Oxford Astrophysics

SWIFT OPERATIONS MANUAL

Document Number	SWF-MAN-SCI-01
Author	Fraser Clarke
Revision:	2.2
Date	11/10/2016
Status	Draft

CHANGE HISTORY

Version	Date	Author	Change
1.0	2008-12-08	FJC	Original version in LaTeX for PALAO operation
1.2	2009-04-29	FJC	Changes relevant for April/May 2009 run
1.3	2010-03-22	FJC	Extended and updated to capture lessons for past semester. Relevant for 10A observations. Draft release for March 2010 run
1.4	2011-04-08	FJC	Added in requirement for twilight flats for calibration of field illumination.
2.0	2013-01-19	FJC	Updated for P3K and non-AO operation. Moved into Word file.
2.1	2014-05-16	FJC	Further updates and reformatting (moved into new template). Included more information on software start-up and shutdown. Included requirement to leave instrument ‘parked’ in 235mas scale. Included DCS information within user manual (as opposed to separate document).
2.2	2016-10-11	FJC	Small updates during October 2016 run (final one perhaps!).

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AO	Adaptive Optics
CCD	Charge Coupled Device
DCS	Detector Control System
DM	Deformable Mirror
FM3	Fold Mirror '3' (actually the first mirror inside SWIFT)
GUI	Graphical User Interface
HODM	High Order Deformable Mirror
ICS	Instrument Control System
IFU	Integral Field Unit
IFS	Integral Field Spectrograph
LODM	Low Order Deformable Mirror
P3K	See PALM3K
PA	Position Angle (angle of minor axis of field North through East)
PALM3K	Palomar Adaptive Optics System (post-2011)
PALAO	Palomar Adaptive Optics System (pre-2011)
RTD	Real Time Display
SDSU	San Diego State University
Spaxel	SPAtial PIXel, referring to a single spatial pixel in the reconstructed data cube, to differentiate it from a single pixel on the CCD detector
SSM	Star Select Mirror
TBD	To be decided
TCS	Telescope Control System
TO	Telescope Operator
TTM	Tip-Tilt Mirror

1 APPLICABLE AND REFERENCE DOCUMENTS

1.1 Applicable Documents

The following documents at their indicated revision form part of this document to the extent specified herein.

AD1. SWIFT Guider Manual (v2.3; 2016-10-11)

1.2 Reference Documents

The following documents provide useful reference information associated with this document. These documents are to be used for information only. Changes to the date and/or revision number do not make this document out of date

RD1. SWIFT ArcVIEW command reference (v2.0; 2016-10-12)
RD2. SWIFT Calibration Settings
RD3. SWIFT Useful Commands
RD4. Caltech Instrumentation Note #617
RD5. Lakeshore 218 manual (P/N 119-007 v2.1 2009-11-18)
RD6. Lakeshore 325 manual (P/N 119-041 v1.1 2007-02-23)
RD7. Newport ESP300 Motion Controller/Driver manual (P/N 28187-02 Rev. F)

2 INSTRUMENT OVERVIEW

SWIFT is an integral field spectrograph covering the spectral range 6300-10100Å in a single shot at a spectral resolution of R~4000. The instrument uses an image slicer integral field unit, with a size of 44 by 91 spatial pixels ("spaxels"). There are three interchangeable pre-optics scales, which give 0.235, 0.08 and 0.016 arcseconds per spaxel. This corresponds to fields of view of 21x10.5, 7x3.5 and 1.2x0.6 arcseconds. The instrument is mounted to, and fed by, the PALM3K adaptive optics system.

Figure 1 shows the block diagram of the system, including the major control/data links in SWIFT, and the major optical components in PALM3K. The whole optomechanical system is mounted to the Cassegrain rotator of the 200-inch telescope, and can be fed either by light from the telescope, or from a stimulus source mounted on-top of the PALM3K bench. The SWIFT guide camera is also mounted on-top of the PALM3K bench, and picks off a ~1-arcminute diameter guide field ~4-arcminutes off-axis from the science field. Light passes through the PALM3K optical system, with wavefront sensing light (<650nm or <750nm, depending on the choice of dichroic) picked off just before the exit of PALM3K. Light redward of the dichroic is passed to SWIFT.

SWIFT has its own internal calibration sources (A Halogen lamp, Neon and Argon arc lamps), which can be selected to feed the instrument instead of light from PALM3K. The focus is inside SWIFT, and a set of selectable calibration masks are located on a wheel at the focus. The scale changing optics are located after the focus, and before the shutter (normally closed) and filter (blocking light <630nm). The pre-optics form a focus on the image slicer, which converts the rectangular field of view into to linear pseudo-slits. These slits feed identical twin spectrographs which form a spectrum onto the detectors (known as 'master' and 'slave'). The detectors are in liquid nitrogen cryostats at ~155-160K. The rest of the instrument is at ambient temperature.

The detectors are controlled by a pair of SDSU 'Leech' controllers, which in turn talk to the SWIFTLCU computer (located in the computer room) via fibre link. The swift guider is controlled by the SWIFTIC machine, which is mounted on the main body of the instrument. The remainder of the instrument electronics are connected to the electronics rack in the Cassegrain cage, and controlled from there via Ethernet connection. SWIFTWS located in the data room is the main user interface to the instrument control software.

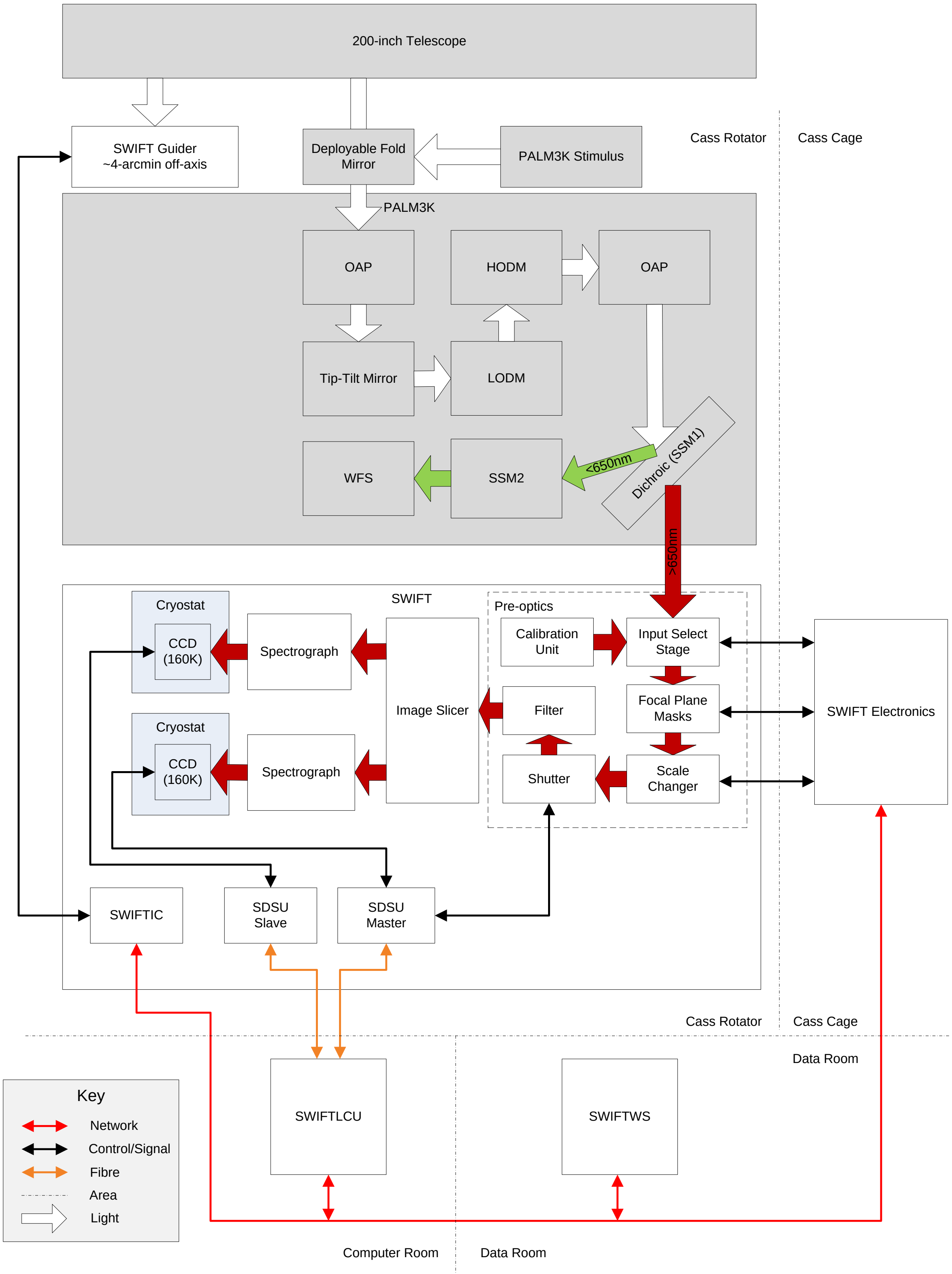

Figure 1: System block diagram of SWIFT showing main components and control/data lines.

2.1 Operational Modes

SWIFT itself has very few configurable options (i.e. pixel scale), and these do not affect how the instrument is used. Depending on the targets of interest however, the AO system can be used in two distinct modes.

2.1.1 Full-AO mode

The adaptive optics system is used in its normal full configuration, and SWIFT is typically used in the 80mas and 16mas spaxel scales. Acquisition and guiding is provided entirely by the AO system – the differential flexure measured between AO and SWIFT is very small (few milli-arcseconds per hour).

2.1.2 non-AO mode

For many science programmes with SWIFT, there is not a bright enough guide star to allow NGS correction with P3K. We have therefore implemented a “non-AO” mode for SWIFT, which makes for easier observing when AO correction won’t be used during a run. The main changes from the AO mode are that the **HODM is turned off**, and the **cooling draglines are not connected** to the telescope. SWIFT is used in the 235mas scale only.

The AO system is still used to provide a “flatmap” with the LODM, which corrects static aberrations in the telescope mirror. It is updated every few hours by pointing to a bright star near the science targets. A guide camera is mounted on top of the AO bench (before the AO optics), which provides guiding for non-AO operations.

3 DAILY CHECKLISTS

This section provides a quick check list of daily startup/shutdown tasks. It is only intended as a reminder for experienced observers. Details on each step are provided later in the manual.

