LBT PROJECT

2 X 8,4m OPTICAL TELESCOPE

LBT672 optical calibration procedure

	Signature	Date
Prepared	R. Briguglio	19 Jan 2016
Issued		
Reviewed		
Approved		

INAF – Osservatorio Astrofisico di Arcetri - Largo E. Fermi, 5 - 50125 Firenze - ITALY
LBT Observatory – 933 N. Cherry Ave – 85721 Tucson, AZ - USA

Modification Record

Issue	Date	Responsible	Section affected	Reason/Remarks
A-draft02	24 Nov 2014	R. Briguglio		First release delivered to LBTO (G. Brusa) on 21/May/2015
A-draft03	19 Jan 2016	R. Briguglio		Review version for procedure handover
A-draft06	16 Feb 2016	R. Briguglio		Creation of AO control basis added

Contents

1 Abbreviations, acronyms and symbols

Symbol	Description
ASM	Adaptive Secondary Mirror
FF	Feed Forward
HP	Hexapod
LBT	Large Binocular Telescope
RR	Retro Reflector
RRH	Retro reflector holder
TS	Thin Shell

2 Purpose of the document

The document describes the optical calibration procedure for the ASM, requested to provide the AO subsystem with a flattening command for seeing limited observations and the modal basis for close loop operations.

3 Applicable and Reference Documents

3.1 Applicable Documents

[AD1] NA

3.2 Reference Documents

[RD1] NA

4 Executive Summary

In the present document the optical calibration of the LBT672 adaptive secondaries is described. The procedure is requested to compensate for the poor actuator calibration, responsible for a distorted mirror shape. The procedure will provide the ASM mirrors with the flat shape requested for seeing limited operations and with the AO control basis requested for the AO reconstructor. The procedure shall be run after a major ASM refurbishment (e.g. magnets, actuators or boards replacement), TS swap…

The procedure described includes the measurement of the HP interaction matrix, to allow automatic ASM alignment, the measurement of the stiffness modes optical data (influence functions or mirror modes), the preparation of the mirror-interferometer interaction matrix, the mirror flattening and the preparation of the AO modal basis.

The procedure is composed by a sequence of fully automated operations, mostly in the form of IDL scripts; the procedure shall be executed by an operator skilled in optical alignment and ASM operations.

5 General overview of the procedure

The optical calibration procedure is a sequence of operations requested to prepare the adaptive secondary mirrors, after the set of EM verifications, for AO operations. The procedure is semi-automated, i.e. it is a sequence of atomic tasks where the operator is in charge of the data inspection to proceed with the following step.

5.1 Procedure steps

The main blocks of the calibration procedure are:

1. The optical alignment and the preliminary mirror flattening;
2. The measurement of the HP interaction matrix;
3. The measurement of the IF to prepare the mirror-interferometer optical interaction matrix;
4. The mirror flattening;
5. The definition of the AO control basis.

In the following, we will not discuss the optical alignment of the ASM, involving the interferometer, the tertiary mirror and the initial ASM positioning by mean of the HP. The automated HP compensation of drift will be described.

The calibration procedure is iterative: the data quality is strongly affected by the initial mirror figuring error, which in turn is improved by running the flattening process. Therefore, the steps #3 and #4 are repeated as long as the data quality and the figuring are improved; each iteration is performed increasing the number of the mirror modes which are sampled in step #3 and corrected in step #4.

5.2 Key tools

The procedure involves the ASM, the 4D interferometer and the ASM workstation running the IDL environment. The optical data are collected via the 4D interferometer, which is installed on the telescope platform on a dedicated focal station; the interferometer is housed in an thermalized insulated box to prevent damages from cold temperature; the workstation is located in the lower-right tree-house and cable-connected to the interferometer; a trigger line is routed from the workstation to the ASM to synchronize the data collection with the application of mirror commands. In particular the trigger line is an optical fiber running from the ASM cabinet (left or right) to the upper-right tree-house; it is then routed from the tree-house to the lower one, where the optical signal is converted into a TTL by a converter box, then input to the 4D workstation.

The 4D is controlled by its proprietary SW, accessible in VNC from any workstation within the LBTO. The interferometer image is viewed by such SW. The data are collected through IDL as described below.

5.3 Operator tasks

The operator is in charge of checking the data integrity and quality and of tuning the procedure parameters beyond the defaults which are already defined. In particular, the interferometer images should be verified to avoid reconstruction errors, caused by too large fringes density, and to define the interferometer mask accordingly during the initial calibration steps. As soon as the flattening process is converging, i.e. the mirror visible area is enlarging and the residual wavefront error is decreasing, the operator will enlarge the interferometer detector mask and restart the calibration process under such new conditions.

The detector mask is designed as in **Error! Reference source not found.**; the list of relevant parameters and their selection guide is given in Table 5.

5.4 Risks description

5.4.1 ASM

No damage can result on the ASM with the application of wrong actuator commands (for instance, exceeding the force threshold); both the internal metrology and the supervising SW are in charge of checking the actuator forces and positions and reject bad commands. In case of failure, the TS is safely set into RIP state.

5.4.2 HP

The HP movements envelope is collision-free by design, i.e. the ASM cannot touch its enclosure (hub) for any HP position. However, in order to preserve some further correction range (for instance for the AO offloading), the HP position for the ASM alignment should be close to its mid range. The operator should check the HP position and the alignment command during the procedure to ensure the HP is not placed close to its limit positions.

5.4.3 Interferometer

No specific failure or damage can happen to the interferometer running the procedure. Please remember that the interferometer head is contained within an insulated, thermalized box to keep the CCD camera temperature at 20°C, in order to avoid damages caused by cold temperature to the pixelated mask placed in front of the detector. The current temperature may be read from the box control panel located in the lower-right tree-house.

5.4.4 Laser exposure

A 1 mW laser is output from the interferometer head located inside its insulated box. The laser source is powered on/off with the interferometer control box installed in the lower-right tree-house.

!!! The laser key does not power the laser source on/off.

The operator should pay attention to avoid direct eye exposure to the beam while installing the F/15 optics and during mounting/dismounting the RRH.

While mounting/dismounting the RRH the laser should be powered off.

5.5 Data organization

The calibration procedure creates a large amount of datasets, both as direct measurements and as post-processing results. Data are saved in the adsecdx (sx) folder $ADOPT_MEAS/adsec_calib/, containing the following subfolders:

Name	Content	Name	Content
4d/	interferometer .h5 images, organized in folders /TRACKNUM/hdf5/	if_function	IF dataset
flat/	flattening result realizations and optical interaction matrices	hexapod/	hexapod-interferometer interaction matrix data
disturbance/	mirror command sequences		

Table 1 List and description of data folders.

Within each folder, data are organized in sub-folders named according to the convention of the tracking number, which is a 15 characters date label in the form: yyyymmdd_hhmmss, e.g. 20100203_040000. When a brand new dataset is created, the relative tracking number is created as well; as soon as a dataset is processed to obtain a new, elaborated dataset, it is saved in the destination folder, named with the same parent tracking number label.

