MRO FTT/NAS & FLC

Software User Manual

MRO-MAN-CAM-1160-0165

John Young <jsy1001@cam.ac.uk>

rev 1.0

4 April 2016

Cavendish Laboratory
JJ Thomson Avenue
Cambridge CB3 0HE
UK

Change Record

Revision	Date	Author(s)	Changes
0.1	2014-08-01	JSY	Initial version
0.2	2016-02-23	JSY	Updates
0.3	2016-04-01	EBS	Updated configuration, system software
0.4	2016-04-01	EBS	Updated GUI images
1.0	2016-04-04	JSY	Expanded GUI documentation

Objective

To describe the operation of the FLC and FTT/NAS software from the user's point of view.

Scope

This document describes how to start and operate the FLC and FTT/NAS software, using the supplied control GUI applications. Deployment scenarios where the FTT software is controlled via the MRO ISS are beyond the scope of this document.

Reference Documents

[RD1] *FTT/NAS Software Preliminary Design Report*, rev 1.0, Apr. 2012, ID: `MRO-TRE-CAM-1160-0143`.

[RD2] J. S. Young et al., "The MROI fast tip-tilt correction and target acquisition system", in: *Proc. SPIE* 8445, 84451V (2012).

[RD3] *FLC to ISS ICD*, rev 1.14, Feb. 2016, ID: `MRO-ICD-CAM-1200-0113`.

[RD4] *FTT/NAS to ISS ICD*, rev 1.14, Feb. 2016, ID: `MRO-ICD-CAM-1100-0112`.

[RD5] J. Young, *Software Release Notes*, rev 1.5, Feb. 2016, ID: `MRO-MAN-CAM-1160-0163`.

[RD6] *FTT/NAS Electronics to FTTA Controller ICD*, rev 1.0, May 2013, ID: `MRO-ICD-CAM-1100-0108`.

[RD7] J. Young, *The dlmsg FITS convention*, rev 0.4, Sept. 2014, ID: `MRO-MAN-CAM-1161-0166`.

Acronyms and Abbreviations

API Applications Programming Interface
EMCCD Electron Multiplying Charge Coupled Device
FTT Fast Tip-Tilt
FTTA Fast Tip-Tilt Actuator
FLC First Light Camera
GSI Generic System Interface
ICD Interface Control Document
ISS Interferometer Supervisory System
MCDB Monitor and Configuration DataBase
MROI Magdalena Ridge Observatory Interferometer
NAS Narrow-field Acquisition System
NMT New Mexico Tech
ROI Region Of Interest
SDK Software Development Kit
TBC To be confirmed
TBD To be determined

Table of Contents

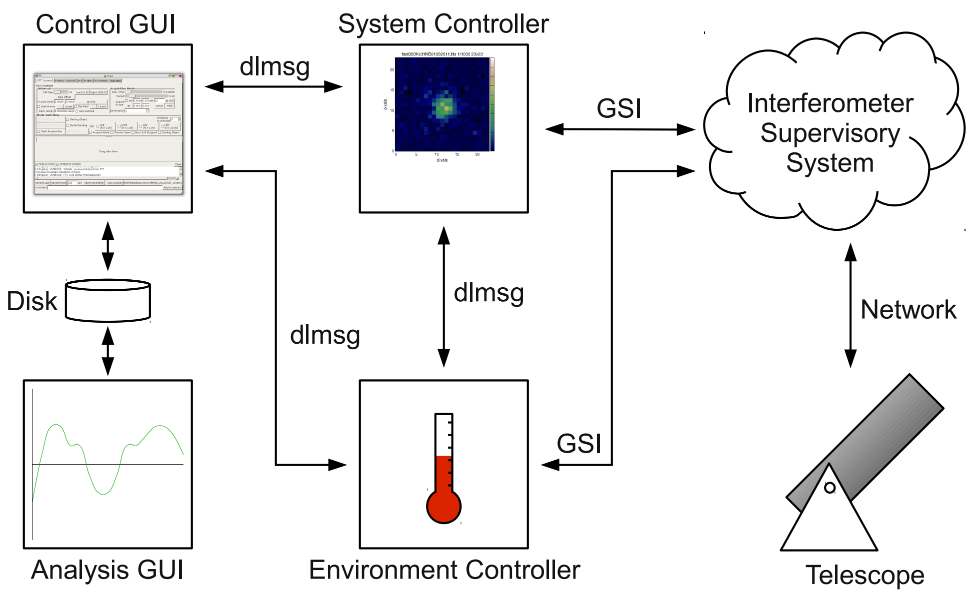

Figure 1: Overview of the FTT/NAS and FLC software. The software can be used in a "standalone" mode without the Interferometer Supervisory System (ISS).

1 Introduction

This document is part of the MROI FTT/NAS software release, and describes how to run the FLC or FTT system using the supplied graphical user interface applications. The overall architecture of the software is described in the PDR report [RD1] and our SPIE conference paper from 2012 [RD2].

