- Checkout svn oaa_lib code from Arcetri repository:

```
svn co svn+ssh://obelix.arcetri.astro.it/4lbt/svn/IdlTools/trunk ./idltools
```

- Add elab_lib folder and idltools folder to the IDL PATH
- Configure the environment variables:
  - $ADOPT_DATA
  - $ADOPT_ELAB
  - $ADOPT_PHASEMAPS

Environment variables

(Apologize: naming is wrong. Don't try to underdstand: $ADOPT_ELAB is a tmp folder, adsec_calib and adsec_data contains data of the entire FLAO system, not of AdSec alone)

$ADOPT_DATA must point to the root folder of the tree containing tracknums data and configurations. In $ADOPT_DATA the folders "adsec_calib" and "adsec_data" must exists.

$ADOPT_ELAB must point to a folder on which temporary files are stored. Better to have it on a local disk

$ADOPT_PHASEMAPS must point to a folder containing the phasemaps .sav files (typically named KL_vXX.sav)

Copying snapshot data and calibration data

- Create the folder $ADOPT_DATA (must end with "_right", e.g. "/home/dlmiller/elab/data_right")
- Create the corresponding "_left" folder (e.g. "/home/dlmiller/elab/data_left")
- Copy data in $ADOPT_DATA. You want to have copy /local/towerdata/adsec_calib and /local/towerdata/adsec_data. As an example, copying tracking number of 20140513 from adsecsx must results in a folder called $ADOPT_DATA/towerdata/adsec_data/20140513/Data_20140513_215858
- Copy calibration from /local/towerdata/adsec_calib/blabla... into $ADOPT_DATA/towerdata/adsec_calib/blabla...
- Copy phasemaps (files named KL_vXX.save) from adsecdx:/local/phase_maps or adsecsx:/local/phase_maps

Configuration Examples

Please copy from legolas.arcetri.astro.it on which a gbrusa account (usual password) is available

```
gbrusa@legolas:~$ printenv | grep ADOPT
ADOPT_DATA=/home/gbrusa/LBT_data_right
ADOPT_PHASEMAPS=/home/gbrusa/phase_maps
ADOPT_ELAB=/home/gbrusa/elab_right
gbrusa@legolas:~$ ls -l /home/gbrusa/LBT_data_right/
total 4
drwxr-xr-x+ 1 50026 50026     212 Sep 15  2013 adsec_calib
drwxrwxr-x+ 1 50026 50026    6184 Apr 22 21:40 adsec_data
```

You may want to add the following to

```
export ADOPT_DATA=$HOME/LBT_data_right
export ADOPT_ELAB=$HOME/elab_right
export ADOPT_PHASEMAPS=$HOME/phase_maps
```

Topic revision: r3 - 2014-05-22 - LorenzoBusoni

Getting started with ELAB

General considerations about Object-Oriented IDL

We assume the user is skilled with IDL use. Here follows just a small help about OO syntax with IDL

IDL version previous to 8.0

In IDL <8.0 the syntax is quite bad.

To create an object of class pippo (whose code is written in file "pippo__define.pro"):

```
IDL> instanceOfTheClass=  obj_new('pippo', argumentsToCreatePippoIfNeeded)
```

If obj is an instance of a class having function func() the following is needed to execute the function:

```
IDL> ret= obj->func()
```

If obj is an instance of a class having a procedure "pro" the following is needed to execute the procedure:

```
IDL> obj->pro, arguments
```

If obj s an instance of a class having function func() returning the instance of another class "subobj" having a function pippo() the followinf is needed to execute "pippo":

```
IDL>subobj= obj->func()
IDL>ret= subobj->pippo()
```

which is identical to:

```
IDL>ret= (obj->func())->pippo()
```

A real example, supposing "ee" is an object of aoelab class.

```
IDL> powerOfSlopeNumber120From12To50Hz= (ee->slopes())->power(120, from_freq=12, to_freq=50)
```

IDL version from 8.0

ElabLib can run on all IDL versions after 7.2 (as far as I know!). From 8.0 the OO syntax has been slightly simplified (even if the old one is still compatible) and "->" can be replaced with "." For instance creating an

object becomes:

```
IDL> instanceOfTheClass=  pippo(argumentsToCreatePippoIfNeeded))
```

While the last example becomes:

```
IDL> powerOfSlopeNumber120From12To50Hz= (ee.slopes()).power(120, from_freq=12, to_freq=50)
```