3.1 Afternoon startup

- Check the cryostats have been filled (on white board in data room)
- Check disc space on SWIFTWS and SWIFTLCU
- Check/restart ArcVIEW on SWIFTLCU
- Check/restart SwiftUserGUI on SWIFTLCU
- Check/restart SwiftNightLog tool [optional]
- Check instrument status
  - Check temperatures (in particular that CCDs are at 160K)
  - Check cryostat pressures (<10^{-4} bar)
  - Check power status (SWIFTTS, SWIFTIC & SDSU on, NEARC and ARARC off)
  - Check motors are on in the ESP300 GUI
- Start/initialise CCD control GUI (after midday)
- Take a quick frame to check everything is working (may need to do erase/epurge)
- Check/restart data displayer on SWIFTWS
- Take some standard calibrations (dome is dark from 3--4pm)
  - Darks, if you know your exposure times
  - Flats in all scales to be used
  - Arcs in all scales to be used
  - Vertical lines in all scales to be used
  - Horizontal lines in all scales to be used
  - PSF images in 235mas scale
  - Flexure mapping sequences (need to move telescope)

3.2 Morning shutdown

- Take any calibrations you need (dome lights will be on from ~7:30am)
- Check power is off to all lamps
- **Leave Pre-Optics in 235mas scale (important!)**
- Power off CCD detectors (but not SDSUs)
- Close down SwiftNightLog and send night report (click Yes)
- Backup/transfer data from SWIFTWS

4 SWIFT START-UP AND SHUTDOWN

This section describes the steps required to start SWIFT at the beginning of a run, and shut it down at the end. It assumes the instrument has been installed by the day crew, and the computers turned on in the data room.

Some observers may want to follow these start-up/shutdown procedures each night, but this is generally not necessary.

4.1 Quick start reminders

Below is a quick start reminder for regular observers. More detailed instructions on each step are provided in the following sections.

4.1.1 On SWIFTWS

- Log into SWIFTWS
- Open VNC connection to SWIFTLCU (icon on desktop)
- Open Rdesktop connection to SWIFTIC (icon on desktop)
- Start `datadisplayer` tool (icon on desktop)
- Start `drpipeline.sh` tool

4.1.2 On SWIFTLCU

- Check/Start ArcVIEW
- Check/Start the DCS
- Check/Start SwiftUserGUI

4.1.3 On SWIFTIC

Only relevant in non-AO mode

- Start-up MaximDL
    - Connect to camera and start cooling
    - Connect to telescope

4.2 Software Startup

SWIFT uses three computers;

- SWIFTWS - Workstation in data room. Main user interface and data reduction machine.
- SWIFTLCU - Control computer in computer room. Controls instrument hardware and detectors
- SWIFTIC - Small windows machine mounted on SWIFT and used for non-AO guiding (SWIFTIC is sometimes also referred to as SWIFTITX for historical reasons…)

At the start of the run, log into the computers as follows:

1. Log into the SWIFT workstation (SWIFTWS) as user swift (ask support for password)

2. Start-up remote connections to SWIFTLCU and SWIFTIC via the icons on the desktop. Log into those with the same username/password.

The control software can then be started as follows

4.2.1 Instrument Control Software

The instrument control software (ICS) runs on SWIFTLCU. It is primarily run in LabVIEW, and specifically through an "ArcVIEW" core. It controls all the instrument functions other than the detectors, and interfaces with the telescope and AO system. It is possible to issue commands to the ICS via a GUI (SwiftUserGUI) or via the command line.

4.2.1.1 ArcVIEW

ArcVIEW is a set of LabVIEW programmes which coordinate communication between the various components of the ICS (e.g. temperature monitoring, motor control, FITS header generation, etc). swiftlcu is configured to start ArcVIEW on boot-up, but it is often necessary to restart ArcVIEW during an observing run. To start ArcVIEW, double click the icon on the SWIFTLCU desktop. Alternatively, in a terminal on SWIFTLCU run;

```
swiftlcu$>    start_ArcVIEW_Server &
```

This will bring up a LabVIEW window. ArcVIEW takes ~1 minute to initialize, and will show "Ready" in the top-left corner when done.

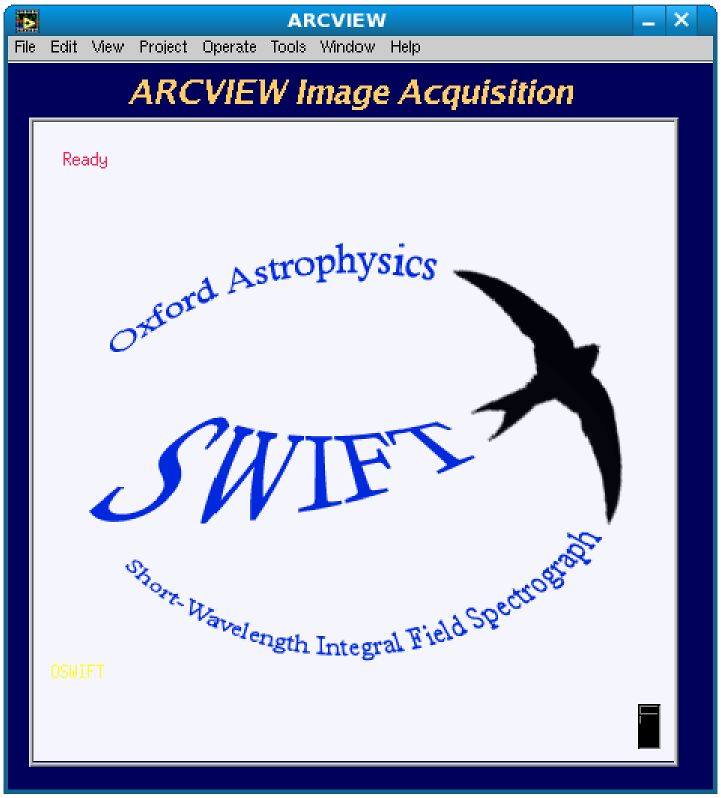

Figure 2: ArcVIEW window after start-up

4.2.1.2 SwiftUserGUI

The SwiftUserGUI is the main GUI for operations. It currently runs only on the SWIFTLCU. To start up the GUI, double click the icon on task bar at the top of the screen, or in a terminal run;

```
swiftlcu$>    SwiftUserGUI &
```

This will bring up a large window with several different sub panels (Figure 3), each of which should populate with a control panel. There are three tabs; "Instrument Control Panels", "Instrument Monitoring Panels", and "General Information". These are described in more detail later in the document.

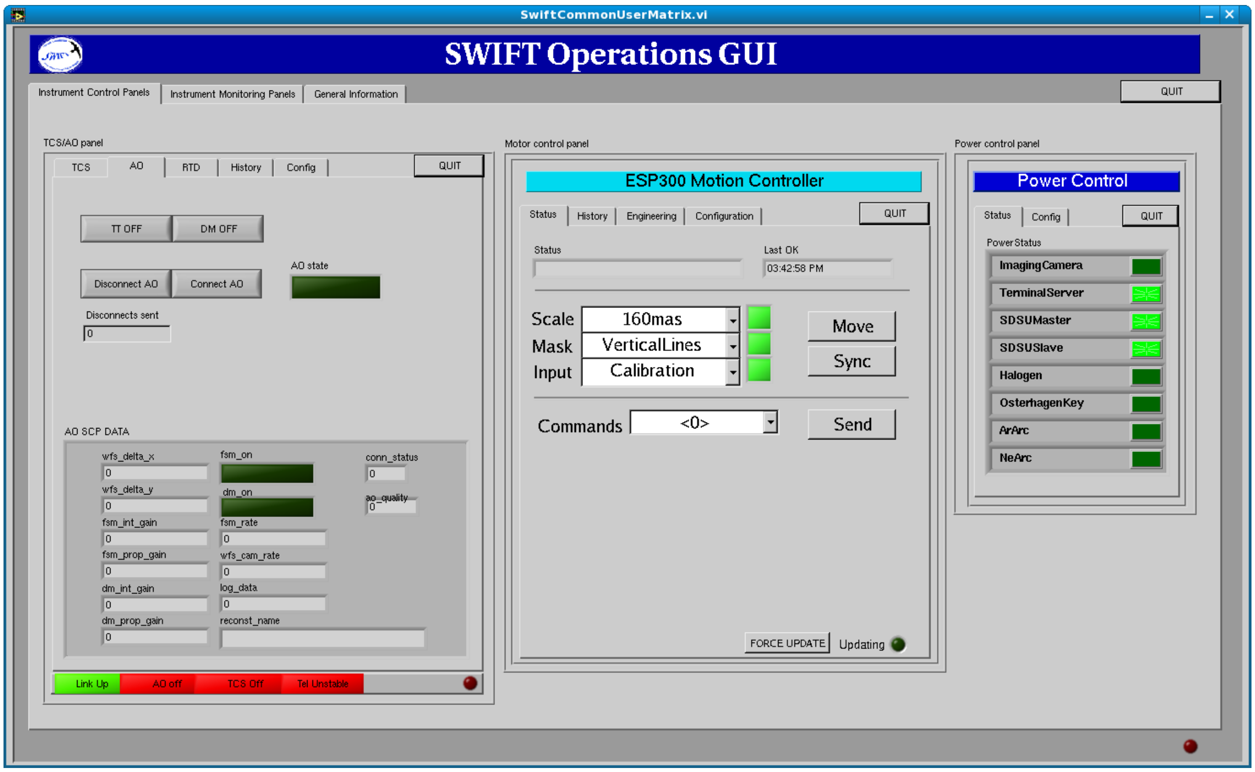

Figure 3: SWIFT User GUI at start-up -- Note that the TCS/AO panel is not currently used or displayed

It is worth noting that all the instrument functions can also be accessed from the command line

4.2.1.2.1 Power initialisation

In the Power Control panel on the SwiftUserGUI, check the status of all the power modes. The SDSUMaster and SDSUSlave powers are left off during installation, but can be powered up once the CCD detector cables are connected (check with the installation crew that this has been done). The default status at the start of the night should be;

Switch	Default Status
SDSUMaster	On
SDSUSlave	On
SwiftITX	On
ImagingCamera	N/A (not connected at present)
TerminalServer	On
OsterhagenKey	N/A (spare port not connected to anything)
NeArc	Off
ArArc	Off

Table 1: Default power status after software start-up

To turn on/off a power button, click on the 'light'. Critical systems (e.g. SDSU power) will pop-up a confirmation box before changing the power state.

4.2.1.2.2 Motor initialisation

IMPORTANT – there is a physical interference with the (new) 16mas scale in SWIFT. The control software avoids this, but homing the motors has the potential to drive the 16mas scale into the instrument cover, probably damaging it. Homing should only be done from the 235mas scale, and NEVER from the 80mas scale. Please 'park' the instrument in the 235mas scale to ensure it starts up in a safe position next time.

If the instrument has been left in the 80mas scale (e.g. due to a power cut) you MUST manually move the stage to a safe location before initialising the motor controller (homing the motors). Please call one of the instrument team (Instrument team contact details) if you need to do this!