The software comprises two components used to control the FTT/NAS hardware, a user interface software application, and a fourth component used for offline visualization and analysis of previously-recorded monitor data. These four components are shown in Fig. 1. The control software is partitioned into two components because of the need to maintain the environmental conditions within the FTT/NAS camera enclosure at all times. The application that controls the camera environment and enables/disables the camera is the "environment controller", and the component that performs the primary FTT/NAS functions such as target acquisition and fast tip-tilt correction is the "system controller".

The environment controller is a simple sensing application which reads and reacts to data on timescales of the order of one second, and hence runs in ordinary Linux. The application communicates with sensors and a heater via two USB-driven Labjack U3 boards. If the camera thermal enclosure is not installed, as expected for the FLC, there is no need to run the environment controller.

The control/display GUI provides a graphical user interface for commanding the system and environment controllers and for live display of their monitor data (including camera images). It can also record monitor data to a set of FITS-format files when requested to do so by the operator. The control/display GUI may be used in one of two modes: a fully-functional "standalone" mode, and a "display-only" mode which can safely be used when the FTT/NA system is under the control of the central ISS. There are two editions of the GUI, which can be used interchangably with the system and environment controllers:

FTT/NAS GUI Provides access to the full functionality of the FTT/NAS system, including the fast tip-tilt correction mode.

FLC GUI Provides access to the FLC functionality (a subset of the full FTT/NAS functionality) as defined in the ICD [RD3]. Compared with the FTT/NAS GUI, has additional data handling capabilities requested by AMOS for the purpose of UTM commissioning:

- Ability to record scalar data (e.g. unaveraged centroid estimates) to CSV files as well as FITS files.
- Scrolling display of recent average and rms centroid estimates.

The analysis GUI is used to visualize and process monitor data previously recorded to FITS files by the control/display GUI. The application provides flexible capabilities for displaying time series of scalar data and images.

The system controller software performs acquisition functions, fast tip-tilt servo loop closure and transmission of system diagnostic information (including camera images) over the network. The core function of the software is closure of the servo loop between the EMCCD camera and the fast tip-tilt mirror, with bounded latency. The real-time software reads a camera image, calculates a centroid and then a correction, and sends the correction to the controller for the mirror. Meanwhile the image data and corrections are buffered into an ordinary Linux application, where they are used for non-real-time tasks and logging.

The commands accepted by the system controller and environment controller, as well as their configuration parameters and the monitor data they publish are enumerated in the Interface Control Documents [RD3, RD4]. This document describes the graphical interface to these commands provided by the control/display GUI.

2 Installation

The procedure for building and installing the software is given in the release notes [RD5]. In this manual, it is assumed that the software has already been installed on the FTT/NAS computer, the embedded processor for the environment controller, and on any other computer being used to run the control GUI and analysis GUI.

3 Orientation

The structure of the source tree is described in the release notes [RD5]. Installing the software copies files to a number of locations as described below. In this section, `[kernel-release]` stands for the "kernel release" as output by the command `uname -r`.

`/usr/local/bin` contains the executables for the system controller, environment controller, and control GUI. Note that test executables are not installed and must be run from the source tree. There are several *variants* of the system and environment controllers as explained in Section 5 and in the FTT/NAS ICD [RD4].

`/usr/local/share/flcgui` contains the installed configuration files (`.ini`) and widget definition files (`.glade`) for the FLC control GUI.

`/usr/local/share/fttgui` contains the installed configuration files (`.ini`)and widget definition files (`.glade`) for the FTT/NAS control GUI.

The FLC and FTT/NAS control GUI applications search for user-modified configuration files in `$HOME/.local/share/flcgui` and `$HOME/.local/share/fttgui` respectively. If found, these files take precedence over the installed versions, and will not be overwritten next time the software is installed.

`/usr/local/share/enggui` and `/usr/local/share/controlgui` contain widget definition files (`.glade`) common to both editions of the control GUI.

`/usr/local/share/doc/ftt` contains the documentation.

`/usr/local/etc/andor` contains configuration files for Andor cameras. These get installed with Andor's software development kit and should not be edited.

`/lib/modules/[kernel-release]/kernel/drivers/xenomai` contains the Xenomai device drivers.

The configuration for the system and environment controllers is normally obtained over the network from the central ISS database (MCDB). Some of the system variants do not use this connection, but instead read the values of the system properties from local text files. As with the example system provided by MRO, these system property files are not installed; instead the full path to the file must be specified on the command line. This is explained fully in Section 5.1.

4 Configuration

4.1 System properties

As described above, system properties are configuration values that are either downloaded from the ISS database or loaded from a configuration file. When either of the system or environment controllers is run in standalone mode, a configuration file is used, and is read during system initialisation. Configuration files are human-readable and can be modified with a text editor.