How to start elab on adsecdx or adsecsx

ElabLib on adsecdx and adsecsx is (usually) well configured and ready to use. Please don't use it (too much) during operations because it can slow down the system

```
% ssh AOeng@adsecdx  (usual passwd)
% aoidl
IDL> ao_init
```

How to start elab on a dedicated machine

- Follow installation instruction
- Starts "idl" and run "ao_init" to setup the elablib

How to restore a snapshot's data

Each snapshot of a system is identified by a date in the form YYYYMMDD_mmhhss ("tracking number", a.k.a. "tracknum")

To load the data the following is needed:

```
IDL> ee= getaoelab(YYYYMMDD_mmhhss)
```

This creates a variable named "ee" which is an instance of the class "aoelab". The aoelab class is the root of a hierarchy of other objects representing the full FLAO system.

Help

Because of a big number of methods available in an aoelab object (and in its sub-object tree) there is a (fairly) comprehensive on-line help that can be used to retrieve the name of the various functions during an interactive session.

Each object has a pro named help that accept a string as argument. Without argument, "obj->help" will print the full list of functions and procedures of the object obj, together with a small description. With the argument the above mentioned list will be filtered matching the passed argument.

In the 2nd example below, the functions of the ee object containing the string "filter" will be listed.

```
IDL> ee->help
IDL> ee->help, 'filter'
IDL> wfs = ee->wfs_status()
IDL> wfs->help
IDL> (ee->wfs_status())->help
```

Most useful commands

```
IDL> ee->modalplot
```

IRTC image object

```
IDL> (set->irtc())->help
IDL> image = (set->irtc())->image()
IDL> help, image
IDL> write_fits, 'filename', image, header
IDL> image_show, image[*,*,4], , /AS, /SH, /LOG
IDL> loadct, 3
IDL> image_show, image[150:225,120:180,4], /AS, /SH
IDL> (set->irtc())->show_psf
IDL> (set->irtc())->show_profile
IDL> cent = (set->irtc())->centroid()
IDL> plot, cent[*,0], cent[*,1]
IDL> print, ((set)->olmodes())->r0(/PLOT)
IDL> print, ((set)->olmodes())->seeing()
```

-- DougMiller - 24 Nov 2010

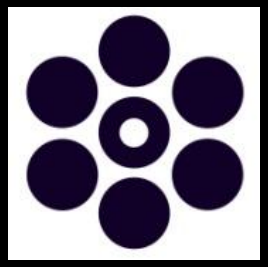
Topic revision: r4 - 2014-05-22 - LorenzoBusoni

Raw data: /adopt_data/towerdata/adsec_data/'date'/'tracking_number'

Processed _data: /adopt_data/elab/'date'/'tracking_number'

-- DougMiller - 24 Nov 2010

Topic revision: r1 - 2010-11-24 - DougMiller



GMT - NGWFSS

Preliminary Design Study

L. Busoni, F. Quiros Pacheco, A. Puglisi, M. Xompero, G. Agapito

ELAB – FLAO DATA PROCESSING TOOL

What's for

- Off-line diagnosis and performance analysis.
  - Evaluate performances of the AO system: Strehl Ratio, modal correction, transfer functions….
  - Compute complex quantities: e.g. seeing value from mirror commands and residual slopes
- Easy & quick check of system setup
  - "Damn, we forgot the calibration source on again!"
- Draw up statistics of the AO system
  - How is SR correlated with seeing values? And with telescope vibration?

Requirements

- AO data are periodically acquired and must be soon analyzed -> **fast** offline analysis
- Let's do it again. And again! -> A single command to perform a complex analysis
- Interactive and scripts-> IDL
- FLAO 1, FLAO2, LBTI, ARGOS… -> Modular
- Completely reproducible -> No input parameters
- Huge amount of data -> Smart use of resources, handle sets of measurements

