

Foswiki > FLAO Web > LbtTel > Oct2011Notes (29 Mar 2014, AlfioPuglisi)

PISCES images

We have a mechanism to automatically start an OptLoopData whenever PISCES saves an image.

The work is done by the PiscesGui, which must be started with this command:

```
[AOeng@wfsdx .] PiscesGui.py
```

keep the xterm open, because the GUI will write on stdout the tracking numbers as they are started, etc.

The sequence is as follows:

1. PiscesGui monitors the status of Pisces opening a socket connection to the Pisces server.
2. when the status goes from IDLE to BUSY, an OptLoopData is triggered. The length of this data may be reduced in order to keep it shorter than Pisces exposure time.
3. when the images are done, Doug's IDL software writes a file in /tmp/piscesdir.txt on wfsdx containing the directory where the images have been stored.
4. PiscesGui wait for this file to appear, copies the image data, and delete the file
5. Repeat from step 2

Related files

(on wfsdx) PiscesGui.py: triggers the OptLoopData and copies data around

(on any obs* computer) ~LBTO/idl/wfsc/pisces/pisces_check_image.pro : after the end of the image cube, calls "trigger_wfsdx"

(on any obs* computer) ~LBTO/idl/wfsc/observe/trigger_wfsdx.pro : writes a file on wfsdx with the directory where the image is taken

Known bugs

Sometimes the Pisces server dies and must be restarted manually. PiscesGui tries to reattach, but often fails and does not process cubes anymore. In this case the Gui must be closed and restarted.

Also, the timing between optical loop data and image cubes is a bit tricky, and it may happen that the GUI goes out of sync. The stdout log will become confused and/or stop completely. Also in this case the Gui must be restarted. This seems to happen with images with a short exposure time: it could be that the reduction in the number of OptLoopData frames is incorrect.

How to correlate Pisces images with OptLoopData

The OptLoopData should already contain the related Pisces image.

In the other direction, a Pisces image has an "UTSTART" keyword in the header, with the starting time in UT.

This should coincide within a few seconds with the optical loop data tracking number. (in case of cubes, only the first image will match the time).

Sinusoidal gain disturbance

The AOArb has a new feature to inject a sinusoidal disturbance every time a loop is closed. This feature is activated by a parameter in AOARB.conf:

```
doOptLoopDisturb int 0
```

if set to "1", the feature is activated for all loops closed by StartAO, e.g. by the AOS. To activate/deactivate the feature, the AOArbitrator must be restarted to re-read the configuration file.

The disturbance to load is a file in the following format:

```
/towerdata/adsec_calib/CMD/disturb/optloopgain_XXXX.fits
```

where "XXXX" is the current loop framerate in Hz (four digits with the format %04d).

If no file is found, no disturbance is applied but no error is generated.

The disturbance files are generated by the IDL procedure (tested on wfsdx but it should work on adsecdx too):

```
$ADOPT_SOURCE/idl/wfs_lib/sinusmode_optloopgain.pro
```

which accept arguments with the following defaults:

MODES = [3, 10, 150, 200] : which modes to modulate (they will be summed up in a single disturbance files)

AMPS = [2e-9, 2e-9, 2e-9, 2e-9] : for each mode, modulation amplitude in meters

LOOPFREQS = [100,200,312,400,625,800,990] : loop framerates to be generated (each framerate will need a different disturbance)

FREQBIN = 5.7 : frequency bin to shift the modulated modes

BASE = 'KL_v7' : modal basis to use

The script takes some minutes to run and generate the optloopgain_XXXX files directly in the right place.

Intermediate files (CmdDisturb_XXXX.fits and so on) are removed.

Related files

`$ADOPT_SOURCE/idl/wfs_lib/sinusmode_optloopgain.pro` : procedure to generate the disturbance files. Run with no arguments for the default values, or modify the defaults inside the .pro

`/towerdata/adsec_calib/CMD/disturb/optloopgain_XXXX.fits` : disturbance files generated by the .pro above

`$ADOPT_ROOT/conf/adsec/current/processConf/AOARB.conf` : AOArb configuration file where the "doOptLoopDisturb" keyword must be set.

Known problems

When the feature is activated, the loop may have problems during StopAO because the disturbance MUST be disabled by the wfs slope header. If the WFS has a problem and slopes stop arriving, the loop cannot be automatically opened. The following recovery procedure should work:

1. set the gain to zero on AdSec Control GUI
2. disable the disturbance on the Wfs Control GUI
3. close the loop on the Wfs Control GUI. If the command cannot be executed, set to 1 the variable "slopecompctrl.R.FASTLINK.ENABLE.REQ"
4. open the loop on the Wfs Control GUI, or setting to zero the variable
5. if the variable was used, also press "StopAO" on the AdSec control GUI.

Known bugs of the AO system

List of GUIs to keep open to look for problems:

- CCD39 viewer
- CCD47 viewer
- Wfs Hardware gui on the "Pupil rerotator" panel
- Wfs Control Gui to look at the wfs log.

Bcu39 stuck

We still had some instances of the bcu39 stopping. This always happened during the PresetAO command.

Symptom

the ccd39 display goes "not live" with a red indicator. This should never happen after the system has been turned on.