The distribution contains two such files as examples:

`[ftt-root]/systems/FTTCamSystem/test/property.txt`
and `[ftt-root]/systems/FTTEnvSystem/test/property.txt`

for the system and environment controllers respectively. A description of all of the system properties within these files can be found in the FTT/NAS ICD [RD4].

During system execution, scalar system properties can be examined by issuing a `getFixedParams` call via the GSI. Boolean system properties are GSI monitor points and can be monitored accordingly. When using the Cambridge control GUI, scalar system properties appear in the FTT-PARAM and FTTENV-PARAM tabs, while boolean system properties appear in the FTT and FTTENV tabs with the other boolean monitor points.

4.1.1 Configuring for common hardware scenarios

Some system properties define whether the systems should use real hardware or internal emulators (here, "real hardware" can mean "the other controller"). If emulation is selected, any real connected hardware is ignored. If real hardware is selected, then system initialisation will fail if there is an error communicating with that hardware (for example, if the hardware is not connected or the necessary drivers are not installed).

The relevant system controller properties are as follows:

UseCamera Boolean ("true" or "false"). Enable/disable use of the real camera (otherwise an internal simulation is used).

UseIOCard Boolean ("true" or "false"). Enable/disable the interface to the FTTA.

UcamConnectToEnv Boolean ("true" or "false"). When true, the system will connect to the environmental controller and watch its ***CameraEnable*** monitor point. If that value should become false, the system will commence an orderly shutdown.

The relevant environment controller properties are as follows:

UseInterface Boolean ("true" or "false"). When true, the environmental controller will use the Labjacks. Otherwise, software emulation is used.

FTTShutdownEnable Boolean ("true" or "false"). When true, the environment controller is permitted to shutdown the camera system software and the FTT computer.

FTTPowerControlEnable Boolean ("true" or "false"). When true, the environmental controller is permitted to use the ethernet-switched mains power board within telescope enclosure control rack Q5 ([RD6]) to cut the power to the camera and the FTT computer.

Scenario 1: No real hardware, no communication between systems In this scenario, all hardware is emulated and communication between the system and environment controllers is disabled. This configuration is appropriate for running unit tests, where each controller is isolated from the other (Table 1):

System	System Property	Value
FTTCamSystem	UseCamera	false
FTTCamSystem	UseIOCard	false
FTTCamSystem	UcamConnectToEnv	false
FTTEnvSystem	UseInterface	false
FTTEnvSystem	FTTShutdownEnable	false
FTTEnvSystem	FTTPowerControlEnable	false

Table 1: Property settings for scenario 1: no real hardware, no communication between systems

Scenario 2: No real hardware, communication between systems In this scenario, all hardware is emulated but the system and environmental controllers can communicate. This scenario is useful for testing that the environment controller can shut down the system controller when it determines the environment is unsafe (Table 2):

System	System Property	Value
FTTCamSystem	UseCamera	false
FTTCamSystem	UseIOCard	false
FTTCamSystem	UcamConnectToEnv	true
FTTEnvSystem	UseInterface	false
FTTEnvSystem	FTTShutdownEnable	true
FTTEnvSystem	FTTPowerControlEnable	false

Table 2: Property settings for scenario 2: no real hardware, systems can communicate

Note that communication between the two systems is currently implemented using the Cambridge DLMsg interface. It is expected that this will be reimplemented using the GSI publish/subscribe system when that becomes available.

Scenario 3: Unit Telescope Commissioning In this scenario (Table 3), the only necessary peripheral hardware is the Andor iXon 897 EMCCD camera. There is no camera enclosure. Other peripheral hardware, if present, is disabled. This includes fast tip-tilt mirror control, environmental control and monitoring, and power board control.

System	System Property	Value
FTTCamSystem	UseCamera	true
FTTCamSystem	UseIOCard	false
FTTCamSystem	UcamConnectToEnv	false
FTTEnvSystem	UseInterface	false
FTTEnvSystem	FTTShutdownEnable	false
FTTEnvSystem	FTTPowerControlEnable	false

Table 3: Property settings for scenario 3: unit telescope commissioning

Scenario 4: Fully implemented except for power board control In this scenario (Table 4), all peripheral hardware is used. For the system controller, this consists of the Andor iXon 897 EMCCD camera and the Advantech PCI1716 analogue I/O PCI card. For the environmental controller, this is the two Labjack U3 interface units and the environmental sensors and heater control that connect to them. Other configurations are also possible, for example, enabling the camera but not the analogue I/O, or enabling both of these but not the environmental controller I/O.

It is also possible to enable control of the power board (an Eaton IPC3401-NET), however it is disabled in this example. Power board control software has been implemented, but uses a network protocol that is not supported by the board (the documentation was incorrect). If power board control is enabled, it will automatically be disabled after warnings are issued.