6 System startup and preparation

6.1 Pre-requisites for optical calibration

In order to start the optical calibration procedure, the telescope shall be set according the following requirements:

System	Status
Dome	Preferred: parked in order to have the calibration side (L or R) in South-West direction (to minimize solar heating on the walls). No doors open. No vibration sources on the floor. No matter the light status.

Telescope	Zenith pointing, HBS on, PIN off No people on the platform, RRH installed, RRH optics cap removed.
ASM	EM verification ok, FF calibrated,HP aligned (see procedure)
M3	working and preliminarily set for interferometer focal station
4D	power box ON, laser ON, workstation ON and accessible via VNC, 100Gb free on the HD, F/15 lens installed, static IP, remote controller (handpad) installed on the 4D electronics box.
AOS	All processes are up and running. The Pyro_Client.conf file uodated with the correct 4D IP (edit $ADOPT_SOURCE/conf/adsec/672XX/processConf/Pyro_Client.conf, then install the new configuration).

6.2 Operator workspaces

The operator is supposed to work from a terminal in order to open communication shells to the ASM workstation and to the 4D. In the following we will describe the connections.

To get connected to the ASM workstation:
ssh –X AOeng@adsecdx (passwd=m1rr0r) for DX = RIGHT side= LBT672A

or

ssh –X AOeng@adsecsx (passwd=m1rr0r) for SX = LEFT side= LBT672B

To get connected to the 4D workstation:
Vncviewer 10.144.0.92 (usr =PhaseCam; passwd = 4D)

Once logged in, commands can be sent:

- from the adsecdx(sx):
  - from the linux prompt (system commands),
  - from the *IDLController* process shell,
  - from a IDL shell communicating with the ASM, or
  - from an IDL service shell, initialized with the proper variables;
- from the 4D workstation:
  - commands are given from the proprietary 4D software GUI or
  - from an embedded Python console for scripting.

In Table 2 is indicated how to access these interfaces and the associated prompt, used as a representation adopted throughout this document.

Prompt symbol	**Environment**
$	*Linux shell*
AdOpt>	*IDL shell of the IDLController process.* It is accessible from “IDL Terminal” button on the “ADSEC INTERFACE”

	GUI, or running the “terminal” command from the Linux prompt. This shell cannot be used for graphics by default. For graphics use the Service IDL shell. For the flattening procedure, when some graphics is requested , use the following command: AdOpt> set_plot “X” test the graphics with a dummy command like: AdOpt> plot, indgen(100) if no graphics is displayed, rest the shell: AdOpt> print, fsm_rip() stop and restart the IDL process, and reload the DSP program: AdOpt> print, fsm_load_program(/AUTO) and try again with the set_plot command. The graphics is shown in the screen where the IDL process has been launched from. At the end, restore the set_plot setting with the command: AdOpt> set_plot, “Z”
IDL>	*aoidl* IDL shell in communication with the MsgDB IDL > @startup-onlycom
IDL>	*aoidl* Service IDL shell. Process not communicating with the MsgDB. Used for data visualization and post-processing without keeping busy the IDL Controller shell. For the IDL session initial configuration run as first the command: IDL> @startup
>>>	Python console on 4D
4Sight:	The label represents the 4D proprietary SW to manage and visualize the interferometer images. The SW is named 4Sight and can be opened by clicking the 4Sight1.8 icon on the desktop. 4Sight > File: indicates to access the “File” menu from the SW 4Sight > Measurement Console: indicates to access the “Measurement” menu from the SW

Table 2

6.3 4D startup

The panel of the 4Sight program is shown in Figure 1, where the main buttons are shown.

Commands and actions	Notes
From File Manager verify that 100Gb disk space is available on D:/	
From the File Manager verify that a Z:/ disk (ram disk) is installed and that contains only a 4D/ folder. Verify that at	if the RAMdisk is not available, refer to **Error! Reference source not found.**

Deleted: Table 4

least 1Gb is available in Z:/	
Start 4Sight1.8 (icon on desktop)	
4Sight> Tools>**Select Default Script** 4Sight> Tools>**Run default script**	File to select: C:\4D\scripts\ServerStartup.py

6.3.1 Installation of the RAM disk on the 4D workstation

The RAM disk is used to allow a speed-efficient frame transfer from the frame grabber to the HD. If the RAM disk Z is not present, it may be created with the following procedure.

Command or action	**Notes**
On the 4D workstation desktop, double click on the icon “RAM DISK”	
Select 1Gb as disk size	
Select RAM drive	
Select Z as disk label	
Press “Done”	Verify with File Manager that the RAM disk has been successfully created
Create the folder Z:/4D	

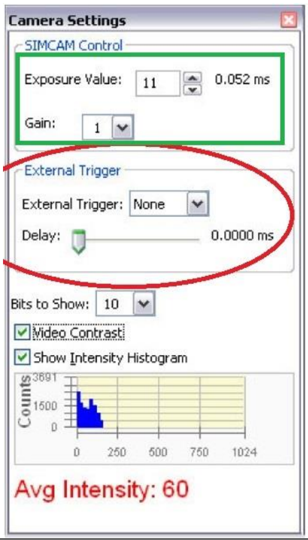
Table 3 Creation of the RAM disk

Deleted: 4

6.4 4D sampling parameters optimization

As soon as you have fringes of the ASM on the interferometer screen you are able to tweak the sampling parameters, in order to optimize the measurement. It is recommended to iterate such optimization as soon as the image quality (i.e. fringes density) is improved during the flattening process.

4Sight> Measurements (#5 in Figure 1)> Camera Settings> Exposure and Gain	Adjust the values to obtain a suitable illumination in the image. Check the intensity histogram and the image for under or over-exposed image areas (over-exposed areas are drawn in red). !! exposure time larger than 0.1ms are NOT suitable. If a large exposure time is needed, please contact Arcetri or 4D for further information, as it is a sign of

	interferometer malfunctioning. 
4Sight> Processing Options (#6 in Figure 1, **Error! Reference source not found.**) **Detector Mask**	Draw the detector mask in order to reject the hidden areas in the image. For the detailed instructions, refer to **Error! Reference source not found.** Save the mask as tracknum_mask
4Sight> Measurements > Camera Settings > External trigger	Set external trigger to “None” (the trigger line is automatically selected) and the delay to 5ms.
4Sight> Processing Options > Modulation threshold	Refer to 6.4.2
4Sight> Menu> Save Configuration File	save file as D:4D/Data/configuration/tracknum.ini
4Sight> Menu> Load Configuration file	Load the freshly saved configuration file
open the file C:/Obelix_CVS/Supervisor/PyModules/I4D/Constant4d.py Edit the ConfigurationFile string to match the configuration file name	
4Sight> Tools>Debug>Open Scripting Console	a Python console pops-up
>>> from I4D import comm4d from I4D import Constant4d	
>>> reload(Constant4d) >>> reload(comm4d)	All the requested Python variables are restored.