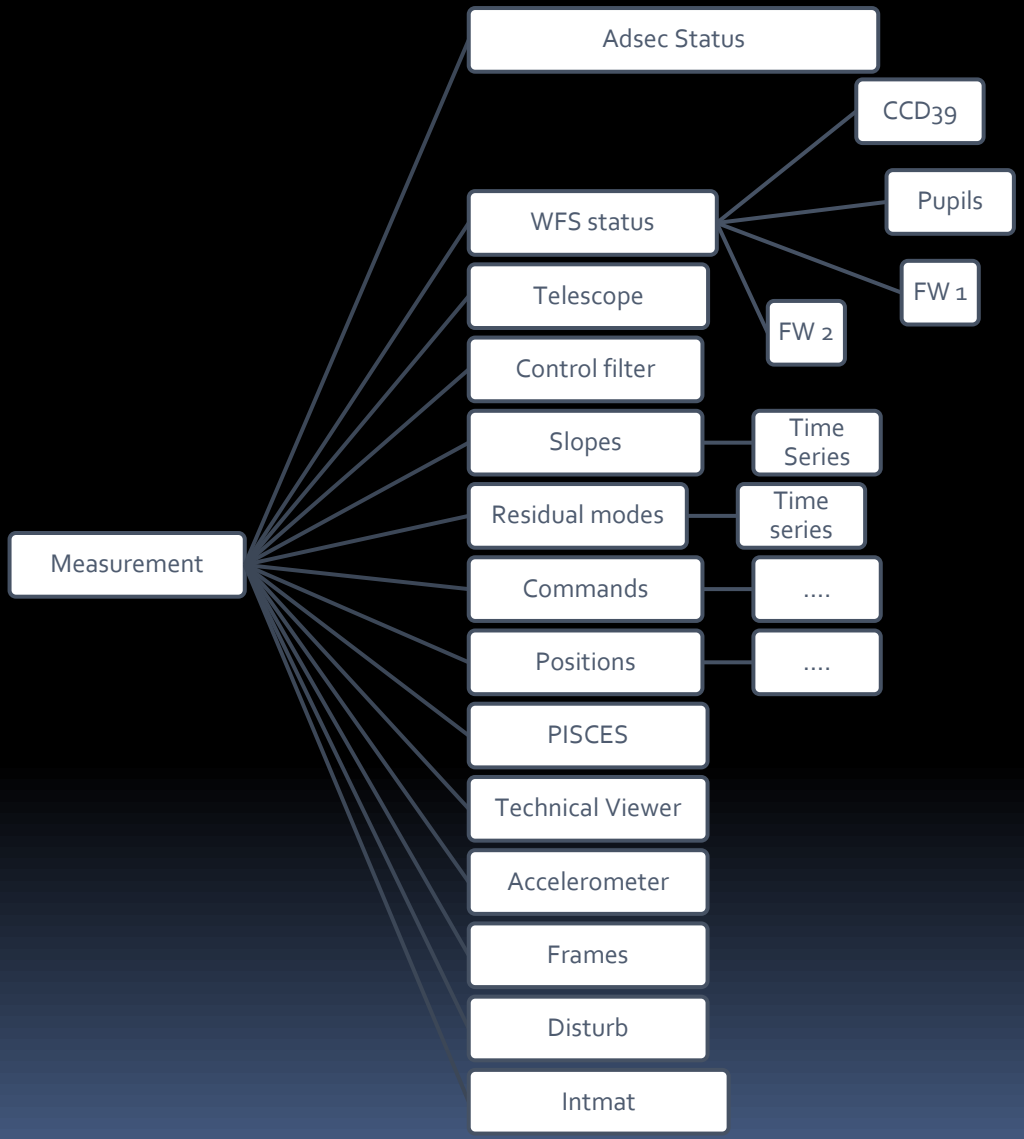
Input data

- Each measurement (tracking number) contains a snapshot of the system.
- Every slowly varying parameter (system configuration) is saved: motors position, CCD config, ASM config files, ….
- 4-20 sec of closed-loop data (slopes, mirror positions, ccd frames, …) at <400Hz are saved
- Relevant telescope parameters are saved
- Images of the scientific camera are saved

Data storage

- Data are stored as FITS files in a fixed folder structure.
- Every filename contains a Tracking Number in the format YYYYMMDD_HHMMSS
- FITS header contains information about possible parents: header of rec file stores the name of the interaction matrix from which originates, that, in turn, stores the parameters used for the IM acquisition.