System	System Property	Value
FTTCamSystem	UseCamera	true
FTTCamSystem	UseIOCard	true
FTTCamSystem	UcamConnectToEnv	true
FTTEnvSystem	UseInterface	true
FTTEnvSystem	FTTShutdownEnable	true
FTTEnvSystem	FTTPowerControlEnable	false

Table 4: Property settings for scenario 4: fully implemented except for power board control

4.2 Control GUI configuration

The control GUI configuration file contains the following sections:

Files Pathnames of relevant files

Addresses IP names/addresses

FttSystem Parameters related to the system controller

FttEnvControl Parameters related to the environment controller

The GUI can be configured to prevent changes to the system state, by changing the “Control” value in the “FttSystem” or “FttEnvControl” section to “false”. This is in addition to any restriction enforced by the system server (see Section 5). The configuration file also contains the address and ports on which the system and environment controllers are assumed to be listening for connections. The default addresses are "localhost", and hence should be replaced with the appropriate IP addresses/names if the GUI is to be run on a separate computer.

The location for saving FITS files is given by the value of "LogParentDir" in the "Files" section. Note that the default location is a subdirectory of `/tmp`, from which all of the contents are automatically deleted when the computer is restarted! If for some reason multiple control GUI instances are being run on the same computer, they must be configured to record data in different directories.

Scenario A: Remote control, no environment controller In this scenario, the GUI is configured to control the system controller running on a separate computer, and not to attempt connecting to the environment controller (Table 5). This is a possible configuration for UTM commissioning.

Section	Key	Value
Addresses	FttSystem	(FTT rack-mount PC address)
Addresses	FttEnvControl	(ignored)
FttSystem	Enable	true
FttSystem	Control	true
FttEnvControl	Enable	false
FttEnvControl	Control	false

Table 5: Control GUI configuration for scenario A: remote control, no environment controller

Scenario B: Remote monitoring In this scenario (Table 6), the GUI is configured to display monitor data from the system and environmental controllers. They may be controlled by the ISS or a second instance of the control GUI.

Section	Key	Value
Addresses	FttSystem	(FTT rack-mount PC address)
Addresses	FttEnvControl	(Raspberry Pi address)
FttSystem	Enable	true
FttSystem	Control	false
FttEnvControl	Enable	true
FttEnvControl	Control	false

Table 6: Control GUI configuration for scenario B: remote monitoring

4.3 Environment

Several environment variables affect the behaviour of the control GUI. It should not normally be necessary to set any of these variables, but they are listed here for completeness.

`DISPLAY`	Specifies where to display the graphical user interface. Manually setting `DISPLAY` is rarely needed nowadays since it can be automatically and intelligently set by applications such as SSH.
`XDG_DATA_HOME`	Specifies the directory in which to store the user's private data for an application. This variable is typically unset since a sensible default is defined by the XDG specifications.
`XDG_CONFIG_HOME`	Specifies the directory in which to store the user's configuration information for an application. This variable is typically unset since a sensible default is defined by the XDG specifications.
`XDG_DATA_DIRS`	A colon-separated list of directories where applications search for data. This variable is typically unset since a sensible default is defined by the XDG specifications.
`XDG_CONFIG_DIRS`	A colon-separated list of directories where applications search for configuration information. This variable is typically unset since a sensible default is defined by the XDG specifications.

5 System software

There are two control interfaces to the system controller and the environment controller: the GSI and Cambridge's DLMsg interface (as used by the control GUI). Additionally, it is possible for several DLMsg clients to attempt a connection. Therefore, commands can potentially arrive from more than one source and some form of arbitration is required. This is further complicated by the requirement that Cambridge's DLMsg interface be easily removed from the code as the GSI acquires more functionality.

For each controller, the issue is resolved by making three variants available. The user simply runs the one which uses the most convenient control environment at the time. The variants are identified by their name suffix:

- `FTTCamSystem` and `FTTEnvSystem`: "Pure" controller variants, containing no DLMsg code. They can only be controlled by the GSI and all monitor data goes to the GSI. These currently do not get built, as the full GSI is not yet available.
- `FTTCamSystem_ucamcontrol` and `FTTEnvSystem_ucamcontrol`: These variants only allow control using Cambridge's DLMsg interface (and hence the included control GUI). Monitor data is sent via both DLMsg and the GSI (if available). Multiple control GUIs may connect and monitor, but only the first one to do so is able to control each system. These variants are appropriate for standalone tasks like telescope commissioning, as they do not require the GSI or a network connection. Furthermore, these variants should be used if the environmental controller is to communicate with the system controller (the DLMsg protocol is needed, because the GSI publish-subscribe system is not yet available).
- `FTTCamSystem_ucamdisplay` and `FTTEnvSystem_ucamdisplay`: These variants can only be controlled via the GSI interface. However, any number of control GUIs can connect at any time to monitor. These variants are useful when a diagnostic view of the status of each system is needed without interfering with GSI control.

