry Plot	Pushbutton	pushbutton10	Loads in the appropriate Binary Table Calls the appropriate Shear System telemetry plotting script: PlotShearTelem_gui

GUI Area	Object	Component	Tag	Actions
Choose signal 1 Choose signal 2 Choose signal 3	Available Data	Pop-up Menu	popupmenu11	1. populates reference signal popup menu 2. populates ordinate popup menu 3. loads binary table (if not already loaded) 4. defaults ordinate to UTC
	Available Signals	Pop-up Menu	popupmenu12	1. gets the signal data 2. adjusts signal according to reference time 3. plots signal versus UTC
	Ordinate	Pop-up Menu	popupmenu13	1. gets a new ordinate signal for the plot 2. checks if ordinate is compatible 3. adjusts for the reference time if necessary 4. interpolate/decimate as necessary
	Analyse	Pushbutton	pushbutton11	1. places data in the MATLAB workspace 2. transfers data to Analysis GUI
	Figure	Pushbutton	pushbutton12	1. spawns a MATLAB figure and plots the data
	PLOT	Pushbutton	pushbutton13	1. updates the current plot based on the reference time

GUI Area	Object	Component	Tag	Actions
Analysis Axes	Available Selection Abscissa	Pop-up Menu	popupmenu11	<ol style="list-style-type: none"> gets the selected signal data from choice 1 - 3 defaults ordinate to UTC
	Available Selection Ordinate	Pop-up Menu	popupmenu12	<ol style="list-style-type: none"> gets the selected signal data from choice 1 – 3 or UTC checks if ordinate is compatible interpolate/decimate as necessary replaces UTC ordinate with new ordinate
	Function	Pop-up Menu	popupmenu13	<ol style="list-style-type: none"> sets the function to be used on abscissa and ordinate checks that data is available according to function
	Export	Pushbutton	pushbutton11	<ol style="list-style-type: none"> places data in the MATLAB workspace
	Figure	Pushbutton	pushbutton12	<ol style="list-style-type: none"> spawns a MATLAB figure and plots the data
	PLOT	Pushbutton	pushbutton13	<ol style="list-style-type: none"> plots/updates the current plot
	Plot axes	Axes	axes1	Object for the current plot

2.5.2 Tag Property Names – menu functions

GUI Area	Object	Component	Tag	Action
Menu bar [File]	File Menu	Menu	FileMenu	Opens the drop down list of file menu items
	Open File	Menu item	OpenFile	Presents Open File dialog box OK causes file header to be loaded <ol style="list-style-type: none"> File details displayed in GUI and command window File comment (if any) is displayed in edit text box Table list is displayed in command window Quick plot buttons are activated for available data Available data pop-up menus are populated A ‘zero reference time’ is established
	Print Setup	Menu item	PrintSetup	Presents Print Setup dialog box
	Print	Menu item	PrintWindow	Presents Print dialog box
	Close Gui	Menu item	CloseGui	Presents Close GUI dialog box CANCEL closes dialog box OK closes GUI but not spawned figures
Menu bar [View]	View Menu	Menu	view_menu	Opens the drop down list of view menu items
	Close all figures	Menu item	close_figs	Closes all open figures, however created.

3 Documentation

The documentation to be provided with the production version GUI is described below

3.1 Scripts

3.1.1 The GUI components

A MATLAB GUI requires at least two files to exist, the GUI *figure* and the GUI *script* which contains a block of code or ‘call-back’ for each control. They can both be created using the MATLAB programming language in the MATLAB editor or using a built-in tool called GUIDE. If the GUI is created using the GUIDE tool, then these two files are created automatically and the call-backs within the GUI script are then modified by hand to include extra code and/or to call separate scripts written to produce the desired response for each control. When the GUI script file is automatically generated it includes a certain level of automatic embedded documentation. Any additional code introduced into the GUI will also be accompanied by a suitable level of embedded documentation.

3.1.2 Standard Scripts

These are individual MATLAB script files (m-files) called by the selection of pushbuttons in the GUI or within other scripts. Scripts are documented with header comments describing the action of the script and the inputs required. Further more detailed comments will be embedded in the script. A description of the purpose and actions of each script will be provided in a software description document.

3.1.3 Function Scripts

Function scripts contain a special header which identifies them so that they act like existing MATLAB functions and are called with input and output parameters as defined in the script. Scripts are documented with header comments describing the action of the script, the inputs required and the outputs provided. For clarity, further comments may be embedded in the script. The actions of these scripts are also defined in the software description document and an example of how this is presented is given below for the simple function ‘stats’.

Function	avedrift	Local Variables
Purpose	This function calculates the average, peak-to-peak and standard deviation of a signal	
Syntax	[a,b,c] = stats(X) a, b and c are scalar outputs, respectively the average, peak-to-peak and standard deviation. X is an input vector	Inputs: x Outputs: ave, pk2pk, stdev
Called by	Script PlotVmeTelem2_gui.m Script PlotVmeTelem3_gui.m	
Actions	The mean value of x is calculated The peak to peak value of x is calculated The standard deviation of x is calculated	ave pk2pk stdev

3.2 The Contents file

MATLAB has a facility which builds a contents file from all the script files gathered in a directory. The contents file, which resides in the same directory, lists each filename together with the first line of the script. This provides a basic facility for checking that all the necessary files which support the GUI are present in the directory.

3.3 Installation and the MATLAB 'path'

The minimum MATLAB requirements to run the GUI are MATLAB plus the Signal Processing Toolbox.

Any scripts that are called must be on the MATLAB path so that MATLAB knows what to do when it encounters a function or name which is not inherent. The GUI software should be contained in one directory and that directory must be added to the MATLAB path. Also, the location of DL FITS log files must be on the MATLAB path. This is best achieved by recursively adding the main directory for DL FITS log files to the MATLAB path when calling the GUI. Instructions are included and documented within the calling script for the GUI i.e. DelayLineGUI.m

3.4 Extending the GUI

A section describing how to extend the analysis functions within the GUI will be included in the software documentation. This will explain how to add further options to pull-down menus by editing the GUI script and calling a new script if necessary. If the GUI figure is to be modified then the appropriate MATLAB documentation must be referenced.