Fuzzy Object Hierarchy



>500 methods

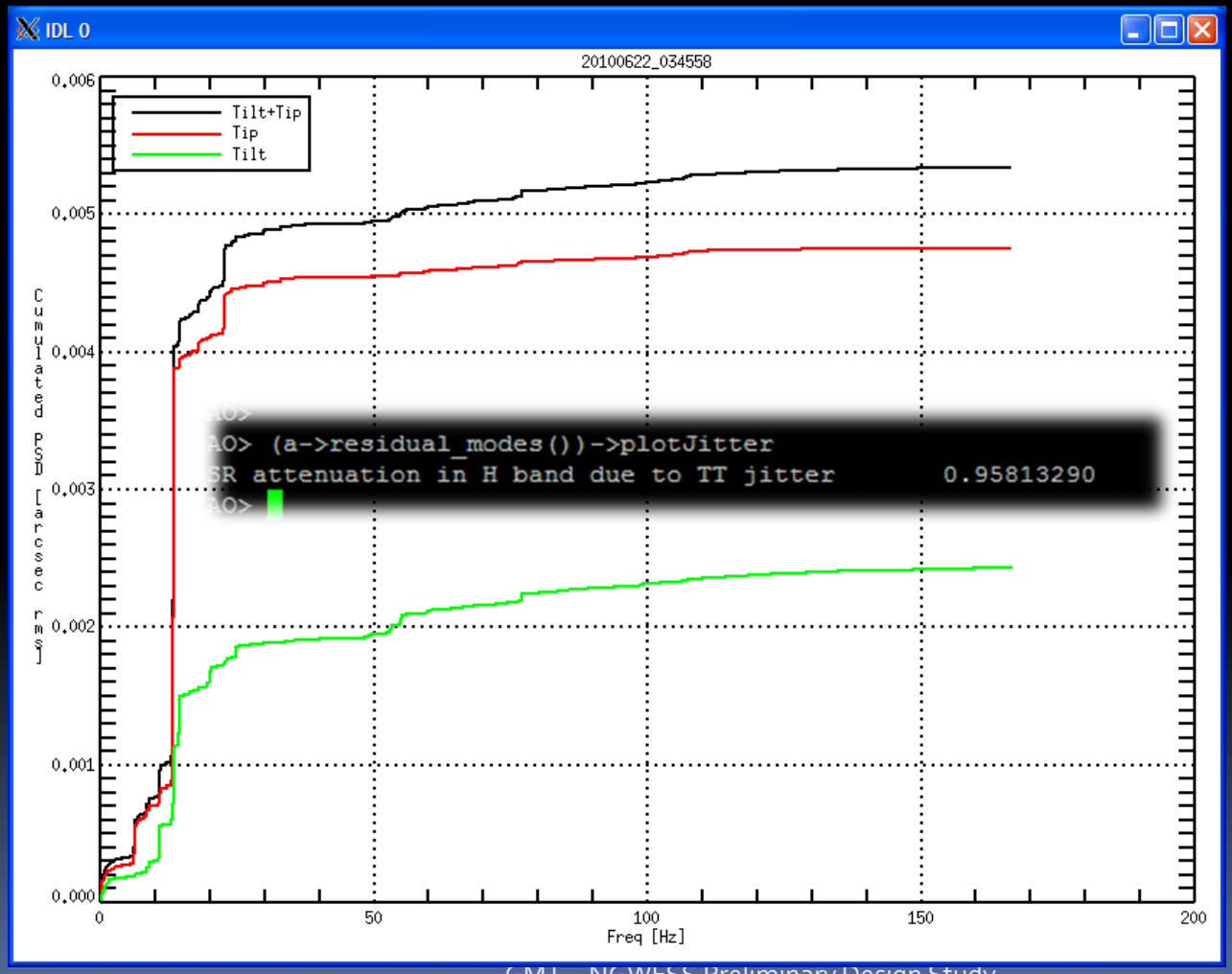
lbusoni@legolas: ~

```
AO> (a->wfs_status())->help
*** AOwfs_status                          : Represent WFS status
--- fitsfile()                                    : wfs status fitsfile name (string)
--- header()                                      : header of wfs status fitsfile (strarr)
--- wunit()                                       : W unit number (1,2,...)
--- ccd39()                                       : reference to ccd39 object
--- pupils()                                      : reference to pupils object
--- filtw1()                                      : reference to filter wheel 1 object
--- filtw2()                                      : reference to filter wheel 2 object
--- modulation()                                  : TT modulation (lambda/D)
--- rerotator()                                   : rirotator angle (degrees)
--- camera_lens()                                 : Position [x,y] of camera lens (mm)
--- stages()                                      : Position [x,y,z] of stages (mm)
--- lamp_intensity()                              : lamp intensity (a.u.)
--- cube_angle()                                  : cube rotator angle (degree)
--- cube_stage()                                  : cube stage position (mm)
--- slopes_null_fname()                           : slopesnull vector fitsfile name (string)
--- summary                                       : Summary of WFS status
****** AOccd39                          : Represent CCD39
------ framerate()                               : frame rate [Hz] (float)
------ readout_speed ()                          : readout speed [kpix/s] (float)
------ binning()                                 : binning (long)
------ status()                                  : status (string)
------ dark_fname()                              : filename of dark frame
****** AOpupils                         : Represent WFS pupils
------ indpup()                                  : pupil indexes  (lonarr)
------ nsub()                                    : number of valid subapertures (long)
------ radius()                                  : nominal radius of pupils (long[4])
------ cx()                                      : nominal x-coord of pupils centers (long[4])
------ cy()                                      : nominal y-coord of pupils centers (long[4])
------ real_radius()                            : measured radius of pupils (long[4])
------ real_cx()                                : measured x-coord of pupils centers (long[4])
------ real_cy()                                : measured y-coord of pupils centers (long[4])
------ real_side()                              : measured sides between pupils centers (long[4])
------ diffx()                                   : pupils centers error along x-axis (long[4])
------ diffy()                                   : pupils centers error along y-axis (long[4])
------ pup_tracknum()                            : pupils tracking number (string)
****** AOfiltw                          : Represent WFS filter wheel object
------ fw_number()                               : filter wheel number  (long)
------ fw_pos()                                  : filter position  (long)
------ fw_data()                                 : {name, reflectivity, transmittivity, central lambda, bandwidth}  (lo
------ name()                                    : filter name e.g. 'Dichroic 600-100 nm'
------ reflectivity()                            : Reflectivity
------ transmissivity()                          : Transmissivity
------ cw()                                      : Central wavelength (nm)
------ bw()                                      : Bandwidth (nm)
------ header()                                  : header of fitsfile (strarr)
```