Deleted: Figure 3

Deleted: 6.3.2

Deleted: 6.5.1

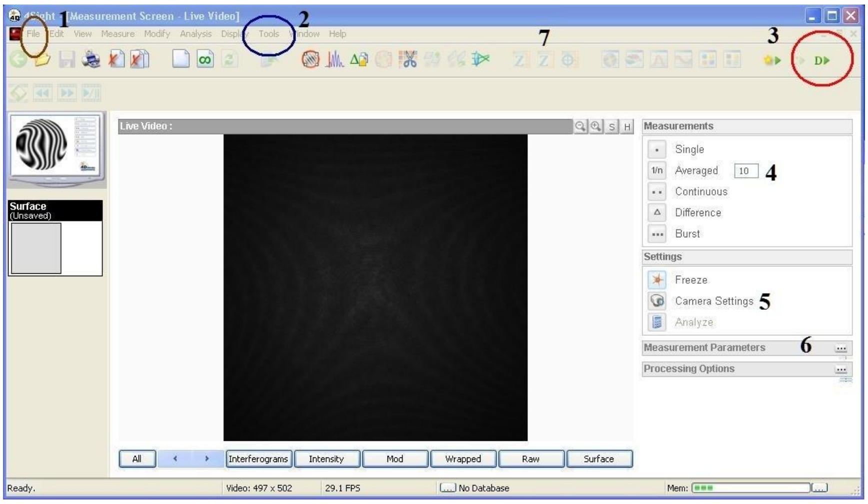


Figure 1 4Sight panel

6.4.1 Detector mask definition

The detector mask panel in the 4Sight GUI is opened by accessing the “Processing options” (#6 in Figure 1). A mask is designed by selecting the desired shape (circular, for instance), placing it on the live view image and tailoring its dimension and position. For the best definition of the detector mask, the operator could consider the following tips:

- Points outside the optical pupil must be masked;
- Areas with large fringes density must be masked;
- At the beginning of the flattening process, it is preferable to have a smaller detector mask in order to improve the definition of the measured mirror modes.
- During the process, it is preferable to enlarge the detector area to include (at least partially) the highly warped regions: this in order to allow the flattening procedure to partially correct them.
- To automatically reject border pixels, use the “Trim pixels” option (in “Processing options”)
- Select “Data Fill” option (in “Processing options”): it is useful to fill single

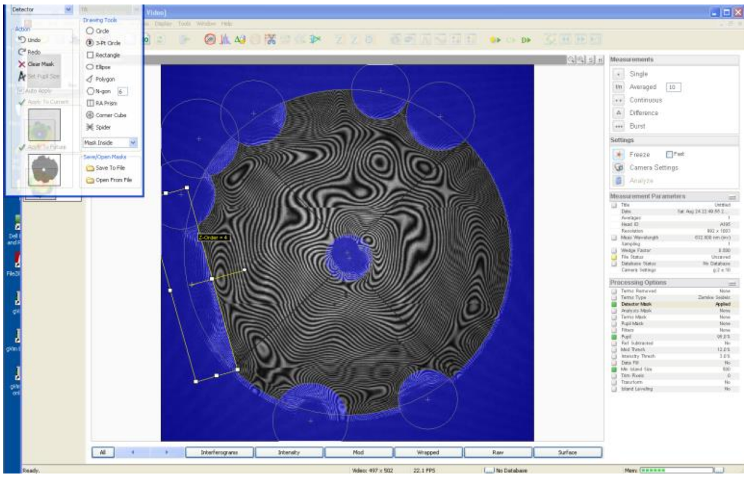


Figure 2 The “Detector Mask” panel options

Deleted: 3

6.4.2 Modulation threshold setting

The modulation threshold panel may be opened from the “Processing options” menu after an image is acquired. The modulation thresholding is a way to automatically mask out those image areas where the interference signal is low: for instance, the area outside the optical pupil and the noisy regions. Those areas must be rejected to avoid a wrong phasemap reconstruction and artifacts in the image. This is done by selecting a minimum modulation value, and rejecting those pixels where the signal is lower. The procedure for selecting the modulation threshold is shown below.

Command or action	Notes
take a frame in continuous mode	
4Sight>Processing Options> Modulation threshold	
Look at the modulation histogram (refer to Figure 3)	

Deleted: Figure 2

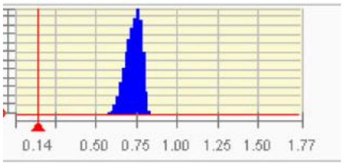
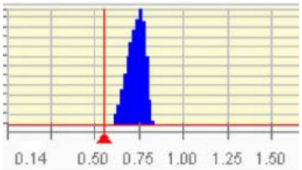
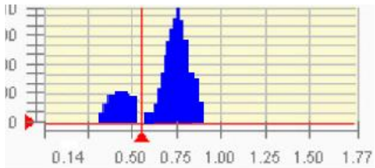
Place the vertical sign BEFORE the histogram, as in the picture: choice #1 (low threshold) will exclude less pixels and is more suitable at the beginning of the procedure; choice #2 is more correct as it will exclude pixels with insufficient modulation (at the cost of being less robust).	1:  or 2: 
Click the "Apply changes" and copy to future" buttons	
check the image as soon as it is updated	Check for the presence of artifacts or missing areas.
Iterate the sequence: move the threshold → apply changes; checking the effect in the captured image.	Stop the process when the best results are obtained
Take a new image and verify the results are stable	
In the case of more complex histogram shapes, try first the mentioned procedure discarding the left-most structures.	

Table 4 Modulation threshold definition procedure

Deleted: 3

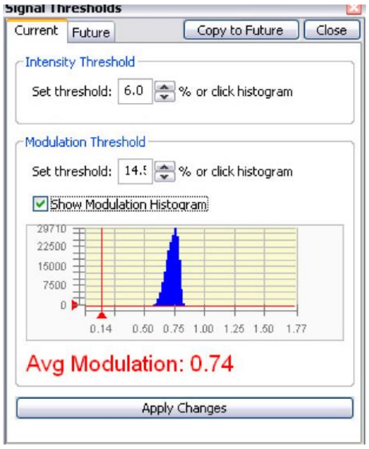


Figure 3 Modulation threshold panel.

6.5 ASM startup

The ASM is started as usual.

Commands and actions	Notes
At the very first step of the calibration procedure, the mirror is commanded with a force-relaxing command, with the following command: AdOpt> print, fsm_set_flat(/NO_FLAT)	
as soon as a new (even preliminary) flattening command is obtained, it can be applied with: AdOpt>restore, 'flat_name.sav' AdOpt> print, fsm_load_shape(flattened_status)	Select the correct flat file name

6.6 Test of trigger signal

The test is required to verify the trigger line in terms of connections and hardware functionality.

Commands and actions	**Notes**
Verify that the trigger interface box is correctly connected and powered up. See Figure 5.	the trigger interface box is located in the lower-right tree-house and is powered by a 5V power supply. It shall be connected to a fiber cable on one side and to a BNC cable to the interferometer box to the other side.
preliminary check: connect a voltmeter in “Hz” mode to the BNC output	use a BNC-banana connector to correctly connect the box and the voltmeter
AdOpt> print, i4d_opt_intmat(0,200, /sim)	a sequence of trigger pulses should be output from box
verify from the voltmeter that the gauge is stably reading 16 Hz	the test checked the integrity of the trigger line
AdOpt> print, fsm_load_program(/auto) AdOpt> print, fsm_set_flat(/no_flat)	
AdOpt> print, i4d_opt_intmat(0,200)	
Open the file D:/4D/Log/date/log.txt and check for the registered frame rate	Refer to Figure 4

Figure 4 The 4D acquisition log, indicating that the sampling has been performed at 25.0Hz

Figure 5 The trigger interface module installed in the lower-right tree-house.