The ESP300 motion controller uses a relative encoder, which must be 'homed' before it knows where it is (true after any power-on). After the SwiftUserGUI is started, on the ESP300 Motion Controller panel select "Motors On" from the "Commands" drop-down list and click "Send". The lights next to the scale/mask/input fields should 'turn on'. Now select 'Home Motors' from the "commands" drop-down list and click "Send". The lights will flash yellow/orange in turn as the motors move. When they stop, all the lights may be red (indicating the motor is not in the position indicated in the scale/mask/input drop-down lists). Click 'sync' and all the fields should read 0.0 and lights turn green.

Alternative, from the command line on SWIFTLCU or SWIFTWS, you can do:

`swiftlcu$>    esp300 POWER ON` [wait 1-2 seconds]
`swiftlcu$>    esp300 HOME ALL` [moves the motors in sequence to their home positions (0 degrees). Takes about 3 minutes to complete]

The motors should now be 'homed' in the 0.0 degree positions. You can now select some default configuration (e.g. 235mas/Open/Calibration) from the ESP300 Motion Controller panel and click 'Move'.

4.2.1.3 Detector temperature control

The detector temperatures are controller by a lakeshore 325 module. This module is configured to start controlling the temperatures at power-on, but sometimes fails to (possibly due to the order the cables are connected when installing the instrument?). You can manually turn the control loops on once the ArcVIEW software is running with the following commands on SWIFTLCU or SWIFTWS:

```
swiftlcu$>   ls325 PASS RANGE 1,1
swiftlcu$>   ls325 PASS RANGE 2,1
```

If the weather is particularly warm, the detectors may not reach their nominal operating temperature of 155K. In this case, it is better to increase the set point temperature to ensure they stay at a constant (but higher) temperature. 160K is usually a fairly safe number. To change the set point, from SWIFTLCU or SWIFTWS, do:

```
swiftlcu$>   ls325 PASS SETP 1,160       [to set to 160K]
swiftlcu$>   ls325 PASS SETP 1,160
```

You can watch the effect on the temperature graph on the LS325 panel on the SwiftUserGUI. The detectors should take 10-20 minutes to stabilise at the new set point.

4.2.2 Detector Control Software

The CCDs are controlled from a Python GUI running on SWIFTLCU. To start it double click on one of the “DCS” icons on the desktop. There is an icon for “DCS (Science)”, which starts the detectors up in slow (48kHz) low noise (3-5e-) readout mode for standard science observations, and “DCS (Engineering)” which starts the detectors in fast (190kHz) high noise (~30e-) readout mode. The latter is useful for testing and initial acquisition, as the readout time is considerably shorter. Only one mode/GUI can be used at a time.

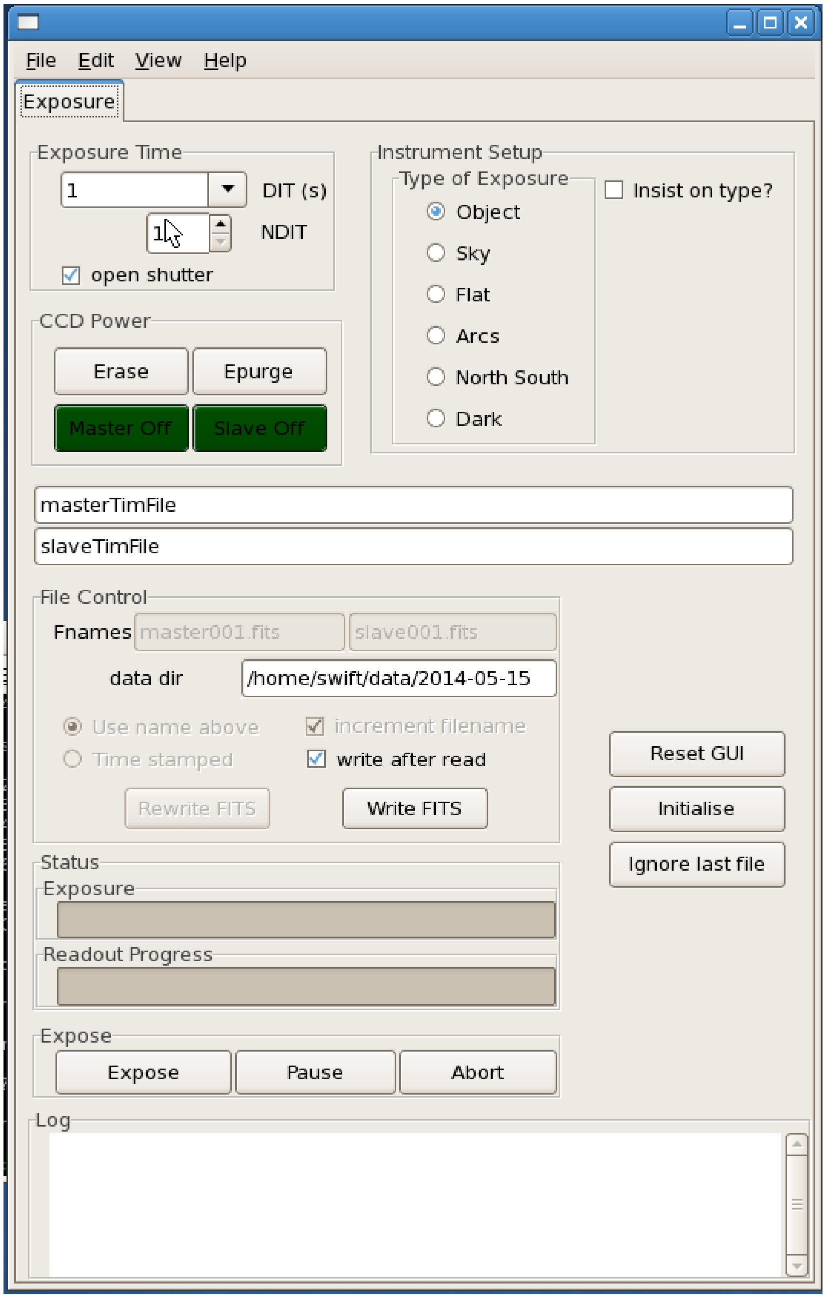

Figure 4: SWIFT Detector controller GUI.

The icon starts up an ipython shell and a GUI (Figure 4 – though note that this screen grab was taken without the controller hardware connected; in operation the masterTimFile/slaveTimFile show the timing files loaded on the controller (i.e. what read-speed is being used)). Check that it does not generate memory allocation errors in the ipython shell. If they do occur, try quitting/restarting the DCS.

Within the python command prompt, type;

```
ipython:>   master.testTDL() ; slave.testTDL()
```

This tests the connection to the two controllers, and should return '0' if everything is OK. If not, see the troubleshooting section at the end.

The detectors are now ready to use, but not yet powered on. They should not be powered on until the cass cage lights are off to avoid any risk of over-illumination. To power on the detectors click the power buttons on the GUI (unlike several of the functions on the 'GUI', the power buttons DO report the correct and current (within 1s) status of the detector power).

Now perform a couple of Erase/Epurge cycles with the GUI to clear residual electronic signal in the detectors.

4.2.3 General software start-up/checks

4.2.3.1 Check disc space

Make sure the /home disc on SWIFTLCU and SWIFTWS are not too full, there should be at least 10Gb free for a night's observing. Use the command `df -h` to check the disc space. If /home is too full (>80%, or <8Gb free), cd into the `/home/swift/data/` directory and remove the oldest folders. Be sure they have been backed up!

4.2.3.2 DataDisplayer

The data displayer script automatically displays images as they arrive on SWIFTWS. It also sounds an alert to let the observer know an exposure is finished. To set up the data displayer, from a terminal on SWIFTWS run;

```
swiftws$>    datadisplayer.sh master slave
```

This will bring up two ds9 windows (ds9isis and ds9cherwell) which will automatically display master and slave images respectively as they arrive. Images are displayed with long wavelengths to the top of the image.

4.2.3.3 Data reduction pipeline

SWIFTWS includes a version of the SWIFT pipeline which will process images as they arrive from SWIFTLCU (see §5.7 for more details). It takes ~2 minutes to generate a cube. To start the pipeline;

```
swiftws$>    drpipeline.sh
```

Once cubes have been reduced, they will appear in the nightly data directory on SWIFTWS as msXXX.sci.cube.fits, where XXX is the file number.

4.3 Software Shutdown

There are only a few steps which need to be taken to shut down the software at the end of the run/night. This is mainly ensuring that the hardware is in a safe state to power off.

4.3.1 Detector Control Software

Power off the CCDs via the GUI or the ipython shell;

```
ipython:>   master.powerOff(); master.slaveOff()
```

Quit the DCS GUI and exit the ipython shell/window.

It is safe to turn off the SDSU power supplies once the detectors are turned off.

4.3.2 Instrument Control Software

Move the pixel scale to 235mas ***(IMPORTANT! This ensures the motors will start up in a safe position at the next power-on)***

```
swiftlcu$>   scale 235mas
```

Make sure the lamps are off via the power panel in SwiftUserGUI, or via the command line;

```
swiftlcu$>   arcs off
swiftlcu$>   halogen off
```

The software can then be quit/closed.

4.3.3 General software

There are no specific shutdown requirements. Quit/close windows and terminals. Ctrl-C programs.

5 OBSERVING WITH SWIFT

Some hints and tips based on using SWIFT.

5.1 Useful commands and quick tips

This is list of common commands and quick-tips can be found on a sheet of paper on the right hand side of the SWIFTWS desk for easy reference.

Commands in `courier` can be issued from the command prompt on SWIFTLCU or SWIFTWS

Command	Description
tcs offset E.E N.N	Offset the telescope E and N in arcseconds (nonAO)
p3k sci_motion n/s/e/w X.X	Offset P3K (inc Telescope and SSMs) by X.X arseconds in given direction (AO). 5” max steps recommended.
halogen on/off	Turn on/off the Halogen lamp
halogen set 235mas/80mas/16mas	Set the Halogen lamp voltage for given plate scale
arcs on/off	Turn on/off Arc lamps
scale 235mas/80mas/16mas	Change the scale to 235/80/16mas
input Telescope/Calibration	Select the input to the instrument
gortd	Run the reconstructor on a given file number.

Table 2: Common commands used during observing

5.1.1 Quick tips

To **reconstruct a 2d image** from the data, run `gortd DARK# ../235mas.recon all FRAME#` from with the night’s data directory. Image is displayed in the RTDDS9 window.

To **measure an offset in the RTD**, draw a “ruler” (default shape) from the object to where you want to go. Double click to bring up Ruler properties. Do Length->WCS and Length->Arcsec. Read-off the East and North from the Axis Length fields, and apply with `tcs offset` or `p3k sci_motion`. Zoom->Align will display the image with North up and East left. *Remember that offsets are to the telescope not to the object – objects will appear to move in the opposite direction on the IFU.*

To **measure seeing from a reconstructed image,** start-up iraf (desktop icon) and type `imexam`. The cursor on rtdds9 will turn into a circle; move it over the star and hit ‘r’ (for a radial plot) or ‘a’ (for an aperture measure). FWHM is reported in pixels. Hit ‘q’ on the rtdds9 window to quit imexam. Note, iraf will speak to the most recent ds9 window opened, so to be safe quit and restart rtdds9 before doing this.