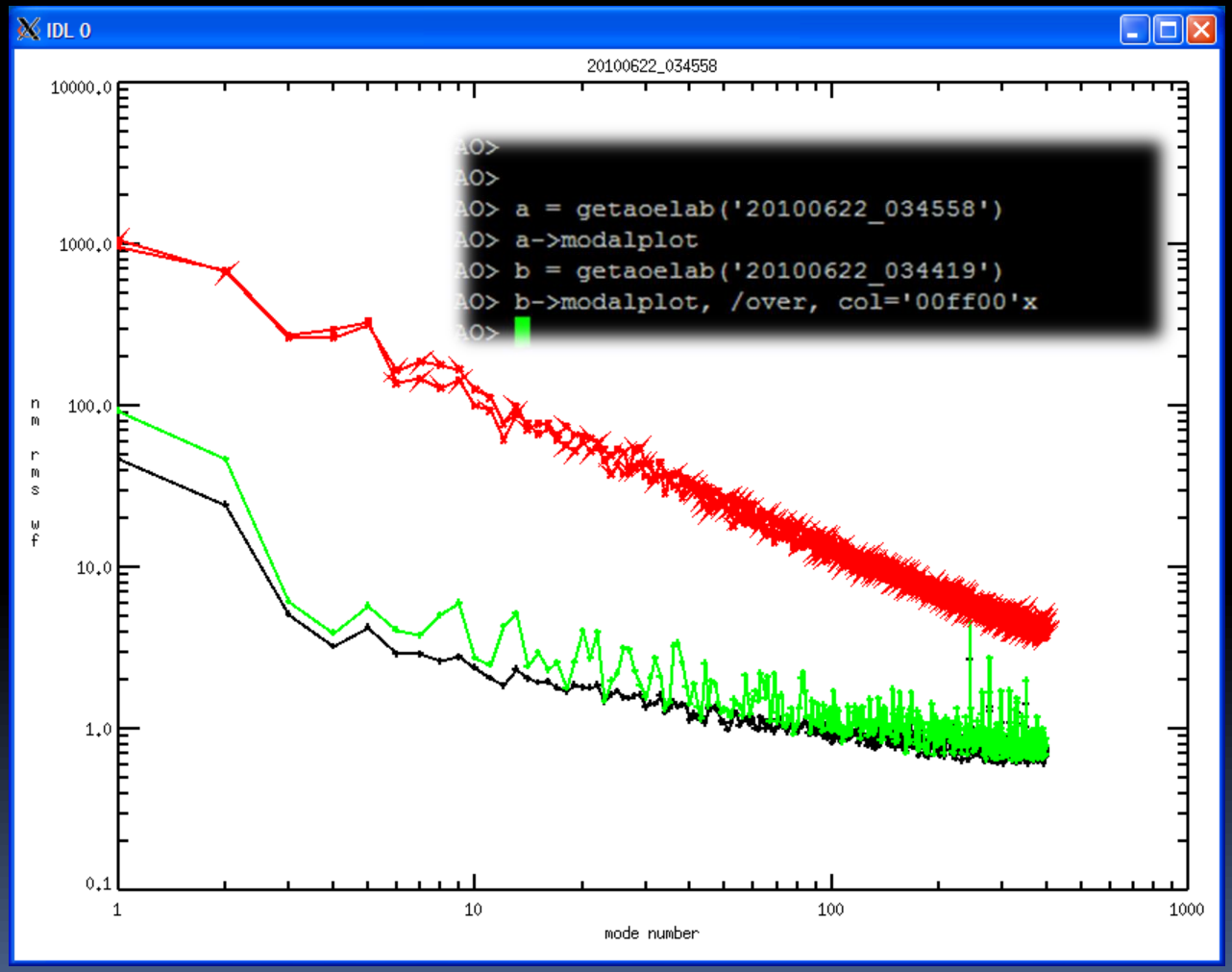
Implementation

- Big numbers: 12000+ measurements for FLAO1 commissioning, 2TB of stored data. A single measurement analysis can use 100MB of memory
- IDL Object Oriented
- Lazy initialization: data are computed when needed
- Singletons (RECs, IMs, pupil)
- Dynamic memory management (swap to disk)
- Analyzed data are stored in a temporary folder for fast retrieval (FFTs, processed frames,…)
- Time-series quantities inherit spectral analysis and filtering.
- Images inherit basic image processing

Spectral analysis of image jitter from WFS signals



Modal analysis of residual WF



Generate tables for wiki

aowiki.arcetri.astro.it/FLAO/LabLog20100622

Arcetri | ARGOS TWiki | La Repubblica | Corriere | ilmeteo.it | BSCW | Paschi Home | Facebook | WordReference | Gmail

First Light AO for LBT

Jump | Search

FLAO

Hello Lorenzo B
Log Out
Create person

FLAO Web
Create New
Index
Search
Changes
Notifications
Statistics
Preferences

Webs
APE
ARGOS
AtmDev
ERIS
FLAO
FLAOScience
FLAOtest
GMT
HOT
Magellan
Main
Private
Public
System
TecnoINAF2010
UpLoad
WLBTI

```
AO>
AO> set = obj_new('AOdataset', from='20100622_032500', to='20100622_040000')
AO> log_twiki, set, ref='Wref93'
 *TrackNo* | *RefStar* | *Mag* | *El* | *Wind* | *DIMM/OL* | *Rec* | *bin* | *#mod* | *freq* |
 *#frames* | *disturb* | *SN* | *notes* |
 20100622_032642 | Wref93 |  7.7 | 85 | 4 |   NaN  0.63 | 185716 | 1 | 400 | 1000 |  1.2   0.7
 20100622_032710 | Wref93 |  7.7 | 85 | 4 |   NaN  0.64 | 185716 | 1 | 400 | 1000 |  1.2   0.7
 20100622_032730 | Wref93 |  7.7 | 85 | 3 |  0.74  0.67 | 185716 | 1 | 400 | 1000 |  1.2   0.7
```

Seeing (DIMM) 1.0"