7 Preliminary flattening

The preliminary flattening procedure consists in the open loop correction of the first low order mirror modes or any arbitrary shape, commanded manually by the operator. The selection of mirror modes and their sign and amplitude is found by considering that the first mirror modes may be roughly approximated by the first Zernike modes: here, we will consider astigmatisms and trefoils.

Commands and actions	Notes
4Sight> Processing options > Detector mask	Create a preliminary mask, considering the central part of the mirror where the fringes signal is good.
4Sight> Measurement (#4 in Figure 1) > Continous mode	
4Sight> Zernike worksheet (#7 in Figure 1) 15 terms	check the Zernike modes and their amplitude; consider astigmatisms and trefoils.
AdOpt> V = adsec.ff_p_svec	the modes matrix is considered
AdOpt> display, /smooth, v[I,*] (i= 0, 1, 2,…)	check the image and identify the modes similar to astigmatism (a) and trefoil (b)
AdOpt> print, fsm_apply_cmd(/delta,/passed, 1e-	the first “astigmatism like” mode is applied.

6*V[a,*])	Check in the Zernike table the variation of the associated amplitude.
4Sight> Processing Options (#6 in Figure 1, **Error! Reference source not found.**) **Detector Mask** adjust the mask	if the visible area is larger, adjust the detector mask accordingly and save the configuration file.
iterate the 2 steps above to lower the amplitude of the astigmatisms and trefoils.	
To apply arbitrary shapes, create the wanted shape *s* as a vector of 672 actuator commands, then apply it with: AdOpt> print, fsm_apply_cmd(/delta,/passed, s)	
at the end, check the Power amplitude and correct it with the hexapod	Use the OSS hexapod GUI to find the best HP Z position to minimize power.

Deleted: Figure 3

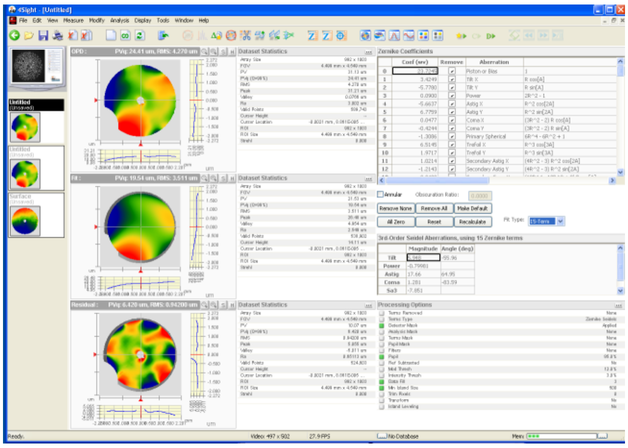


Figure 6 Zernike worksheet

8 Automated HP correction

The procedure shall be started when the full ASM pupil is visible and the finges density is low enough to allow the measurement of large amplitude tilt or coma.

8.1 Measurement of HP-interferometer interaction matrix

Formatted: Heading 2

Commands and actions	Notes
Align the HP	

$ aoidl IDL> @startup-onlycom	open an IDL session communicationg with the MsgRTDB.
IDL> print, hexapod_opt_calib(/getpos)	the hexapod will be moved to produce tip, tilt, focus and coma misalignment. An image is taken at each movement. **!!check that the images are correctly reconstructed.**
$ ls $ADOPT_MEAS/adsec_calib/hexapod/	check that the dataset has been correctly saved as TRACKNUM/hexapod_tf_mat.sav

8.2 Hexapod alignment

$ aoidl IDL> @startup-onlycom	open an IDL session communicationg with the MsgRTDB.
IDL> print, hexalign() !! no movement will be performed.	!!check that the wanted movement is safely within HP allowed range.
IDL> print, hexalign(/apply)	

9 Sampling of the influence function data

The interferometer and the ASM workstation must be initialized as described in 6.3 and 0

Deleted: 6.3.3

Commands and actions	Notes
4Sight> Processing Options (#6 in Figure 1, **Error! Reference source not found.**) **Detector Mask** adjust the mask	
4Sight> File > Save Configuration File	The latest mask configuration is saved and used
open the file C:/Obelix_CVS/Supervisor/PyModules/I4D/Constant4d.py Edit the ConfigurationFile string to match the configuration file name	
>>> reload(Constant4d) >>> reload(comm4d)	
AdOpt> print, i4d_opt_intmat(start,end,AMPRMS=a, aver=n) Tracking number is 20100101_020202	record the TRACKNUM from the console start, end, a, n are discussed in Table 5

Deleted: Figure 3

$ tail –f $ADOPT_MEAS/adsec_calib/if4d.txt	check the file, it should log: Tracknum = TRACKNUM Modes ## start end Modes Amp a Num +- applic n ……
>>> comm4D.produce('20100101_020202')	the raw frames dataset is processed and a folder of .h5 files is created in: 4d/TRACKNUM/hdf5/

Referring to the command:
AdOpt> print, i4d_opt_intmat(start,end,AMPRMS=a, aver=n)
We will discuss in the following table the parameters and their acceptable values

Parameter	Value	Description
start, end	0 →# actuators-1	the first and last mode to be sampled. The sampling is grouped in modes subset, typically 0:249, 250:499, 500:671 **!!it is important to split correctly a modes sequence into subsets, without repetitions.**
A	1e-9 →5e-6 [m]	amplitude RMS surface for the command application
n	3 →51	# of command repetitions as a sequence of push-pull: 3 = 1push, 1pull, 1push

Table 5 IF sampling parameters

10 Preparation of the interaction matrix

So far, a group of interferometer images have been collected, each of them designated with an individual tracking number: Tr1, Tr2, Tr3. The interaction matrix requested for flattening is assembled by:

1. Compute the mode optical shape (IF) from the sequence of push-pull;
2. Normalize the images to the surface amplitude applied;
3. Pile up the modal images with the associated mask.

The sequence of operations is described below; no specific initialization is foreseen for the ASM, ASM workstation and 4D workstation.