To **see the instrument status**, open up a web-browser and go to http://swiftlcu.palomar.caltech.edu/php/status.php . Check that UTC-TCS is being updated correctly, and if not ArcVIEW needs to be restarted.

To **check for saturation on images** in the data displayer windows, set Scale->Minmax and then View->Horizontal Graph. Mouse over to see any peaks above 65000. View->Vertical Graph can be used for arc

To **transfer images from the guider computer** (swiftic), connect using the icon on the swiftws desktop, which will provide a drive called “Data on swiftws”. Drag images/folders in here. They will be copied over to `/home/swift/data/GuiderImages/` on swiftws.

To **change the guider position angle after rotating** the cass ring, in MaximDL on swiftic go to the Camera Control window and Guide tab, click “Settings” and change “Angle” in the “Manual Calibration” area. The Angle should be approximately cass Ring – 180 degrees.

Small moves can be made via **guider offsets**; increasing guider X moves the science object up in Y on the RTD (across slices). Increasing guider Y moves science object down in X on the RTD (along slice). 1 guider pixel (8x8 binning) is equivalent to 0.22”.

5.2 Preparing the observations

5.2.1 Dichroic choice

SWIFT has two dichroics, which send "blue" light to the AO system for wavefront sensing. The general choice is the 650nm dichroic, which sends light blue of 650nm to the AO system, and allows SWIFT to use its full wavelength range. There is also a 750nm dichroic, which sends slightly more light to the AO system, but blocks the blue portion of SWIFT's coverage (Figure 5). You should chose the latter if you have no features of interest below 750nm and are using NGS, as it will maximize AO performance. This is particularly true if your guide stars are relatively cool (<8000K; Figure 6), where the additional red light on the wave front sensor can gain upwards of a magnitude in effective brightness.

The dichroics are mounted in the AO system, and although they can be changed during the night this is not recommended. In general, the telescope operators ***do not*** know how to change the AO dichroics, and this is done by the daytime support personnel. If possible, you should use one dichroic for your run, and let the instrument/AO team know which one beforehand.

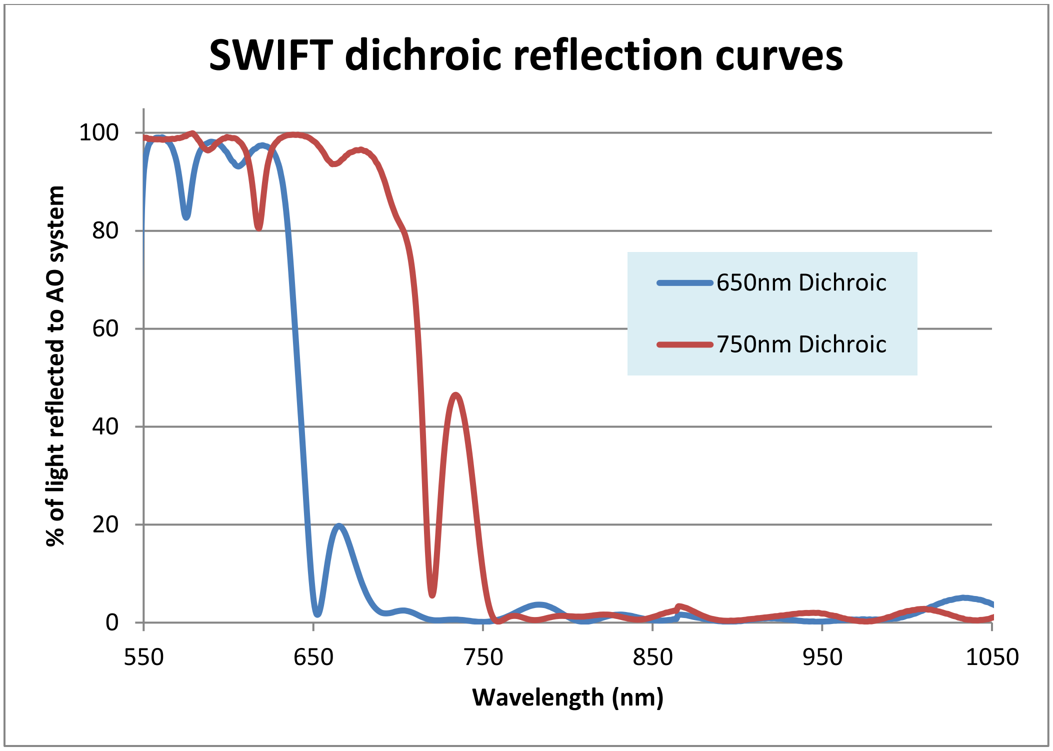

Figure 5: Reflection profiles of the two SWIFT dichroics.

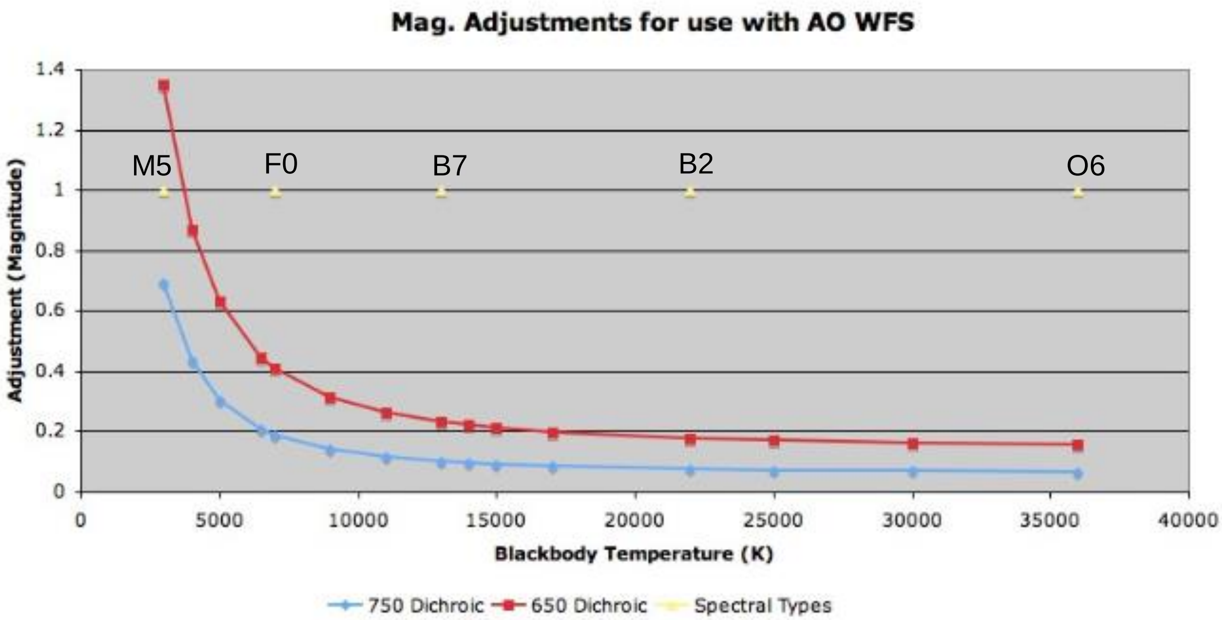

Figure 6: Effective loss of AO guide star magnitude as a function of star temperature

5.2.2 Target coordinate files

Target coordinate files can be prepared in the Palomar format for use with the FACSUM system, as described at:

http://www.astro.caltech.edu/palomar/software/facsum/index.html

5.2.3 Position Angle

The default position angle for SWIFT is North along the short axis of the IFU, and East along the long axis of the IFU. The default Cassegrain rotator angle for SWIFT is 100.9 degrees. This puts the master chip North and the slave chip South. The WCS information (i.e N-E arrows) on the RTD is reliable. Note that, by default, the RTD will display North down (i.e. master on the bottom). To fix this, choose "Zoom->Align" from the rtdds9 menu.

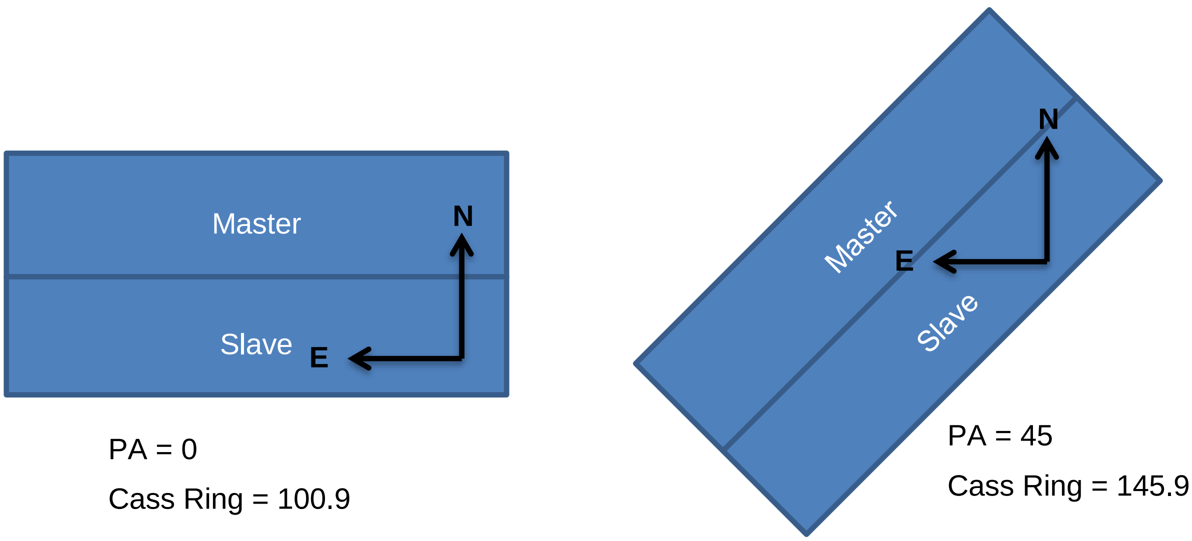

Figure 7: Cartoon of field geometry at different rotator angles.

Changing position angle during the night is possible, but requires you to park the telescope, go up to Cass and move the rotator locally from inside the Cassegrain cage. This is due to very complex/extensive cabling associated with P3K.

To set a given position angle (PA), north through east, **move the Cassegrain rotator to PA+100.9 degrees**.