In a standalone environment, a typical workflow pattern is to start the system controller and environmental controller, then start the control GUI of choice (either the FLC GUI, for unit telescope commissioning, or the FTT/NAS GUI, for full fast tip-tilt functionality).

5.1 Starting the systems

To run the standalone-mode FTTCamSystem and FTTEnvSystem variants which can be controlled by the FLC GUI, run the following commands (in separate terminals):

```
$ start_FTTCamSystem_ucamcontrol /path/to/property.txt
```

```
$ start_FTTEnvSystem_ucamcontrol /path/to/property.txt
```

where `property.txt` is a configuration file for each controller. Pathnames of ASCII files containing the system property values must be specified on the command line as shown above. Example `property.txt` files can be found in the source tree at `[ftt-root]/systems/FTTCamSystem/test` and `[ftt-root]/systems/FTTEnvSystem/test`. These are not installed by `make install`.

5.2 System Controller

The system controller normally runs on the FTT computer, rack-mounted in unit telescope enclosure cabinet Q5 [RD6].

The system controller acquires camera data and uses camera images to calculate corrections to send to the fast tip-tilt mirror. A full list of commands and monitor data for the system controller can be found in the interface control documents [RD3, RD4].

The system controller has three principal modes:

Idle This is the default mode after startup and initialisation. Startup checks have been done, and communications with the camera and FTTA interface have been established (if they are enabled). The software is waiting for commands and is sending monitor data about the current status at a rate of 10Hz.

Acquisition This mode is entered from idle mode by issuing one of the acquisition commands. In this mode, the camera reads out full-frame 512×512 pixel images. Three sets of monitor data are returned:

- The standard monitor data mentioned above, at 10Hz.
- Additional monitor data sampled at the camera frame rate (a maximum of about 30Hz) containing image metrics, such as a photon count, calculated image centroid, and seeing measurement.
- Telescope offloads, sent at 1Hz or at the camera frame rate, whichever is lower. Note that no method currently exists for transmitting the telescope offloads to the telescope itself.

The system can be placed back in idle mode by issuing a “StopRun” command. Some acquisition commands, such as “GetDarkFrame”, will automatically place the system in idle mode after the requested acquisition sequence is complete. The command “acquireAndStartTipTiltRun” will automatically place the system in fast tip-tilt mode (below) when a target is successfully acquired.

Fast tip-tilt This mode is not available with the FLC GUI. When the FTT/NAS GUI or the GSI is used, this mode is entered from idle mode by issuing one of the fast tip-tilt commands, or by issuing the “acquireAndStartTipTiltRun” command and waiting for the acquisition phase to complete successfully. In this mode, a CCD subframe (currently 32×32 pixels) is read out rapidly, with a frame rate of up to 1kHz. The images are optionally used to send a correction signal to the FTTA. This mode returns the same data as the acquisition mode, above, with the addition of a temporal seeing measurement, and reporting of the FTTA signals, at the camera frame rate.

The system can be placed back in idle mode by issuing a "StopRun" command.

The system controller is able to publish only a fraction of the camera images, to save network bandwidth. This fraction is controlled by separate "decimation" parameters for acquisition and fast tip-tilt mode. Only one image in every "decimation" is published, accompanied by image metrics for all of the frames including the omitted ones.

Additionally, the system controller can be shut down by the environmental controller if such functionality is enabled.

5.3 Environment Controller

The environment controller normally runs on a Raspberry Pi single board computer, embedded in the FTT computer's interface module, which is rack mounted in unit telescope enclosure cabinet Q5 [RD6]. The environment controller is not needed for telescope commissioning.

Its purpose is to monitor environmental conditions internal and external to the FTT camera enclosure, regulate the internal temperature, and to send alerts or shut down the system controller if the environment should become hostile to the FTT camera's continued operation.

A full list of commands and monitor data for the environment controller can be found in the interface control documents [RD3, RD4].

During normal operation, the environment controller only has one state, where it is monitoring the environment and sending monitor data at 1Hz. However, there are two controls available:

- Heater control can be set to automatic or manual. The default is automatic.

  When heater control is automatic, the environment controller tries to keep the temperature in the camera enclosure within bounds specified in the configuration. When it is manual, the user can turn it on or off directly.

- Camera control, if enabled, can be set to automatic or manual. The default is automatic.

  When camera control is automatic, the environmental controller can initiate a shutdown of the system controller when environmental conditions (as specified in the configuration) become hazardous to FTT camera operation. When it is manual, the user can initiate such a shutdown.

The FTT camera is rated to survive a much greater range of environmental conditions if no power is applied, and the only way to remove power completely is to shut down the FTT computer. Hence a complex shutdown procedure is required. It is as follows:

- When camera control is automatic and the environmental conditions go outside the specified bounds, the environmental controller goes into a "shutdown pending" state, where it continues to function as normal, but an alert is sent and a countdown timer starts. During the countdown, the shutdown will be aborted if the environmental conditions go back within bounds, or the user switches to manual camera control mode.