Commands and actions	Notes
$ less $ADOPT_MEAS/adsec_calib/if4d.txt	check in the IF acquisition logfile
identify the tracking number sampled and	note the wanted tracking number: Tr1, Tr2, Tr3

processed by the 4D *produce* script	
$ aoidl IDL> @startup	Start an IDL process without communication with MsgDB. Only for data processing
IDL> print, if_redux(/MODAL, Tr1)	
IDL> print, if_redux(/MODAL, Tr2) …..	
IDL> print, build_int_mat([Tr1, Tr2, Tr3])	the tracking numbers list is passed as a string vector.
$ ls $ADOPT_MEAS/adsec_calib/flat/Tr1/	verify that the file flat_ data_'n modes'.sav (containing the interaction matrix) has been correctly saved. Refer to such file for the interaction matrix assembled with the datasets in Tr1, Tr2, Tr3

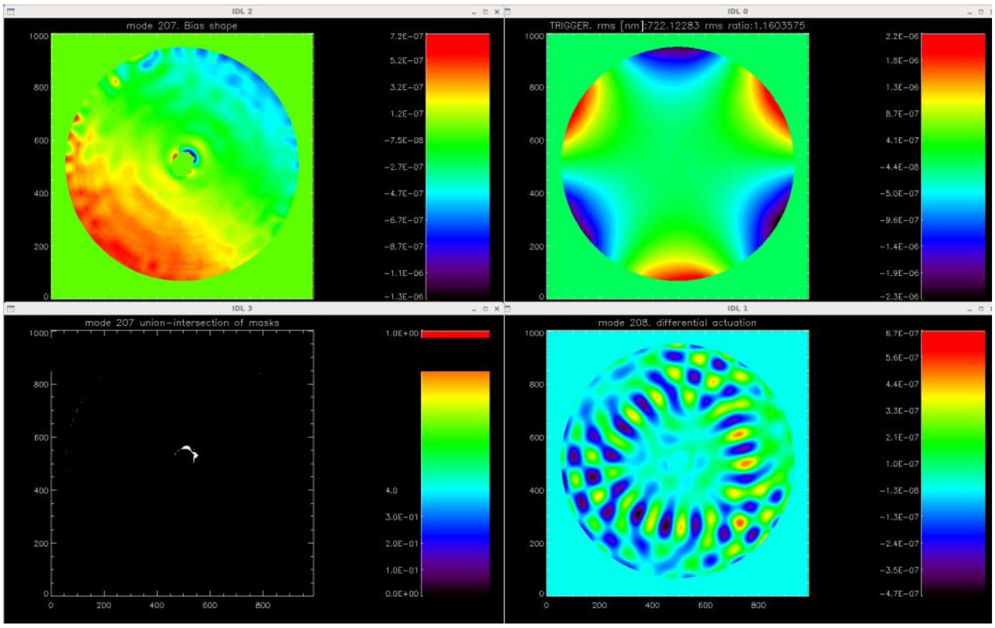


Figure 7 if_redux script graphical output. Top-left: bias image; top-right: initial trefoil shape applied as a trigger signal; bottom-right: the reconstructed mode, computed as the average of the push-pull differences; bottom-left: intersection mask.

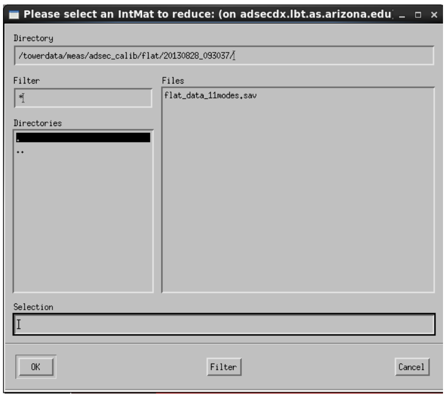
Note:

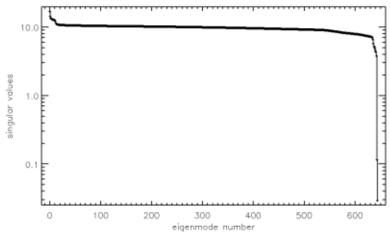
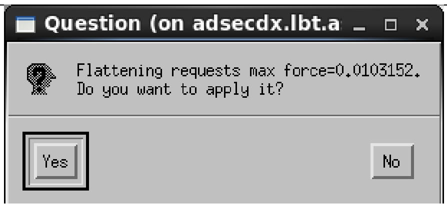
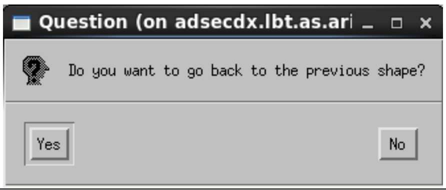
If an IF dataset is replicated (for instance, Tr4 contains the same mode images as Tr3), you may pile up a new interaction matrix using Tr1, Tr2, Tr4. Because of the tracking number convention, the interaction matrix file will be save in the same folder Tr1 (as in the example above), the new file will be labeled with "_TRACKNUM" (TRACKNUM = current date).

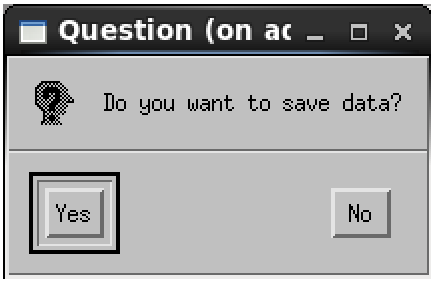
11 Flattening procedure

So far, we measured n modes and built the optical interaction matrix relative to those modes. The interaction matrix is saved in the tracking number Tr1. The ASM may now be flattened using the sampled modes as described below.

The interferometer and the ASM workstation must be initialized as described in 6.3 and 0

Commands and actions	Notes
$ aoidl IDL> @startup-onlycom	Start (if not already previously started) an IDL process enabling communication with MsgDB
IDL> print, opt_int_mat_reduce2(n_modes,track=Tr1, avetime=t)	flat_data_"nmodes".sav
select the wanted int_mat file in the pup-up window. As different interaction matrices are saved in the same folder (those sharing the same mode #0), select the one with the desired number of modes.	
Chech the singular values plot, where the last 3 modes are discarded.	

	
in the dialog box, check the maximum force requested	 the max applicable force is 0.7N. However, a typical flattening realization will not request a peak force larger than 0.3N. If a larger force is required, this may be a symptom of errors in the procedure (bad pixel, tracknum mismatch, poor SNR in the data…)
4Sight>Measurement>Single	take a reference image with the interferometer
4Sight>Processing Options> Removed terms	remove piston, tip/tilt, coma and power
4Sight>RMS	take note of the initial value of the wavefront error RMS
press YES in the dialog window: “do you want to apply the command?”	the command is applied
4Sight>Measurement>Single	take a result image with the interferometer
4Sight>Processing Options> Removed terms	remove piston, tip/tilt, coma and power
4Sight>RMS	take note of the final value of the wavefront error RMS. Check if the final value is lower than the initial one
press NO in the dialog window: “do you want to go back to the previous shape?”	

press YES in the dialog window: “do you want to save data?”	

12 Save the flat file as default

$ idl IDL > restore, flat_nameXXX,/ver	Open an IDL session and restore a previously saved flattening file, e.g. /flat/XXXX/flattening_result.sav
IDL > pos_cube=’undefined item’ IDL > cur_cube=’undefined item’	empty position and force dataset: a shortcut to create readable flat file.
IDL > save,file=’$ADOPT_MEAS/ adsec_calib/flat/Applicable/default.sav’, flattened_status, pos_cube, cur_cube	If necessary, make sure to have a backup copy of the existing /flat/Apllicable/default.sav file

13 Create the offload projection matrices

The projection matrices are requested to compute the offload commands. Such matrices are derived from the optical interaction matrix.