The limits on Cassegrain rotator (cf PA) are 90 (-10.9) to 270 (169.1)

5.2.4 Exposure times

The thick detectors in SWIFT are more sensitive to cosmics than normal CCDs. Maximum practical exposure time is 1200 seconds. It is sensible to “cosmic split” your integrations, and take at least 3 exposures to reject the cosmics. The instrument still has problems with flexure, and you may want to limit exposures to 10/15 minutes to reduce this effect.

The minimum exposure time is ~0.1 seconds. Below this there are some effects from synchronising the shutter with the CCD readout.

5.3 Running the instrument

This section outlines the tools and techniques used during normal observations with SWIFT

5.3.1 Instrument Control

Figure 8 shows a top level architecture of the instrument control software, outlining how the major components interact with each other. The observer can control the instrument (and interact with the telescope and AO system) either via the command line or via the SwiftUserGUI, and can view the system status via a webpage.

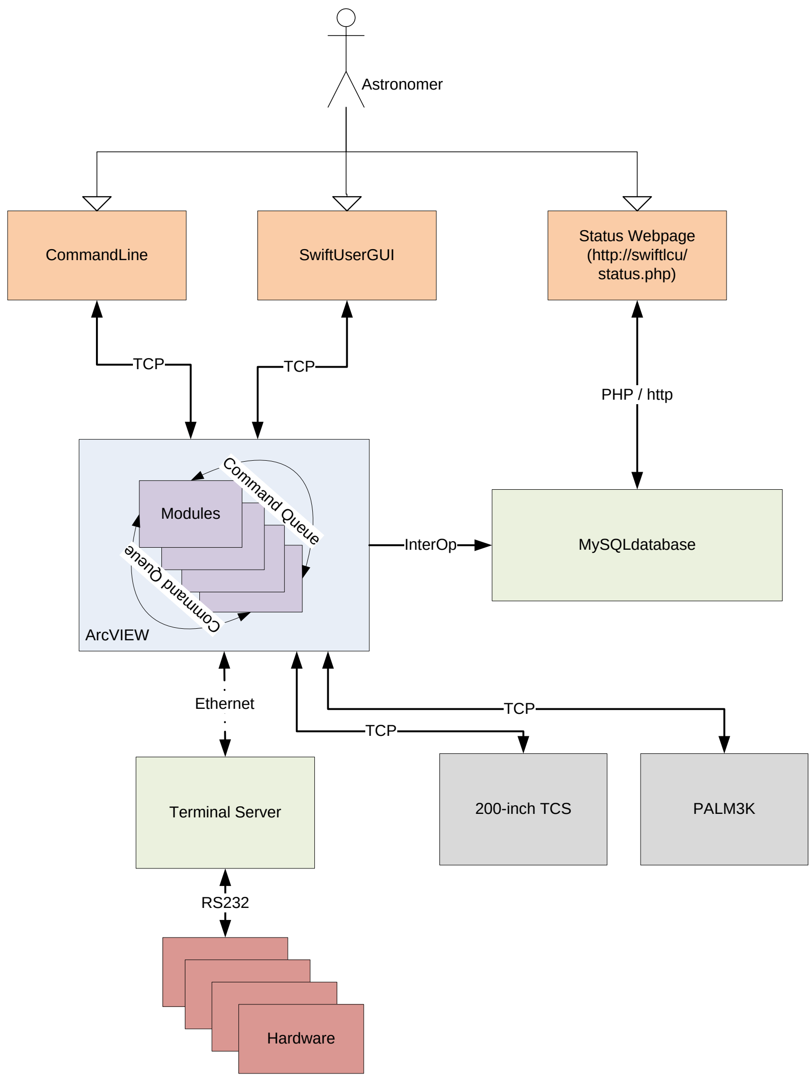

Figure 8 : ICS software architecture.

5.3.1.1 Status monitoring

The easiest way to monitor the status of the instrument is via a website on SWIFTLCU:

http://swiftlcu.palomar.caltech.edu/status.php [only visible within Palomar]

This page lists the current status of the major components of the instrument. Green values are in range/normal, Yellow are OK – but perhaps not what you want for observing (e.g. if the Calibration mirror is in), Red values are out of range or in error, and Orange shows that the data is suspiciously old, indicating that the ArcVIEW server may need to be restarted.

5.3.1.2 Command line control

It is possible to issue commands to ArcVIEW (and hence in turn the instrument hardware, PALM3K, and the TCS) via the command line on SWIFTWS and SWIFTLCU. For a detailed list of commands see The SWIFT ArcVIEW command reference [RD1]. The most useful are given in short in §5.1 above.

Note that at present, the command line interface is the only way to send commands to the TCS (via `tcs`) and PALM3K (via `p3k`)

5.3.1.3 GUI control

The SwiftUserGUI provides inputs and outputs needed to control and monitor the instrument functions. It is made up several individual panels arranged into three tabs; Instrument Control, Instrument Monitoring and General Information. Each is described in more detail in the sections below.

5.3.1.3.1 Instrument Control Panels

These panels are the main ones for controlling the instrument functions.

5.3.1.3.1.1 TCS/AO control

This panel has not been updated for P3K operation. Use instead the command line tools `p3k` and `tcs`.

5.3.1.3.1.2 Motor Control

This panel controls the position of the three moveable stages in SWIFT (image scale, focal plane mask \& input selection). Each motor has a drop down list of predefined positions. A bright green light next indicates the motor is on and in position. Dark green indicates the motor is off. Red indicates the motor is not at the position indicated by the panel. Orange/Yellow indicates the motor is moving.

To move to a new position, select the position from the list (the light will turn red) and click ``Move'' (the light should turn orange/yellow). When the motion has finished, the light will turn green again. You can move all the stages simultaneously.

``Sync'' button retrieves the actual motor position from the controller and updates the GUI. If the motor is in an undefined position, the angular position of the motor is returned.

After a power cycle, you will need to turn power on to the Motors. Use the ``Commands'' drop down menu. If the motors lose their reference, send the ``Home'' command. You should home the motors after a power cycle. ***IMPORTANT – there is a physical interference with the (new) 16mas scale in SWIFT. The control software avoids this, but homing the motors has the potential to drive the 16mas scale into the instrument cover, probably damaging it. Homing***

should only be done from the 235mas scale, and NEVER from the 80mas scale. Please 'park' the instrument in the 235mas scale to ensure it starts up in a safe position next time.

The ``Mask'' and ``Input'' stages take ~80s to make a full revolution. We have tried to put the common functions as close together on these stages. The ``Scale'' stage takes ~20s for a full revolution.

5.3.1.3.1.3 Power Control

This panel controls the network power supply, and can be used to turn power on and off to various functions in the instrument. To turn a function on or off, simply click on the green button to the right. Critical functions (i.e. SDSU power supplies) will ask for confirmation before cycling the power status.

If the information in the front panel disappears, try going to the config tab and changing the connection mode from "direct" to "ArcVIEW", and back again. This should reset the front panel.

5.3.1.3.2 Instrument Monitoring Panels

This tab shows the monitoring information from the instrument; temperatures and pressures. Shouldn't be anything to worry about. Check the temps occasionally to make sure the coldplates are at ~77K and the detectors are at 155K (160K during warm summer runs). If the coldplates warm up, the cryostats need to be filled.

5.3.1.3.3 General Information

Currently only shows the 200-inch weather/temperature information.

5.3.2 Detector Control

The detector control GUI should be self-explanatory. The only odd feature of the SWIFT detectors is the need to 'erase' and 'epurge' after powering on the detectors. This removes static electrical signal from the detector which would otherwise cause a lot of bad pixels at the edge of the devices. It is worth doing 2—3 erase/epurge cycles every so often (few hours), or if the detector has been saturated. After a power-on, it is worth taking a few sacrificial exposures to fully clear the chips before observing.

5.3.2.1 Changing read modes

The SWIFT detector can currently only be read out in single amp mode due to a failure of one of the amplifiers. A future fix is planned to bring back 2-amp readout. Unfortunately, 4-amp readout will never be possible with the current set of detectors.

5.3.2.2 Changing read speeds

To change read speed you need to restart the DCS (extensive efforts to change the read speed without doing this have met with limited success unfortunately!). Power off the detectors, quit the python GUI/shell, and restart the relevant DCS from the desktop. The two options are:

- DCS (Science)   48kHz readout speed. 3-5 e- readnoise taking ~180s
- DCS (Engineering)   190kHz readout speed. 20-30e- readnoise taking ~40s

Due to the overhead in changing (erase/epurge cycles), it is only worth going to the engineering mode if you are doing a lot of tests (i.e. focusing the system).

5.3.3 Real Time Display (RTD)

The Real Time Display is a simple tool to reconstruct a 2-d image from the raw data format. The goal of the tool is to provide a quick reconstruction to allow the observer to check the acquisition and place the target exactly where they want on the IFU.

5.3.3.1 Reconstructing an image

The reconstruction programme is called `rtdpipe`, and should be accessible from the command prompt on both SWIFTWS and SWIFTLCU. The programme has several command line options to control the reconstruction, and typing `rtdpipe` on its own will give a list of these. A far easier way to run the reconstruction however is through the `gortd` script, which simplifies the input to the programme, and can be run from the nightly data directory as follows:

```
swiftlcu$>   gortd [BACKGROUND#] ../recon.vectors
[red/blue/all/psf] IMAGE#
```

where;

- [IMAGE #] is the number of the file to reconstruct (i.e. master011.fits would be 011).
- `../recon.vectors` is the reconstruction vector file, typically one per scale in the `/home/swift/data` directory as `235mas.recon/80mas.recon/16mas.recon`.
- `red/blue/all/ha/psf` [optional] the wavelength range over which the reconstruction should be done
- DARK# [optional] a "dark" or "sky" file to subtract before reconstruction.

An example reconstruction command could be;

```
swiftlcu$>   gortd 33 ../235mas.recon psf 34
```

which would reconstruct a 2-d image of the `master034` and `slave034` frames, using the vectors `../235mas.recon` and the wavelength range "psf" (~8500-8600Å), and subtracting the frames `master033` and `slave033` from the object frames.

The image is displayed in a ds9 window called '`rtdds9`'. The `gortd` script will start `rtdds9` if it is not already running.

5.3.4 Automated Observation Sequencer

A prototype observation sequencer is available for the brave of heart. In the ipython window opened up by the DCS:

```
ipython:>   import sequencer
ipython:>   sequencer.attach_gui(gui)
ipython:>   sequencer.execute('MyFile.seq')
ipython:>   sequencer.abortSequence()   [if you need to abort the sequence]
```

Be warned! The current implementation is very naïve and has little/no error checking in it. It is not recommended to use the sequencer for on-sky observations, but it is useful for taking calibrations over dinner! Sequence files exist for the standard calibrations in all the scales (e.g. Cals235mas.seq).

5.3.4.1 Sequence file definition

Sequence files are simple text files with a list of commands.