- If the countdown reaches zero, or if shutdown has been manually initiated, the environmental controller enters a "shutdown" state. At this point the environmental controller issues a command to the system controller to shut down.

- If power control is enabled, the environmental controller then enters a "power down pending" state, where a new countdown timer is initialised. This countdown cannot be aborted. The environmental controller then issues operating system commands to the FTT computer to commence powerdown.

- When the countdown reaches zero, the environmental controller issues commands to the power distribution board in cabinet Q5 to cut power to the FTT computer and to the camera itself (if this functionality is enabled - see Section 4.1.1). The environmental controller continues to function so that users can monitor the FTT camera environment.

6 Control GUI

The control GUI is a software application that provides an operator interface to the system controller and environment controller systems. Images and scalar monitor data published by the systems are displayed to the operator, and these displays update automatically as new data arrive. The application can record the images and scalar data to a set of FITS files when requested to do so by the operator. The resulting files can be read by `plotfttgui`. If the system and environment controllers are not being directed by an ISS supervisor, the control GUI applicaion is also used to control the FTT system.

There are two editions of the GUI:

FTT/NAS GUI Provides access to the full functionality of the FTT/NAS system, including the fast tip-tilt correction mode.

FLC GUI Provides access to the FLC functionality as defined in the ICD [RD3]. Compared with the FTT/NAS GUI, has additional data handling capabilities requested by AMOS for the purpose of UTM commissioning:

- Ability to record scalar data (e.g. unaveraged centroid estimates) to CSV files as well as FITS files.
- Scrolling display of recent average and rms centroid estimates.

The two GUI editions are almost identical in look and feel and share much of their code. They may be used interchangably with the system and environment controllers. The intention is that the FLC GUI be used with the FLC system for integration and commissioning of the first UTM. We expect that subsequent UTMs will be commissioned using FTT/NAS system hardware and the FLC GUI software, with the FTT/NAS GUI used to test the fast tip-tilt loop and for subsequent interferometer operations.

6.1 Starting the GUI

The same procedure is used for both editions of the GUI, although the configuration files are read from different locations. In the following instructions, `[gui]` stands for the name of the application, either `flcgui` or `fttgui`.

The GUI requires a configuration file and several widget definition files, which are installed in `/usr/local/share/[gui]` (further widget definition files are installed elsewhere, see Section 3). The application complies with the freedesktop.org XDG Base Directory Specification, and will search for these files firstly in `$HOME/.local/share/[gui]` and then `/usr/local/share/[gui]`. Thus a convenient method for modifying the configuration is to place a customised copy in `$HOME/.local/share/[gui]`. Alternative search locations may be specified using environment variables, see Section 4.3, but this is not recommended. Alternatively, the location of a configuration file may be specified with the `-i` command line option. Some hints for modifying the configuration are given in Section 4.

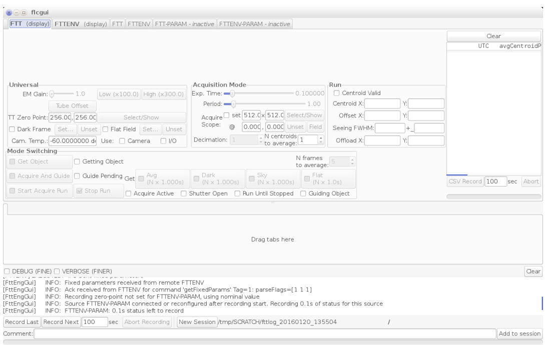
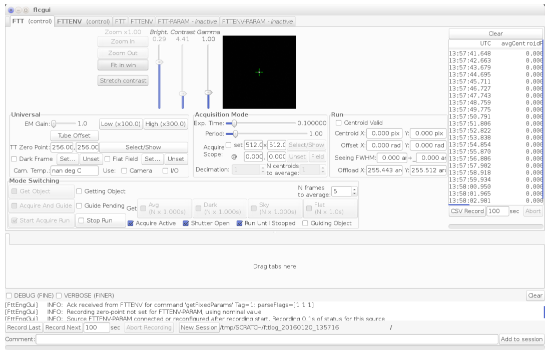

Figure 2: Screenshots of the FLC control GUI. Two views of the application main window are shown: on the left the GUI has been configured for display-only mode (note that most of the buttons are greyed-out) and the system controller is in idle mode; on the right the GUI is in standalone mode (buttons active) and the system controller is in acquisition mode.