$aoidl IDL > @startup	Start an IDL session with the AdSec configurations.
IDL > im2mmmatrix, tn, rebfactor=4	tn= the tracking number of the interaction matrix (corresponding to the tracknum of mode #0); rebfactor is the rebin coefficient. The routine will save a rebinned dataset of the interaction matrix in the folder $ADOPT_MEAS/optical_projection/’tracknum’/mmmatrix.sav
Define the list of actuator whose IF are	This is requested to avoid fitting the image geometry with

poorly visible (e.g.: actuator on the inner/outer edge). The list is ‘badact’	IF data where the peak (i.e. the magnet location) is poorly visible.
IDL> make_ make_offload_matrix_proj, tn, nzernike=22, cl_act=adsec.act_w_cl, act_nomap=badact,/save, tracknum=tn	Data are saved in $ADOPT_MEAS/adsec_calib/optical-projection/tn/ Pmz.fits and pos2modes.fits are saved
Copy the files: Pmz.fits and pos2modes.fits in $ADOPT_SOURCE/calib/adsec/672XXX/data.	
$ cd $ADOPT_SOURCE $ make install-calib	

Deleted: ¶
¶

14 Definition of the AO control basis

The main steps required to define an AO control basis are:

1) *Produce a preliminary IF matrix*. The acquired influence function measurements are processed (i.e. normalized, binned, filtered, …) and the valid OPD points are extracted and rearranged in a matrix form. The details of the procedure are discussed in Section 14.1.

Deleted: 12.1

2) *Tweak the pupil mask*. In general, the mask of “valid points” calibrated in the previous step needs to be adjusted in order to minimize fitting errors. The procedures to identify and remove the “bad” pixels are discussed in Section 14.2.

Deleted: 12.2

3) *(Optional) Slave a selected subset of actuators* (Section 14.3).

Deleted: 12.3

4) *Produce the final IF matrix, and compute its pseudo-inverse*. This “final” IF matrix contains the information of the slaved actuators (if any), and it is defined in the tweaked pupil mask. Its pseudo-inverse is the matrix required to fit the KL modes (Section 14.5).

Deleted: 12.5

5) Compute a theoretical set of KL modes that is orthonormal in the selected pupil mask, and fit it with the final IF matrix produced in the step above (Section 14.5).

Deleted: 12.5

14.1 Preliminary optical interaction matrix

The computation of the AO control basis requires a high SNR full influence function data set (Section **Error! Reference source not found.**). High SNR is achieved by setting the number of push-pull repetitions to the maximum (i.e. **n**=51 in Table 5), and by applying a large amplitude RMS command (i.e. large **A** in Table 5).

Deleted: 9

Commands and actions	Notes
$ aoidl	Start a service IDL shell
IDL> `mm = obj_new('mirmodes', opd_dir_list, data_dir = data_dir, save_dir = save_dir, binning = binning)` Returns the `mm` IDL object that contains all procedures used in the following steps.	Initialize the `mirmodes` tool. Required inputs: `opd_dir_list`: List of TRACKNUM [Tr1, Tr2, … TrN] of the full high SNR influence function data set. `data_dir`: Directory in which the TRACKNUM folders are located (i.e. `if_function` directory in **Error! Reference source not found.**). Optional inputs: `save_dir`: Folder where all data generated by the `mirmodes` tool will be saved. (Default: `save_dir = if_function/{tracknum01}_mirmodes` ; where `{tracknum01}` is the YYYYMMDD of Tr1 in the list of TRACKNUM folders). `binning`: Binning factor to be applied to all OPDs (default: `binning`=2).
IDL> `mm->summary_opds`	Use this command to display a summary table for all OPD files in `opd_dir_list`. The displayed summary table will look like this:

tracknum	**nn**	**amp [nm]**	**first**	**last**	**missing**	**sampling**
20130826_150704	11	150.0	0	49	0	990 x 1002
20130826_150915	11	150.0	50	99	0	990 x 1002
….						

tracknum: tracking number folder
nn: number of 4D measurements to produce a single OPD file (**n** parameter in Table 5).
Amp[1]**:** Modal amplitude applied during measurement with 4D (**A** parameter in Table 5).
First: First mode (e.g. `mode_001.sav`) contained in the tracking number folder.
Last: Last mode (e.g. `mode_005.sav`) contained in the tracking number folder.
Missing: Lets you know if there are missing modes in the [First, Last] range.
Sampling: Size (in pixels) of OPDs at full resolution.

Deleted: Table 1

[1] The amplitude reported by this table is taken from the file info.txt located in each tracknum directory. The amplitude value saved in this file is not the actual one applied. This bug has been reported and its correction is still pending. The actual amplitude can be checked with the command mm-> check_noise

IDL> mm->**collect_opds**	Gather all OPDs from all tracking number folders. A file for each mode (`mode_XXX.sav`) will be saved in the `save_dir` directory. *Note 1*: If there is more than one acquisition for a given mode (e.g. two files `mode_XXX.sav` for the same mode in different tracking number folders) ALL files will be used to compute the Average OPD for that mode. *Note 2*: The saved OPDs will be normalized to a unitary command. The `mirmodes` tool uses the `MODE_NORMALIZED_TO_AMP` and `applied_amp` information in each `mode_XXX.sav` file in order to process each mode as needed.
IDL> mm->**mask_calibration, SHOW=SHOW**	The masks of all OPDs are used to compute the Union and Intersection Masks at the full resolution. Then, the rebinned Union and Intersection masks are computed. *Note 1:* Set the keyword SHOW to display the masks as they are calibrated. *Note 2:* Table 6 lists the files where mask calibration data is saved.
IDL> mm->**produce_ifmat,** modeborder = modeborder	Produces the preliminary optical interaction matrix (i.e. the IF matrix) by rearranging all OPDs at the binned resolution in a matrix form. *Note 1*: As a baseline, OPD data is extracted using the Intersection Masks calibrated in the previous step. *Note 2:* Piston (i.e. mean value) will be removed from all OPDs. *Note 3:* Tip and Tilt (mainly due to vibrations) will be removed from mode number `modeborder` onwards (default: `modeborder = 10`). *Note 4:* The preliminary IF matrix is saved in the file `IFmatrix.sav` in the `save_dir` directory.
IDL> mm->**check_noise**	This command displays the plots in Figure 8. Use these plots to inspect the SNR of the modes in the IF matrix produced in the previous step. *Note 1*: If the SNR of the acquisition is not sufficient, acquire a new set using more push-pull repetitions, and/or larger amplitudes (c.f. Table 5).

File	**Contents**	**Resolution**	**Baseline for IF Matrix**
`Masks_data.sav`	General RAW mask info	Full	
`union_mask_extraction.sav`	Union mask calibration	Full	
`intersection_mask_extraction.sav`	Intersection mask calibration	Full	YES
`union_gmask_rebin.sav`	Union mask calibration	Binned	
`intersect_gmask_rebin.sav`	Intersection mask calibration	Binned	YES

Table 6. Files containing all mask data produced by `mask_calibration` method of the `mirmodes` tool.