Command	Description
NEXP N	Set number of exposures for all following exposures
EXPTIME XXX	Set exposure time for future exposures
EXPOSE	Take NEXP exposures of EXPTIME
DARK	Take NEXP darks of EXPTIME
ARC 16/80/235	Take NEXP arcs in the 16/80/235 scale (configures instrument and exptime automatically)
FLAT 16/80/235	Take NEXP flats in the 16/80/235 scale (configures instrument and exptime automatically)
HL 16/80/80F/235	Take NEXP horizontal line exposures in the 16/80/235 scale. For the 80mas scale, there is a choice of using the fine (80F) or coarse (80) mask. (configures instrument and exptime automatically)
VL 16/80/80F/235	Take NEXP vertical line exposures in the 16/80/235 scale. For the 80mas scale, there is a choice of using the fine (80F) or coarse (80) mask. (configures instrument and exptime automatically)
OFFSET P3K/TCS <ra> <dec>	Offset the TCS or P3K (sci_motion) by <ra> and <dec> arcseconds. ***Note that this does not offset the non-AO guider!***
SCALE 16/80/235	Set the instrument scale to the given value
SHUTDOWN	Shutdown the instrument. Power off lamps and detectors. Move scale to 235mas.

Table 3: Sequence file command list

5.4 NGS-AO observations

The default mode for SWIFT is NGS operation with P3K.

5.4.1 Guide star magnitudes

The minimum useable NGS guide star magnitude is a function of seeing, distance, dichroic choice and guide star spectral type. In median natural seeing of ~1 arcsecond, with the 650nm dichroic and the 64x64 lenslet array, the system typically can lock on a V~12 guide star within 30 arcseconds of the target and provide useful correction. If the star is very red, or the natural seeing poorer, the guide star limit is brighter. The smaller lenslets (32x32 or 8x8) provide less correction, but have substantially fainter limiting magnitudes. The 750nm dichroic also increases the limiting magnitude by ~1 magnitude.

The maximum distance from the NGS to the target is 45 arcseconds, but correction at this distance is likely to be minimal due to the effects of anisoplanatism at SWIFT's relative short observing wavelength.

5.4.2 Target acquisition

The `p3k sci_motion` command is used to change the relative pointing between the AO-WFS and the IFU. For acquiring off-axis targets, the following procedure is suggested;

- Set the SSM positions to the default value determined previously (from the white light during the day, or from a previous observation) via a restore file
- Slew the telescope to the guide star coordinates
- Acquire the guide star in the AO system
- (depending on your confidence) take an image through SWIFT to verify that the guide star appears on the IFU where it should. Adjust with `p3k sci_motion` if not.
- Apply the offset between the guide star and the target with the `p3k sci_motion` command. If the move is large (>10" say), it is sensible to split it into several smaller moves.
- Once stable, start exposure on SWIFT

Remember to reverse the `p3k sci_motion` at the end of the observation to restore the relative pointing of AO and SWIFT to the default value. You could also restore a previous configuration to achieve this.

5.4.3 Offsets

Offsets should be done with the `p3k sci_motion` command to maintain the alignment between the AO-WFS and SWIFT. The exception to this is if you are doing offset skys and do not need AO correction on the sky. In this case, open the AO loops and issue a `tcs offset` command to change the pointing.

5.4.4 Recovering pointing information from the AO

If the NGS stays locked, it will correct the telescope's tracking errors, and maintain the pointing on the instrument at the level of ~2—3 milli-arcseconds/hr. It does NOT however maintain the correct WCS information in the FITS header. If you need the WCS information to combine your data later, you must retrieve it from the AO system logs manually. Descriptions on how to do this are provided in Caltech Instrumentation Note #637 [RD4]. (To attach)

Not sure how relevant this is now (post PALM3K) – need to update!

5.5 noAO (seeing limited) observations

SWIFT is always mounted to the AO system, but it is possible to observe in a "no-AO" mode if there is not a suitable guide star available. A seeing limited guider is now available, which makes this mode significantly more usable.

5.5.1 Target acquisition

A good technique for quick target acquisition in seeing limited mode is to use a nearby (<1 degree) bright star (<10 typ) which can be seen in the AO acquisition camera. The star is centred on the AO-WFS (disappears in the hole in the acquisition camera), and the telescope 'tx'd before slewing to the science target.

The alignment between the AO acquisition camera and SWIFT can be controlled by using the `p3k sci_motion` command (which moves the SSMs), so the observer can choose any point on the IFU as their 'reference'. Once this alignment is set (e.g. on a telluric star) it is not necessary to take an image of acquisition stars through SWIFT.

Detailed steps for this procedure are:

- Set the reference location on a bright target, e.g. a telluric star;
    - Acquire the star with the AO system, and ideally lock at least the tip-tilt loops
    - Take an exposure with SWIFT, and then use `p3k sci_motion` to move the target to the desired position on the IFU whilst remaining in the AO-WFS.
    - The AO-WFS is now co-aligned with your chosen 'reference point' on the IFU, and that alignment is stable <100mas over the sky. It is worth checking each time you go to a bright star though.
- To acquire a faint target to the reference position on the IFU
    - Send the target coordinates to the telescope
    - Ask the TO to select a bright (<10) star near the target (<1 degree) and slew there
    - Look for the star on the AO acquisition camera, and use the AO paddle to put the bright star onto the wavefront sensor (or at least make it disappear into the hole on the AO acquisition camera).
    - ***TX the telescope to zero the offsets***
    - Slew to the target

This technique typically provides pointing to <1", and is limited mainly by the astrometry of the bright stars and the targets.

5.5.2 Guiding

SWIFT now has its own guide camera, which is mounted on the top side of the AO bench. The guide field is ~80x60 arcseconds, located ~4 arcminutes off-axis from the SWIFT field. Please see the associated manual for the guider [AD1].

5.5.3 Offsetting

Offsets should be done with the `tcs offset` command. Remember to stop and restart guiding manually when you offset.

5.5.4 Combing long observation sequences

Guider images can be saved to provide a reference position for combining long sequences on faint objects (where there is insufficient signal in each frame to align on target). See the guider manual [AD1] for details on how to take and save these images.

Note that there is some flexure between the guider and the IFU. If you are running very long (many hour) sequences, it would be worth taking a 'reference image' position (where you have a star in the IFU and star in the guider) every few hours to calibrate this drift.

5.5.5 AO flatmap & focus correction

The AO system can correct the static aberrations of the primary mirror (quite significant) and correct focus telescope by making a "flatmap". This should be done even if you are not using AO correction for science. Ask the TO to make a new flat map every few hours, or after every significant slew. You will also need a new flatmap if you change the rotator angle.

The AO system can also correct the telescope focus by closing the loops and allowing it to offload focus to the telescope for a minute or so until stable.

5.6 Recommended Calibrations

The sections below list the recommended calibrations to take with SWIFT data, including the relevant instrument settings and exposure times. A summary of these settings can be found on the left hand side of the SWIFTWS desk for easy reference.

Sequence files (§5.3.4 above) exist which will take the default set of calibrations in each/all of the scales.

5.6.1 BIAS frames

We have found little/no bias structure in the SWIFT detectors which is not taken out by the overscan regions. Separate bias frames are not usually taken. If you do want to take them, take a dark with an exposure length of 0.001s

5.6.2 DARK frames

Same length as your science observations. If you plan to use darks, take at least 3, preferably more. Many observers do not use darks, and instead rely on the invariable sky subtraction frames to remove the dark signal.

5.6.3 FLAT frames

Turn on Halogen, Set field mask to 'Open'. Take at least 3, preferably more. Due to the large wavelength range of SWIFT, and the spectral profile of the halogen lamp, it is difficult to get good flux at the extremes of the wavelength range without saturating the middle. It is recommended to take multiple flat exposures to reduce the noise particularly in the blue. It may be worth taking flat frames at the same position (HA and declination) as your science data, though it should be possible to correct for flexure in data reduction.

Scale	Exposure Time	Halogen Voltage	Focal Plane Mask
235mas	0.5s	3.5	Open
80mas	0.5s	5.6	Open
16mas	3s	12	Open

Table 4: FLAT frame settings

5.6.4 ARC frames

Used to measure the wavelength solution of the data. Turn on Neon and Argon, Set field mask to 'Open'. One frame should suffice, but more may be preferred. It should be possible to compensate flexure by secondary wavelength calibration on the sky lines (at least in the 235mas scale), but you may want to be safe and take Arc frames at the position of your science data (particularly relevant in the 16mas scale).

Scale	Exposure Time	Iris Setting	Focal Plane Mask
235mas	0.5s	Currently N/A	Open
80mas	4s	Currently N/A	Open
16mas	100s	Currently N/A	Open

Table 5: ARC frame settings

5.6.5 Vertical TRACE frames

Used to align the individual slices to make a reconstructed image. Turn on Halogen. Set field mask to 'Vertical Lines'. One frame should suffice. Possibly want to take Trace frames at the position of your science data, but it is not mandatory.

Scale	Exposure Time	Halogen Voltage	Focal Plane Mask
235mas	0.5s	3.5	CoarseVerticalLines
80mas	0.5s	5.6	FineVerticalLines
16mas	4s	12	FineVerticalLines

Table 6: VTRACE frame settings

5.6.6 Horizontal TRACE frames

Used to correct for field distortion in the pre-optics and image slicer. Turn on Halogen. Set field mask to 'Horizontal Lines'. One frame should suffice.

Scale	Exposure Time	Halogen Voltage	Focal Plane Mask
235mas	0.5s	3.5	CoarseHorizontalLines
80mas	0.5s	5.6	CoarseHorizontalLines
16mas	4s	12	FineHorizontalLines

Table 7: HTRACE frame settings

5.6.7 PSF image / Flexure maps

Engineering use to check the stability of the instrument focus; please take one per run. Neon and Argon lamps on.

Scale	Exposure Time	Iris Setting	Focal Plane Mask
235mas	0.5s	Currently N/A	CoarseVerticalLines

Table 8: PSF frame settings

Another use of these frames is to map the flexure of the instrument at different points across the sky, which can help increase the quality of the data reduction. To take a flexure data set, take a PSF image at zenith, and then at the positions of your observations (including at each position angle used). One frame per hour of tracking is sufficient to map the flexure. Calculating the flexure function and applying it to the calibrations is not explicitly included in the pipeline however.

It is acceptable to use the DCS (engineering) GUI to take flexure data. This offers a much faster readout (~40s) at the expense of higher read-noise.

5.6.8 Gain correction

Once each run, it is worth taking a set of data to calibrate the gain of the detectors. A gain data set consists of pairs of flat fields taken with varying exposure times. Suggest using the 80mas pixel scale, and using exposure times of 0.5,1,2,3,5,7,10 seconds. Take two exposures at each exposure time.