Once the software has been installed and the configuration has been set appropriately, starting the GUI is simply a matter of:

```
$ [gui]
```

Or to specify an alternative configuration file:

```
$ [gui] -i /path/to/file.ini
```

6.2 GUI overview

The released FLC GUI is shown in Fig. 2. The FTT/NAS GUI is very similar in appearance. The main window includes the following regions (differences between the FLC and FTT/NAS GUI are *highlighted* in this description):

Upper notebook User-selectable tabs for system controller ("FTT (control)" or "FTT (display)"), environment controller ("FTTENV (control)" or "FTTENV (display)"), and generic text displays of scalar status items (remaining tabs). Within the main system controller tab, related control and display widgets are grouped together as follows:

Image display Greyscale display of latest (decimated) acquisition or fast tip-tilt mode image, with zoom and pan controls and adjustable mapping from pixel values to displayed grey level. Flat-field and dark frame images are displayed in a separate tab after they are acquired.

Text display of recent centroid estimates *(FLC GUI only)* This is located at the far right of the tab. The scrolling display shows the average and rms of the image centroid coordinates (in pixels) and the image FWHM (in arcsec). The number of frames averaged is set by "N centroids to average" in the "Acquisition Mode" grouping (see below). Underneath the scrolling display is a button for saving unaveraged data to a comma-separated ASCII file (the format is described in Appendix A).

Universal Widgets for displaying/setting universal parameters e.g. electron-multiplying gain and flat-field and dark frame corrections.

Acquisition Mode Widgets for displaying/setting acquisition-mode-specific parameters e.g. exposure time, frame period, decimation factor.

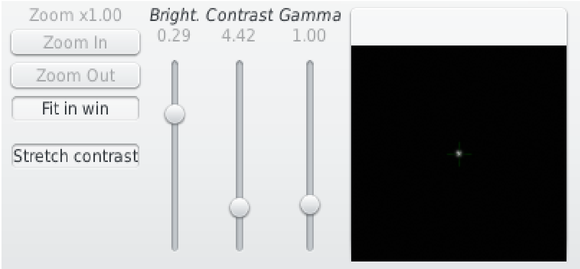

Figure 3: Image display widget.

FTT Mode ***(FTT/NAS GUI only)*** Widgets for displaying/setting fast-tip-tilt-mode-specific parameters e.g. frame period, servo parameters, decimation factor.

Mode Switching Widgets for displaying/setting the current operational mode.

Lower notebook Any of the tabs from the upper notebook may be dragged here so that two displays may be viewed simultaneously

Log message display Scrolling display of human-readable log messages. Checkbuttons toggle whether verbose messages are displayed.

Recording controls Controls to start and stop recording of data in FITS format. See below for more details.

Exception and fault messages are shown in a separate pop-up dialog box.

The widgets on the main environment controller tab are not shown in the figure but include displays of the temperature and humidity sensor readings and widgets for:

- Setting the heater power control to manual or automatic, and the power level if manual.
- Setting camera enable to manual or automatic, and disabling the camera if manual.

The buttons and other widgets on the GUI have tooltips giving brief help text. Further details on the commands executed by the buttons can be found in the ICD [RD3].

6.3 Image display

The greyscale image display widget used in flcgui and fttgui is shown in Fig. 3. The brightness, contrast and gamma settings define the mapping between the original image grey level x and the grey level x' displayed on screen. The mapping used is:

$$x' = \begin{cases} 0 & \text{if } x \leq x_{\text{min}}, \\ (c(x - x_{\text{min}}))^{\gamma} & \text{if } x > x_{\text{min}}, \end{cases} \tag{1}$$

$$\text{where} \quad x_{\text{min}} = \frac{1}{2} - \frac{1}{2c} - b, \tag{2}$$

with b the brightness and c the contrast. The “Stretch contrast” option sets b and c so that the full range of display grey levels are used. Note that changes to the greyscale mapping only take effect when the next image in the sequence is displayed.

By default, the image is sized to fit the available display area. If “Fit in win” is disabled, the “Zoom In” and “Zoom Out” buttons may be used to enlarge or reduce the image and scrollbars are shown for panning the image.

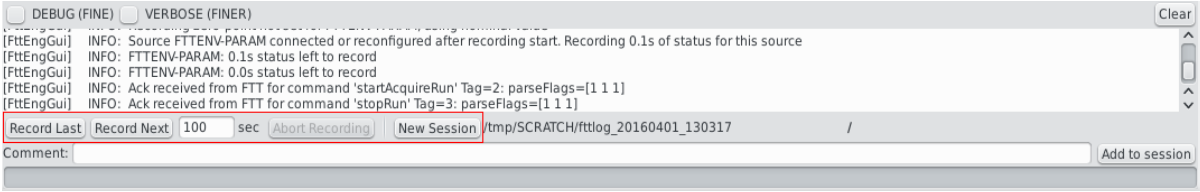

Figure 4: Controls for recording monitor data in flcgui and fttgui (in red rectangle).