Low SNR IF data set (**n**=3, $\mathbf{A}_{max}$=2.5×10^3)

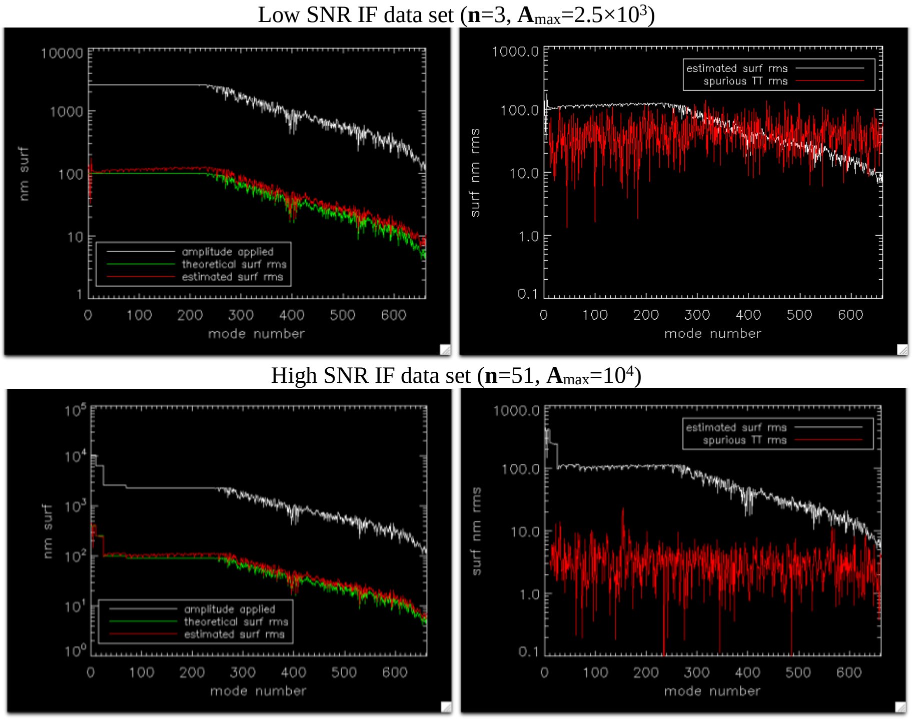


Figure 8. SNR IF data set plots produced by the `mm->check_noise` command. For a high SNR data set, the spurious TT rms values need to be well below the surface RMS value applied to the modes. Recall that for mirror modes the theoretical surface RMS value is equal to ($\mathbf{A}/\sqrt{n_{acts}}$).

14.2 Pupil mask tweaking procedure

The intersection mask (generated by the `mask_calibration` procedure described above) may contain pixels at the border of the pupil (i.e. close to the central obscuration and at the outer perimeter) that contain corrupted information, mostly arising from the binning of pixels with valid and invalid information. If these pixels are not removed from the IF matrix, large KL fitting errors and modal forces will prevail.

We have developed an interactive mask tweaking procedure that finds the "bad pixels" by fitting a subset of 45 Zernike modes with the preliminary inverse IF matrix. The interactive procedure will first allow the user to trim the mask at the outer border, and then the inner border (i.e. close to the central obscuration). In addition to the interactive procedure, the user can also use a parametric one that takes as parameters the desired size of the mask (r<0.999) and central obscuration (oc>0.111), as described below.

Commands and actions	**Notes**
IDL> `mm->`**`inv_ifmat`**	Compute the pseudo-inverse of the preliminary IF matrix. *Note 1*: This inverse IF matrix will be used solely to fit a "test set" of Zernike modes, as required by the interactive mask tweaking procedure. *Note 2*: The plot of singular values of the IF matrix, and the shape of the last eigenmode (lowest sensitivity) will be shown. *Note 3:* The preliminary inverse IF matrix, and the SVD data, are saved in the files `inv_IFmatrix.sav` and `SVD.sav` in the `save_dir` directory, respectively.
IDL> `mm->`**`interactive_tweak_mask`**	Launch the interactive mask tweaking procedure. At the end of the procedure, a `mask_ID` name will be requested to the user (see Figure 9). *Note 1*: The `mask_ID` uniquely identifies all subsequent data files generated with this mask.
(Optional step) IDL> `mm->`**`tweak_mask`**`, mask_id, orig_mask_id=orig_mask_id, oc=oc, radmax=radmax, remove_x=remove_x, remove_y=remove_y`	Tweak an existing mask parametrically. Inputs: `mask_id`: mask ID of the NEW mask to be generated. `orig_mask_id`: mask ID of the mask to be further tweaked. (Default: preliminary intersection mask). `oc`: central obscuration ratio (in general, `oc`>0.111) `radmax`: mask outer radius (in general, `radmax`<0.999) `remove_x,remove_y`: coordinates of additional arbitrary pixels to be removed.
IDL> `mm->`**`tweak_ifmat`**`, mask_ID`	Apply the selected tweaked mask to the IF data set and save a new IF matrix.

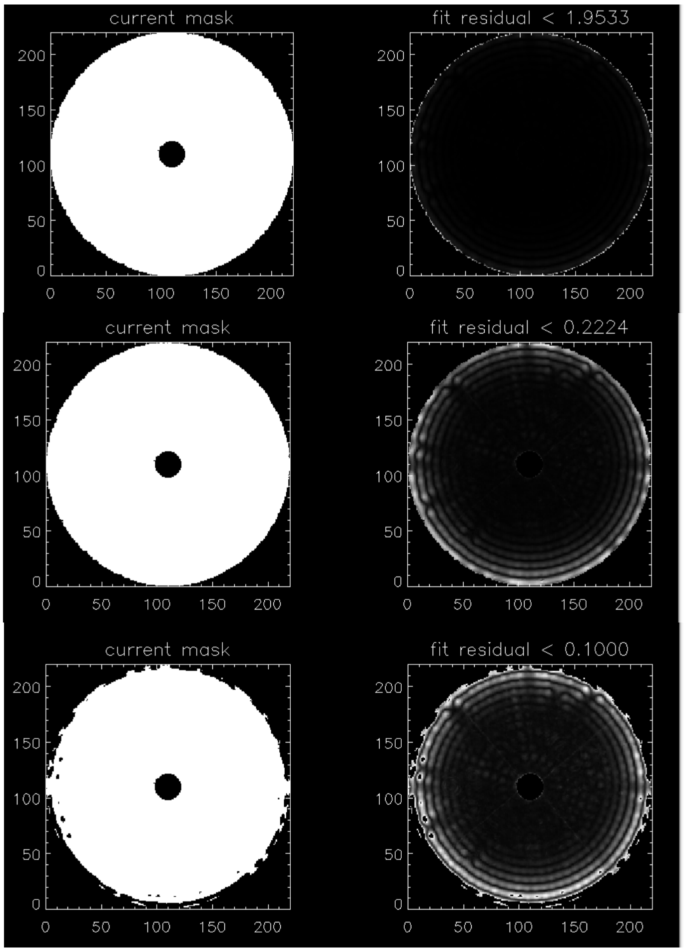


Figure 9. Interactive mask tweaking visualization, showing the current mask on the left, and an estimate of the fitting error on the right. The preliminary intersection mask (top row) produces a large fitting error at the outer border. The interactive procedure allows the user to reduce gradually the number of bad pixels until a reasonable fitting error at the borders is found (middle row). The bottom row shows the effects on the mask when too many pixels are removed.