5.6.9 Sky Flats

Once each run, take a set of evening/morning twilight frames in each pixel scale used to enable the pipeline to properly correct the illumination pattern on the slicer field. Adjust exposure time until you have ~40,000 counts at the peak. Due to the long read-out times, you will probably have to increase/decrease the exposure times as the sky brightness changes (TBC). Aim to take at least 3--5 frames.

5.6.10 Telluric stars

You may want to observe telluric stars to correct for atmospheric water absorption. The Gemini telluric page is a good resource for finding telluric standards:

http://www.gemini.edu/sciops/instruments/nifs/near-ir-resources?q=node/10175.

Suggest using B0 A1 A2 spectral types. Remember to observe the star in both halves of the IFU if required by the science object.

5.6.11 Spectrophotometric standards

If conditions allow (i.e. photometric), please observe a spectrophotometric standard at least once during the run. As well as allowing you to correct your data for the instrument response function, it allows the instrument team to monitor the throughput of the instrument, and pick up any potential problems. Remember to observe the star in both halves of the IFU. The Oke or Humay lists from this page are good choices:

http://www.eso.org/sci/observing/tools/standards/spectra/

5.7 Data reduction pipeline

SWIFTWS includes a version of the SWIFT pipeline which will process images as they arrive from SWIFTLCU. It takes ~2 minutes to generate a cube. To start the pipeline;

```
swiftws$>    drpipeline.sh
```

Once cubes have been reduced, they will appear in the nightly data directory on SWIFTWS as msXXX.sci.cube.fits, where XXX is the file number. The pipeline also copies over the IRAF log from the reduction process as msXXX.drlog

The code for the pipeline is stored in `/home/swift/Pipeline/` on SWIFTWS. This contains several relevant files/directories:

swiftredMS_XXXmas.par IRAF parameters files for each of the three scales, defining how the swiftredMS task should run. The parameters in these should match those used for making the calibrations (i.e. the definition of the slit upper and lower bounds)

CalsXXXmas/ Three directories containing the calibration files for each scale

PIPELINE.QUEUE A text file containing a list of files waiting to be reduced. Files are 'popped' off the top of this list as they are processed by the pipeline.

WDNNNNNNNNN/ A working directory used by the pipeline for the reduction process. The number of the directory is the unix timestamp when the pipeline started the reduction – old directories can be safely deleted as they result from a reduction which failed for some reason. The WD is deleted on successful reduction of a file (once the result + log have been copied to the nightly directories)

5.7.1 Sending files to the pipeline

The pipeline monitors for new files arriving from SWIFTLCU, and automatically adds them to the reduction queue. To add a file (e.g. master001.fits) to the pipeline queue manually, if it didn't reduce for some reason, do;

```
swiftws$>    ls /home/swift/data/2014-04-02/master001.fits >>
             /home/swift/data/Pipeline/PIPELINE.QUEUE
```

(*note* the double ">>" rather than just ">"). You need only add the master file to the queue – the corresponding slave file will be picked up automatically.

5.7.2 Generating new calibration files

New calibration files can be generated by running the pipeline in the standard fashion on a full set of calibrations. The `swift_tarcals` task in the IRAF package can be used to collect all the relevant calibrations together.

These new calibration files should be copied over to a sub-directory in `/home/swift/Pipeline/`, and that sub-directory sym-linked to `CalsXXXmas` (where XXX =

scale, i.e. 235/080/016). The parameters used to generate the calibrations should match those in the `swiftredMS_XXXmas.par` files

5.8 How to back up data from SWIFTWS

Users are responsible for backing up their own data at the end of the run. Data is typically left on swiftws for a long time, but it may be deleted at any time and without warning.

5.8.1 DVD backup

Use the DVD burning programme `GnomeBaker` on the desktop of SWIFTWS. Create a ``new data DVD'' project, and add the required data directories to it (The data are kept in `/home/swift/data/`, with a new directory is created for each night's data (e.g. `2010-01-06/`)). Click to burn. You may need to split the data over multiple discs if you've taken more than ~4.5Gb of data.

5.8.2 USB discs

You can transfer data onto a USB disc from swiftws.

5.8.3 Oxford observers

Oxford observers, please look on the sharepoint site (https://drewett.physics.ox.ac.uk/astro/swift/) for instructions on how to transfer the data back to Oxford directly.

6 NON-STANDARD TASKS

The tasks in this section are not usually needed for a normal observing run. They are usually carried out by the instrument team from time-to-time.

6.1 Pupil alignment

This is an expert user/instrument team job. Ultimately it should be checked by the observatory staff as part of installation.

To maximize throughput, the SWIFT pupil must be matched to the AO pupil (which in turn must be matched to the telescope pupil). FM3 in the SWIFT pre-optics is adjusted to align the AO pupil with the SWIFT pupil. This should not change significantly between observing runs, but it is good to check this at the start of every run.

Note, Jan 2013: We have found the mounting repeatability of SWIFT on the AO system to be excellent, and the pupil alignment is generally consistent to 1—2% from run to run (over many years). You should not need to check or adjust the pupil alignment during a normal observing run, but this information is included for reference.

6.1.1 Checking the pupil alignment

SWIFT contains a set of pupil imaging optics to check the location of the pupil. These optics, which are manually flipped into the pre-optics beam, create an image of the pupil on the image slicer, which can then be reconstructed with the RTD software. To check the alignment of the pupil, do the following;

1. Image the pinhole with the halogen lamp
2. Image the AO stimulus, and move it to the same location as the pinhole
3. Open the top cover to the pre-optics by undoing the screws near the cables. The lid swings down.
4. Carefully flip in the pupil imaging optics
5. Image the instrument pupil (with the pinhole in field mask)
6. Image the telescope pupil, and check if it lies within the instrument pupil (it is useful, but not necessary, to have the pupil mask on the DM which recreates the telescope central obstruction).
7. Flip the pupil imaging optics back out of the beam, and close up the pre-optics cover.

If the above instructions aren't obvious to you, please contact a member of the instrument before trying it! The images below (Figure 9) show how the pupils should look in the reconstructed image, showing the AO pupil comfortably inside (slightly to the right) of the instrument pupil.

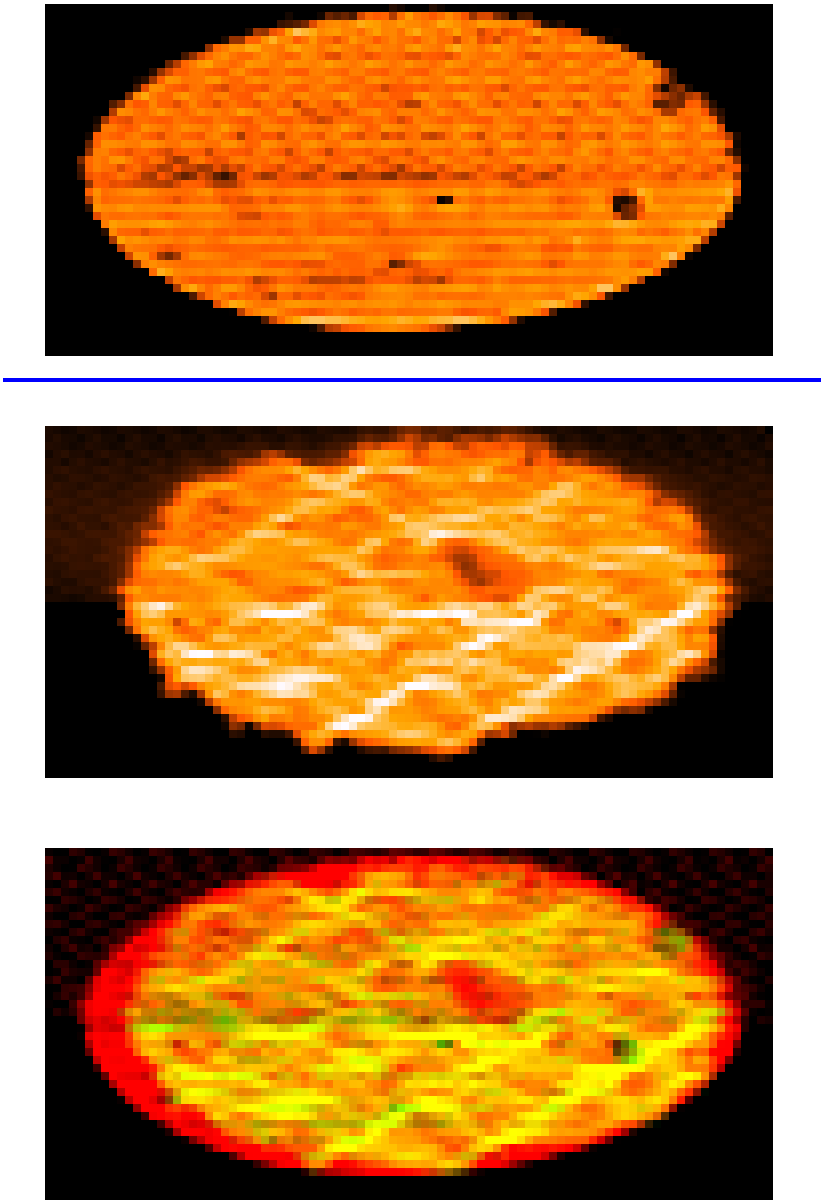

Figure 9: From top to bottom; Instrument internal pupil, AO pupil, and overlay.

6.1.2 Adjusting the pupil alignment

This is <u>definitely</u> an expert-user only procedure

If the telescope pupil does not lie within the instrument pupil, the instrument will lose throughput. To correct this, you need to adjust FM3 in the pre-optics. This involves removing the whole pre-optics cover, and should not be attempted if you are not sure of what you are doing. Please contact a member of the instrument team before doing this.

To add:

- Picture of FM3 denoting screws

- Empirical list of how pupil image moves when screws are turned.

6.2 How to make reconstruction vectors

If the reconstruction looks really bad, you may need to recalculate the reconstruction vectors. If this is the case *please let Matthias and Fraser know; it implies something has moved in the instrument!*. To recalculate them;

1. take a Halogen lamp frame with the vertical line mask in.
2. In IRAF, measure the position of the central trace for each slit (do this for both master and slave)
    a. start at the bottom of the frame, low y values and work up.
    b. use the region around x=2000
    c. imexam + logging + 'k' to fit a vertical guassian is pretty efficient method (the slit spacing is ~93 pixels, if you get confused about which line is which!)
3. extract the y position of each trace from the log file.
4. "cat" the y positions for the slave and master into one file (with 44 lines). The slave should be in lines 1--22, and the master in 23--44 (yes, I know it's the wrong way round!). These are the reconstruction vectors for the RTD.
5. You will probably find that the absolute position of the vectors is not quite right (the edge of the field isn't at the edge of the reconstructed image). Play around with offsets for the vectors to adjust this. I use `awk '{print \$1-N.N}' vectors > newvectors` to adjust the offset (where N.N is the offset).