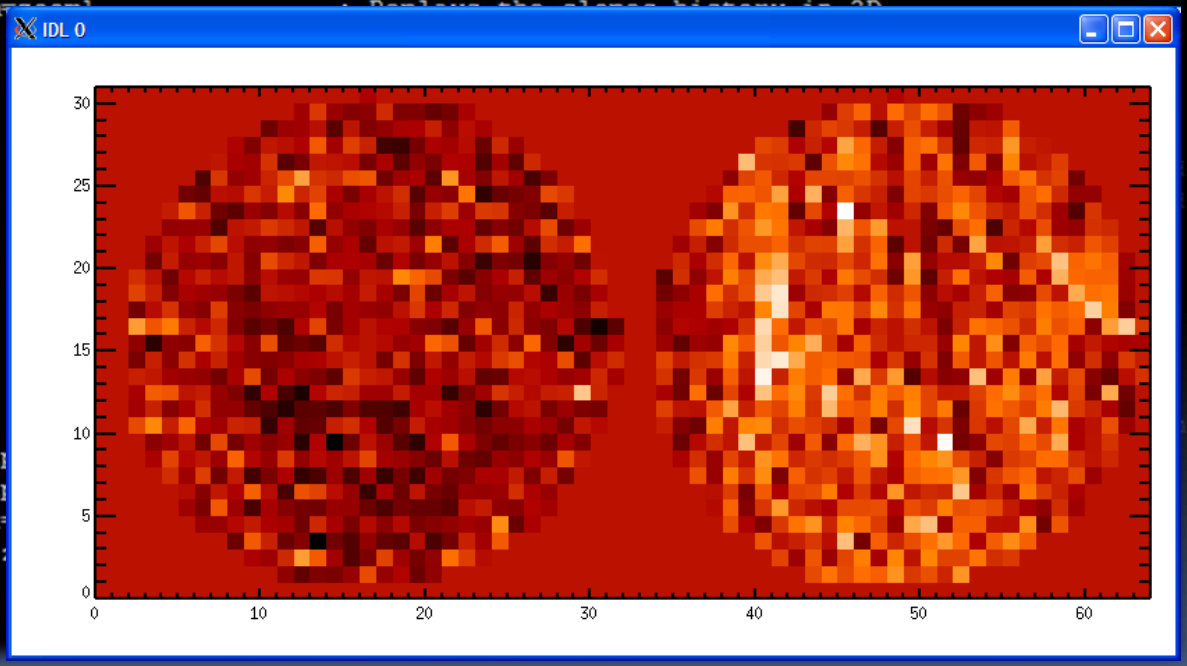
TrackNo	RefStar	Mag	El	Wind	Rec	bin	#mod	freq	gain	mod	nph	SR	band	exp	#frames	notes
20100622_032730	Wref93	7.7	85	1	185716	1	400	1000	0.7 1.2	2	486	66	H	5	1000	
20100622_032903	Wref93	7.7	85	1	185716	1	400	1000	0.7 1.2	2	482	57	H	5	1000	
20100622_033037	Wref93	7.7	85	1	185716	1	400	1000	0.7 1.2	2	476	58	H	5	1000	
20100622_033454	Wref93	7.7	85	1	185716	1	400	1000	0.7 1.3	2	482	58	H	5	1000	
20100622_034419	Wref93	7.7	85	1	185716	1	400	1000	0.7 1.1	2	491	61	H	5	1000	
20100622_034558	Wref93	7.7	84	1	185716	1	400	1000	0.7 1.0	2	487	78	H	5	1000	
20100622_034737	Wref93	7.7	84	1	185716	1	400	1000	0.7 1.0	2	485	72	H	5	1000	
20100622_034912	Wref93	7.7	84	1	185716	1	400	1000	0.7 1.0	2	489	78	H	5	1000	
20100622_035346	Wref93	7.7	84	1	185716	1	400	1000	0.7 1.0	2	481	34	J	20	1000	
20100622_035537	Wref93	7.7	83	0	185716	1	400	1000	0.7 1.0	2	485	37	J	20	1000	
20100622_035718	Wref93	7.7	83	1	185716	1	400	1000	0.7 1.0	2	487	38	J	20	1000	

Pausa per LBC poi di nuovo Wref93

TrackNo	RefStar	Mag	El	Wind	Rec	bin	#mod	freq	gain	mod	nph	SR	band	exp	#frames	notes
20100622_053039	???	7.7	65	2	185716	1	400	1000	0.7 1.1	2	508	82	H	5	1000	
20100622_053226	???	7.7	65	2	185716	1	400	1000	0.7 1.0	2	502	73	H	5	1000	

It also has an online help!

```
AO>
AO> a->help, 'slopes'
ee->slopes()                                                    : reference to slopes object (AOslopes)
ee->slopesnull()                                                : reference to a slopesnull object (AOslopes)