Recovery

The bcu39 must be resetted and reconfigured:

1. Open the wfs hardware GUI
2. in the Power controller panel, turn on the "bcu39 reset", then turn it off
3. in the Status check panel, wait for the bcu39 to go red and then green again after a few seconds
4. in the ccd39 panel, set the binning in the drop-down box selecting the same binning as before
5. in the Wfs Control GUI, press "Acquire HO Dark" to take the dark again

If you have been quick enough, the acquisition sequence will go through automatically. If not, it will fail, and the Preset must be restarted.

Possible solution

the Python code for the preset was multi-threaded to configure the ccd39 and ccd47 in parallel and save time. This code has been refactored to be sequential. After this update, the bcu39 problem did not reappear, but it was only one night.

Ccd47 stuck

In a manner similar to the bcu39, the ccd47 sometimes failed during the [PresetAO](#). The BCU47 seemed fine, but the ccd47 would stop integrating, or refuse to change binning.

Symptom

- The ccd47 shows "not live" with a red indicator. This can happen for a few seconds after a change binning, but not afterwards.
- The ccd47 seems to work, but the Wfs Arbitrator log reports an error about the ccd47 binning or framerate.

Recovery

- If the ccd47 is not live:
 - Open the Wfs hardware GUI
 - in the ccd47 panel, simply press the "Start" button
 - if it has no effect, try to select the binning from the drop-down box
- If the ccd47 is live:
 - send the [PresetAO](#) again

Possible solution

the Python code for the preset was multi-threaded to configure the ccd39 and ccd47 in parallel and save time. This code has been refactored to be sequential. After this update, the ccd47 problem did not reappear, but it was only one night.

Rotator position update

It happened two times during the week that the pupil rerotator stops working during a loop, and the loop stops shortly after because of a skip frame. It appears that the variable notification between the two [MsgD](#) stop working, and the position update do not reach the wfs code.

Symptom

The pupil rerotator stops moving. During a loop, the rerotator state should switch between STATE_READY and STATE_OPERATING at about 1Hz. If this bug happens, the state will be only STATE_READY.

Recovery

The only known way is to restart the Wfs Arbitrator. This will cancel the preset.

1. Open the AO loop
2. Restart the Wfs arbitrator
3. send the Preset again

Possible solution

It seems to be a MsgD bug.

PISCES webcam

The webcam attached to PISCES can be read running the following script:

```
[AOeng@wfsdx] /home/AOeng/webcam.sh
```

the script will continuously acquire an image of the webcam and copy it on wfsdx with the name "webcam.jpg".

There is an HTML file which shows the image and refreshes it every two seconds:

```
/home/AOeng/webcam.html
```

On wfsdx, the Konqueror web browser is installed and can be used to open this html file.

PISCES focus

Using Doug's IDL routines, whenever the PISCES filter is set, a focus command (OffsetZ) is sent to the AO system. The command contains the absolute position of the Z stage for that filter, in mm from the homing position.

The position is read from a lookup table contained in this file:

```
[LBTO@obs3] /home/LBTO/idl/wfsc/pisces/pisces_set_camera.pro
```

which contains now the following table:

```

filter_list = [['0','000',0.0], $
               ['1','100', 51.8], $
               ['2','200',0.0], $
               ['3','300',0.0], $
               ['4','400', 51.1], $
               ['5','500',52.3], $
               ['6','600',51.7], $
               ['7','700',53.1], $ ; ND=53.1 ; no ND = 52.3
               ['8','800', 52.7], $ ; ND=52.7 ; no ND = 51.8
               ['9','900',0.0]]

```

Each line is:

[filter number, encoder position, focus position]

The values are the known focus positions on sky, without ND except where indicated.

If a focus position is modified (or the ND is installed/removed), this table must be updated and the IDL session must recompile this file.

PUPIL digital shift

To shift a pupil, there is a Python routine in the "makePupils" module.

- 1) start thaoshell.py (from an xterm, or from the wfseng button)
- 2) type the following:

```

import makePupils
makePupils.movePupils('old_tracknum', x, y, ccd_w, binning)

```

where:

old_tracknum: pupils to be shifted

x, y: amount of shift in x and y direction in pixels

ccd_w: ccd dimension (80 at bin1)

binning: binning (1,2,3 or 4)

example:

```

makePupils.movePupils('20100925_174242', 1, 0, 80, 1)

```

The procedure will generate the pupils and return at the end the new pupil tracking number. This number must be inserted in the usual AO calibration table to be used by presets, etc.

The new pupils are not reloaded automatically. To force a reload without a preset:

- set a new binning in the Wfs Control Gui, and then set the previous binning again
- or (faster), use the Wfs Hardware GUI and set again the current binning in the ccd39 binning drop-down box. After this, set the loop params on Wfs Control Gui and acquire a dark.

The GUI will always load the last generated pupils, while the preset will read the AO calibration table.

-- [AlfioPuglisi](#) - 31 Oct 2011

[Edit](#) | [Attach](#) | [Print version](#) | [History: r5 < r4 < r3 < r2](#) | [Backlinks](#) | [View wiki text](#) | [Edit wiki text](#) | [More topic actions](#)

Topic revision: r5 - 29 Mar 2014, AlfioPuglisi

Copyright © by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding Foswiki? Send feedback