A reconstructed image (using the 'PSF' band) of the vertical line mask should have straight lines! Figure 10 shows an example of a good reconstruction in the 235mas scale. The lines should appear straight and aligned between the two detectors. Note that the quality of the reconstruction changes with wavelength, due to optical effects in the camera which are not corrected by the RTD code (they are corrected by the science pipeline).

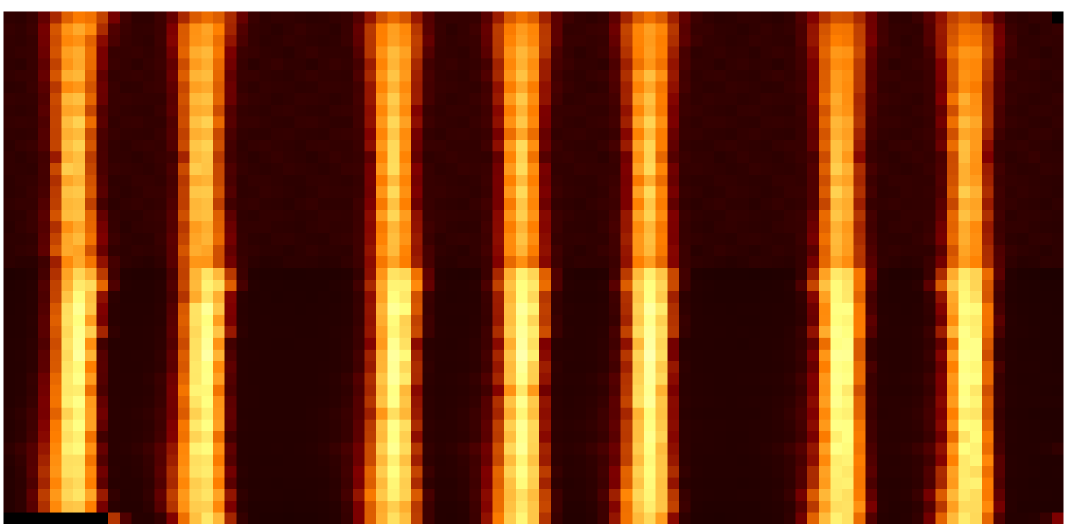

Figure 10: Reconstructed image of the vertical line mask in the 235mas scale.

7 TROUBLE-SHOOTING

7.1 Instrument team contact details

For serious issues with the instrument, please contact one of the instrument team. For software issues, please contact Fraser Clarke first.

	Fraser Clarke	Matthias Tecza	Niranjan Thatte
Email	fraser.clarke@physics.ox.ac.uk	m.tecza1@physics.ox.ac.uk	n.thatte1@physics.ox.ac.uk
Skype	fraser.clarke	matthiastecza	niranjanthatte
Office	+44-1865-283140	+44-1865-273364	+44-1865-273412

Table 9: Instrument team contact details

7.2 Known issues/bugs and workarounds

7.2.1 Can't take an exposure

You may see a message in the DCS GUI about "file ready event is set", and see two lines in the ipython terminal saying "You are already taking an exposure" (one line is normal). This seems to be a bug in the GUI/python code not clearing an event on the detector. Solution is to quit the GUI and ipython and restart. You should not have to reload timing files etc when you restart as the detectors are already running.

7.2.2 Communication lost to detectors

If you get a lot of error text appearing in the ipython window of the detector control, communication to the detector controller PCI cards has probably been lost (key sign is an error relating to "ioctl").

Sometimes this can be recovered in software by issuing a `master.reset() ; slave.reset()` command in the ipython window. You may need to try this several times. If this fails, try to close the DCS GUI and restart – check that you don't get a lot of error messages on start-up. Again, you may need to try several times.

If you can recover communication this way, it is worth powering down the detectors and power-cycling the SDSU boxes (via the power control GUI in the ICS) – this seems to help avoid the PCI cards failing. After the power cycle, the restart the DCS which should come up as normal.

If the software techniques fail, the PCI cards in the back of SWIFTLCU will need to be reset manually by pressing the white buttons on the PCI cards. Restart the DCS GUI after resetting the cards.

7.2.3 SWIFTLCU hangs completely

This is caused by a problem with the USB drivers. The only solution is a hard reboot of the machine. SWIFTLCU is located in the computer room, at the bottom of the second rack from the right. It will take 2—5 minutes to come back.

7.2.4 Detector didn't read out

Click the "Write FITS file" button on the DCS GUI. This will write the current buffer to a file, which may solve the problem. It will however make a few copies and may desynchronise the master & slave numbers, so hit "Initialise" on the GUI once done.

If this doesn't clear the problem, try clicking on "Abort", which will give a number of options.

If this doesn't work, try typing "`master.abort_exposure() ; slave.abort_exposure()`" and/or "`master.abort_readout() ; slave.about_readout()`" on the ipython command line.

If this doesn't work, try to quit and restart ipython. If it starts up with an error message about memory, close the windows and try again. If it fails after 3—4 attempts, it is probably an issue with the PCI cards, and the quickest route will be to reboot swiftlcu (about 3--5 minutes). You will unfortunately lose the exposure in this case.

7.2.5 IP addresses/can't connect to swiftXX

The machine is probably down/crashed. Physically check if possible, or power cycle with the network power supply.

The SWIFT machines should be on fixed IP addresses (Table 10) and should normally be addressed by their name rather than IP addresses. We have found occasionally the IP address has changed to something different (usually WS or LCU), which causes problems as the software cannot communicate.

To fix this, either update the software/config files with the correct address, or (a more universal fix), edit the `/etc/hosts` file with the new IP address (log onto the problem machine locally, and type`/sbin/ifconfig`).

HOST	IP	DESCRIPTION
SWIFTWS	198.202.125.171	SWIFT Workstation. User machine located on the instrument bench in the data room
SWIFTLCU	192.202.125.172	SWIFT control computer. Located in the computer room on the mezzanine floor. Bottom of the second rack from the right as you walk into the room.
SWIFTTS	198.202.125.173	SWIFT terminal server. Mounted in the electronics rack in Cassegrain cage
SWIFTNPS	198.202.125.174	SWIFT network power supply. Mounted in the electronics rack in the Cassegrain cage
SWIFTIC	198.202.125.175	SWIFT guide camera computer. Windows machine mounted on the instrument bench. Controls the guide camera mounted on the PALM3K bench. Also controls the IRIS and BLADE functions within SWIFT.

Table 10 : Normal SWIFT computer IP addresses

7.2.6 TCS information not updating

Update Jan 2013 -- much more stable now

Either the TCS is not connected in ArcVIEW, or the database updater has died in ArcVIEW (check other information on the instrument status pages). The latter requires a restart of ArcVIEW.

7.2.7 Can't offset telescope

Check that the TCS is connected to ArcVIEW (type `tcs connect` from the command line).

7.2.8 Serial connection errors

This usually happens with the LS218 and LS325 modules. If it is intermittent, it is probably just another process trying to talk to the modules at the same time. Try again later... If it persists (and is a problem on system start-up), you may need to check the device ports are properly configured on SWIFTLCU.

On the LCU, try running `NIvisaic` and see if it detects the `ARSL[11-15]:INSTR` serial ports. If it does not, sometimes the default permissions on `/dev/ttyS[1-4]` have reverted back to root only (don't know why). As root on SWIFTLCU, do;

```
swiftlcu$>   chmod 666 /dev/ttyS?
```

If the serial ports `ttyR[1-5]` don't exist, then the ttyredirector service has failed to start up. Log in as root on SWIFTLCU, and do;

```
swiftlcu$>   service ttyredirectord start
```

7.2.9 Can't connect to motor controller

Check if the serial ports /dev/ttyR[1-5] exist (on swiftlcu). If not, see the section above about restarting them.

Occasionally the terminal server gets confused, and needs a hard reboot. You can do this from the power control GUI. This is very rare though.

Similarly, the motor controller occasionally needs to be rebooted. This needs to be done in the cass cage.

7.2.10 Halogen lamps won't turn on

This could be one of several causes with the remote power supply in the Cassegrain rack. First try to clear any error condition (caused I think by turning on the power supply before it is plugged in). There will be an error LED flashing on the front of the power supply. You can clear this with:

```
swiftlcu$>   ttipsu PASS *LSR?
swiftlcu$>   ttipsu PASS *RST
```

The probable cause of this error condition is a bad current limit (which we think is caused by the internal electronics resetting themselves when turned off for a long time). This can be checked and changed by:

```
swiftlcu$>   ttipsu PASS I1? (should return 0.42)
swiftlcu$>   ttipsu PASS I1 0.42
```

That resets the current limit suitable for the halogen lamp (a 5W lamp, which runs at 12V for the 16mas scale calibrations). You should now be able to use the normal commands to turn the lamp on. If this still fails, there is a change the lamp has blown and will need replaced!

7.2.11 (multiple) ArcVIEW problems

7.2.11.1 Loss of communications with ArcVIEW

This seems to happen occasionally, and it seems to be the TCP connection with ArcVIEW that dies. ArcVIEW continues to run, but you cannot communicate with it. Only solution seems to be a restart of ArcVIEW. Perhaps caused by too many simultaneous requests to ArcVIEW

7.2.11.2 ERROR 111 writing_message

This error occurs when the ArcVIEW TCP port is somehow blocked. I suspect this is usually from a rouge process taking a while to timeout. It usually fixes itself after 10-20 seconds. If not, a reboot of ArcVIEW should clear it.

7.2.11.3 ArcVIEW won't shut down

Occasionally one of the ArcVIEW modules (usually `FITS_header`) will hang up and stop ArcVIEW shutting down cleanly. The only solution seems to be to find the labview process and kill it;

```
swiftlcu$>   ps -aux | grep labview
swiftlcu$>   kill XXXX (where XXXX is the process ID of labview)
```

7.2.11.4 Can't restart a module

I don't know why this doesn't work anymore.... only solution is to restart ArcVIEW.

7.2.11.5 ArcVIEW keeps firing up error messages on startup

Please make a note of the error message and let Fraser Clarke know (usually this is caused by an edit to a file which hasn't been properly propagated through the whole code tree). In the meantime, either just click OK/Continue/Accept, or try restarting ArcVIEW

7.3 Spectral Features

There are several spectral "features" in the SWIFT data.

7.3.1 9500Å emission

This scattered light feature has been identified and removed. A very slight feature remains at the level of ~10cnts/hr.

7.3.2 Emission "chains"

Sometimes there are several small emission features throughout the spectrum, taking the form of a ``chain'' at slight angle across the slices. We suspect these come from flourescent lamps somewhere in the dome -- check the door to the dome is closed, and the table lamp has been turned off.

7.3.3 Fringing

There is some fringing visible beyond ~9600Å. It is at quite a low level, but high spatial frequency. Only noticeable in frames with very high counts (i.e. flats)