6.4 Data recording

Monitor data are transmitted continuously to the GUI application using the dlmsg protocols. These data are inserted into a circular buffer on arrival, so that the most recent 100 seconds of data (the amount can be changed at compile time) are available most of the time. Log and fault messages are always written continuously to FITS files, but recording of other kinds of data must be initiated by the user. All kinds of data are grouped into **sessions**, which are intended to group a related set of recordings (such as those from a sequence of tests) made during the same 24 hour period. A session consists of a directory containing multiple FITS-format files (this is for efficiency when writing the data). Within a session, data are grouped into **recordings**, each of which is initiated by the user and spans a continuous time period. Starting the GUI application automatically initializes a new session. The user may close the session and begin a new one by clicking a button on the GUI.

The FITS files use the Heirarchical Grouping Convention (HGC) to describe the relationship between the various FITS files in the session directory. In particular the recording application writes an index file which is subsequently read by the analysis GUI in order to discover the recordings belonging to the session and the files that comprise each recording. The structure of the FITS files is described in Young [RD7].

The following recording functions are available (Fig. 4):

Record Last Record the most recent N seconds of data to the current session.

Record Next Record the next N seconds of data to the current session.

Abort Recording Stop the current recording operation.

New Session Close the current session and start a new one.

Add (comment) to session : Append a user-specified comment string to the header of the current session group table (these comments are displayed by the analysis GUI).

7 Analysis GUI

The tool for plotting time series and power spectra from FITS log files recorded by the control gui (either edition) is called `plotfttgui`. To run it, simply type:

```
$ plotfttgui
```

8 Errors

8.1 Exceptions

If the system controller or environment controller sets an exception, a message is sent to the control GUI. The message is then displayed to the user in a dialog box. If the exception is the result of a command, a second dialog box is also shown.

There are two kinds of exception: a "client" exception is returned if the user sends a command that cannot be executed in the current system state, or that contains an invalid parameter value; whereas an "internal" exception is returned if an apparently valid command fails to execute for any reason. The control GUI is designed to prevent "client" exceptions by automatically disabling controls and limiting parameter values to acceptable ranges. The system state should be unchanged following a client exception, hence it should be safe to continue to use the system. On the other hand, internal exceptions indicate unintended behaviour so please be cautious in subsequent use of the system.

8.2 Faults

The controller applications continuously check that certain parameters, such as the CCD head temperature and the camera enclosure environment, are within acceptable ranges. If a parameter goes out of range, a fault message is sent to the control GUI. The message is then displayed to the user in a dialog box.

8.3 Reporting bugs

Bug reporting is an informal process because MROI is the only user of the software.

When MROI discovers a suspected bug, Cambridge would prefer to firstly be notified via e-mail to John Young (`jsy1001@cam.ac.uk`) who will forward to Bodie Seneta (`bodie@mrao.cam.ac.uk`) if necessary. This ensures they are both kept aware of developments.

It is difficult to be prescriptive as to e-mail content, but please try to include enough information for us to understand and to duplicate the problem. The following is a suggested format modified from Debian's bug reporting guidelines (`https://www.debian.org/Bugs/Reporting`):

1. In the subject heading, a summary of the problem.
2. A description of what you did to make the problem appear.
3. A more detailed description of the problem itself: what you expected to happen, and what actually did happen.
4. If real hardware instead of emulation was used, a description of which hardware was connected.
5. Any speculation as to the cause of the problem, or suggestions for a fix.
6. The Linux kernel version used (from typing `uname -a`).
7. Attachments:
   - The configuration files the software programs were using when the bug occurred.
   - Log files generated by the software programs. Please try to use an example with log files sufficiently small to send as e-mail attachments. Cambridge's control/display GUI saves logs

and faults even when the user has not initiated a data recording, to a file named `log.fits` in the current session directory. The current session is shown in the GUI next to the "New Session" button (by default, this will be a subdirectory of `/tmp/SCRATCH`). The system and environment controllers currently place logs in the `MROI` subdirectory of the directory that the executable was run from.

- If real hardware is used, the output from typing `dmesg`. This might help with any driver problems.

When Cambridge receives such an e-mail, this initiates a dialogue, where we will work with you to try and resolve the problem.

A CSV file format

The "CSV record" feature in flcgui writes tabular data to an ASCII text file, with comma characters as field separators. The filename contains the date and time at the start of recording, for example `FTT-RUN_20160107_165920.csv`. Note that the live display in flcgui shows average and rms centroid coordinates, whereas the CSV file contains unaveraged centroids.

The file contains four columns, as follows. Column headings are given on the first line:

UTC Universal Time in the form HH:MM:SS.SSS .

CentroidPixX Image centroid X coordinate w.r.t. tip-tilt zero point, in pixels.

CentroidPixY Image centroid Y coordinate w.r.t. tip-tilt zero point, in pixels.

InstantSeeingFWHM Image Full Width at Half Maximum in arcsec.