(ee->wfs_status())->slopes_null_fname()                         : slopesnull vector fitsfile name (string)
(ee->slopes())->fname()                                         : fitsfile name (string)
(ee->slopes())->header()                                        : header of fitsfile (strarr)
(ee->slopes())->slopes()                                        : return slopes matrix [nslopes x niter]
(ee->slopes())->nslopes()                                       : return number of slopes
(ee->slopes())->niter()                                         : return number of iteration (eventually after reform
(ee->slopes())->sx( [subap_idx=subap_idx, iter_idx=iter_idx]: return x-slopes [nsubaps x niter].
(ee->slopes())->sy( [subap_idx=subap_idx, iter_idx=iter_idx]: return y-slopes [nsubaps x niter].
(ee->slopes())->slopes2d( [iter_idx=iter_idx] )                 : return cube with remapped slopes in 2D.
(ee->slopes())->replay[,wait=wait ,zoom=zoom]                   : Replays the slopes history in 2D.
(ee->modal_rec())->nslopes()
(ee->modal_rec())->slopes_idx()
(ee->intmat())->sx([idx])
(ee->intmat())->sy([idx])
(ee->intmat())->sx2d([idx])
(ee->intmat())->sy2d([idx])
(ee->intmat())->nslopes()
(ee->intmat())->slopes_idx()
(ee->slopesnull())->fname()
(ee->slopesnull())->header()
(ee->slopesnull())->slopes()
(ee->slopesnull())->nslopes()
(ee->slopesnull())->niter()
(ee->slopesnull())->sx( [subap_idx=subap
(ee->slopesnull())->sy( [subap_idx=subap
(ee->slopesnull())->slopes2d( [iter_idx=
(ee->slopesnull())->replay[,wait=wait ,z
AO> s2d = (a->slopes())->slopes2d()
AO> image_show, /as, s2d[*,*,0]
AO>
```



Database

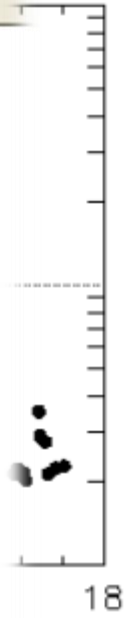
- ELAB implements also a simple database for statistical analysis
  - Contains every measurements (>12000 for FLAO1 commissioning)
  - Query on main AO system parameters and performance estimators
  - Can be easily rebuilt, recomputing all the values (yes, sometime we find bugs!)
  - Implemented in IDL in a single .sav file

Database use case: SR vs Mag

analyse_onsky_flao1.pro + (~\Desktop) - GVIM

File Edit Tools Syntax Buffers Window Help

```
;Load database
;----------------------------------------
db = getdb()
setstar = db_good_for_onsky_sr()
setnotignore = db->query('ignore','eq',0, sub=setstar)

setcl       = db->query('control.mingain', 'gt', 0.05, sub = setnotignore)
setnoKalman = db->query('control.iskalman', 'eq', 0, sub=setcl)
setsr       = db->query('irtc.sr_se', 'in', [.01,1.], sub=setnoKalman)
setH        = db->query('irtc.lambda', 'eq', 1.6e-06, sub=setsr)
set10m      = db->query('irtc.pixelscale', 'in', [0.009, 0.011], sub=setH)
setDarkOld  = db->query('errorDescription', 'like', '*IRTC dark*old*', sub=set10m)
set10m->removeTracknum, setDarkOld->tracknums()

;Remove bin2 buggy rec (one subap more)
buggy_rec  = 'adsec_calib/M2C/KL_v2/RECs/Rec_20100528_043207.fits'
setselect1 = db->query('modal_rec.fname', 'ne', buggy_rec, sub=set10m)

;Valid seeing data:
setselect2 = db->query('olmodes.seeing', 'in', [0.1,3.], sub=setselect1)

;Remove low-gain suboptimal perf:
set_temp = db->query('tracknum', 'in', ['20101126_051600', '20101126_065450'], sub=setselect2)
set_temp = db->query('control.mingain', 'lt', 0.7, sub=set_temp)
setselect2->removeTracknum, set_temp->tracknums()

;Remove low-number of controlled modes:
setbin1  = db->query('wfs_status.ccd39.binning','eq', 1, sub=setselect2)
setbin1a = db->query('modal_rec.nmodes', 'lt', 390, sub=setbin1)
setselect2->removeTracknum, setbin1a->tracknums()

;Remove other particular tracknums:
setselect2->removeTracknum, '20100620_085443'   ;point at mag=16 with 40% SR !!!

;Select those where DIMM seeing exists, bin=3, mag<12.7, elevation>70:
setselect4  = db->query('tel.dimm_seeing', 'in', [0.1,4.], sub=setselect2)
setselect4b = db->query('wfs_status.ccd39.binning', 'lt', 3, sub=setselect4)
setselect4c = db->query('mag', 'lt', 12.7, sub=setselect4b)
setselect4d = db->query('tel.el', 'gt', 70., sub=setselect4c)

-- INSERT --
```



nd versus star magnitude
points for which seeing
ormance from numerical

Conclusions

- Data processing is a fundamental, overlooked tool to test an AO system
- Soon or later, you will wander about every single parameter of the system and you will want to know how you produced that single matrix. So start now organizing the data collection.