14.3 Slaving of actuators

Sometimes it is required to slave down actuators that are not properly seen, e.g. actuators partially covered by the central obscuration, or close to shell imperfections, like the mouse hole on the Thin Shell TS4. If these actuators are not slaved they may cause AO instabilities, as they will likely saturate in force. In fact, most of the times the list of slaved actuators is determined when testing the performance of AO reconstructors, and so it will be required at that point to redefine the AO modal basis. The procedure discussed below will allow the user to compute a new set of mirror modes (an IF matrix) with slaved actuators based on an existing full IF data set (referred to `mm` below).

Commands and actions	Notes
IDL> `mm->slave_ifmat, act_slave, slave_id, mask_id=mask_id`	Required inputs: `act_slave`: list of actuators to slave down. `slave_ID`: String that uniquely identifies all subsequent data files generated with this configuration. `mask_ID`: Name of selected tweaked mask.

For example, the following code creates an IF matrix with all actuators in the first ring slaved down. A file named `IFmatrix_tweak01_slave01.sav` containing the slaved IF matrix and a file named `slavedata_tweak01_slave01.sav` will be save in the `mm->save_dir()` directory.

```
mm = obj_new('mirmodes', opd_dir_list, data_dir=data_dir)
adsec_save = mm->adsec_save()
act_slave = adsec_save.mir_act2act[0:8]
mm->slave_ifmat, act_slave, 'slave01', mask_id='tweak01'
```

14.4 Auxiliary `mirmodes` procedures

Additional procedures available in the `mirmodes` class that can be handy to the user are listed below.

Command	Notes
IDL> `mm->showme_the_modes, mode_num_idx=mode_num_idx, mask_id=mask_id, slave_id=slave_id, zoom=zoom, nrows=nrows, ncol=ncol`	Displays the 2D maps of the requested modes, identified by `mask_id` and `slave_id` tags, arranged in `nrows×ncols` (see Figure 10). `mode_num_idx`: index list of modes to display (Default: select ALL modes in the IF matrix). `nrows, ncols`: Select the number of rows or columns of the display (only one of the two is sufficient). `zoom`: display zoom factor (Default: 0.25)

IDL> print, mm->**data_dir()**	Prints the directory where TRACKNUM folders are located.
IDL> print, mm->**opd_dir_list()**	Prints the list of TRACKNUM folders
IDL> print, mm->**ndirs()**	Prints the number of TRACKNUM folders
IDL> print, mm->**save_dir()**	Prints the directory in which all processed data is saved.
IDL> adsec_save=mm->**adsec_save()**	Returns the adsec_save structure associated with the IF data set.

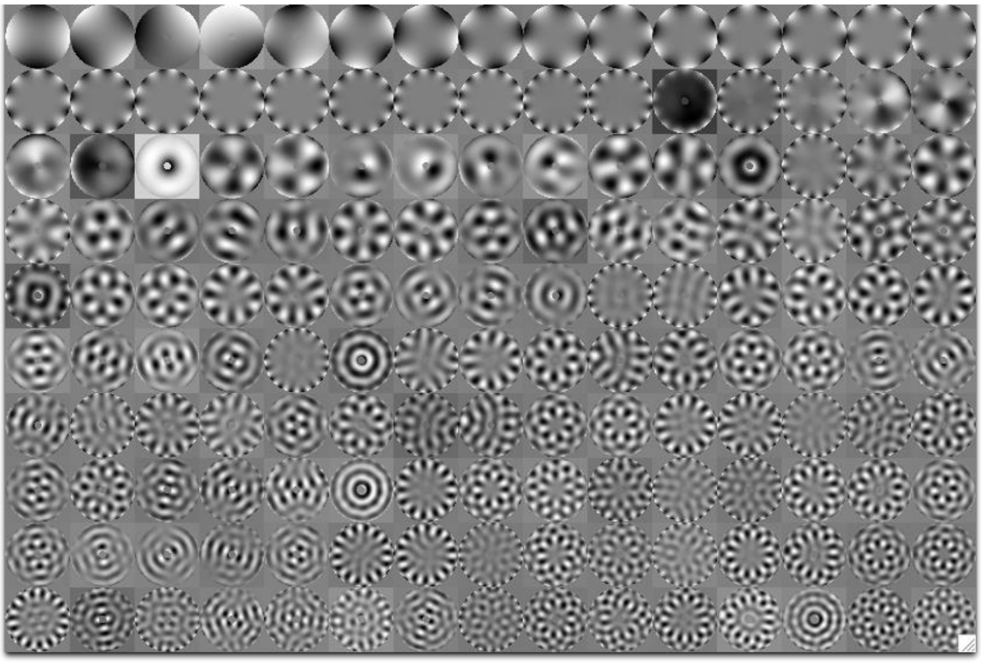


Figure 10. First 150 mirror modes shown with the command: `mm->showme_the_modes, mode_num_idx=lindgen(150), ncol=15`. Note that since no `mask_id` was specified, by default the routine shows the modes in the preliminary IF matrix.

14.5 KL modes fitting

Commands and actions	Notes
IDL> `mm->`**`inv_ifmat`**`, mask_id=mask_ID, slave_id=slave_ID`	Compute the pseudo-inverse of the selected IF matrix. This inverse IF matrix will be used to fit the KL modes. Input Parameters: `mask_ID`: name of selected tweaked mask. `slave_ID`: ID of slaved data (optional).
IDL> `klf = `**`obj_new`**`('klfit', mm, mask_ID, slave_ID)`	Initialize the KL fitting tool. Note that the mirror modes object `mm` is the only mandatory input.
IDL> `klf->`**`create_theo_kls`**`, nzern=nzern, nkl=nkl`	Computes a set of theoretical KL modes, using the method of orthogonalizing the Kolmogorov covariance matrix of Zernike modes. The resulting KL modes are then re-orthogonalized on the selected tweaked mask, with pure tip and tilt modes (Z2 and Z3) replacing the first two KL modes. Input parameters: `nzern`: number of Zernike modes (default: `nzern`=2016). `nkl`: number of computed KL modes (default: `nkl`=703).
IDL> `klf->`**`create_ortho_fitted_kls`**`, klmax=klmax`	Computes a total of `klmax` KL modes fitted by the mirror modes (default: `klmax` =405). *Note 1*: The maximum number of modes controlled in the commissioning of the LBT AO system is `klmax` =400. *Note 2*: The maximum `klmax` that could be in principle computed is 672 minus the number of disabled actuators.
IDL> `klf->`**`check_fitting`**	Displays the fitting error and peak force required to produce a 1 µm RMS SURF fitted KL mode (Figure 11).

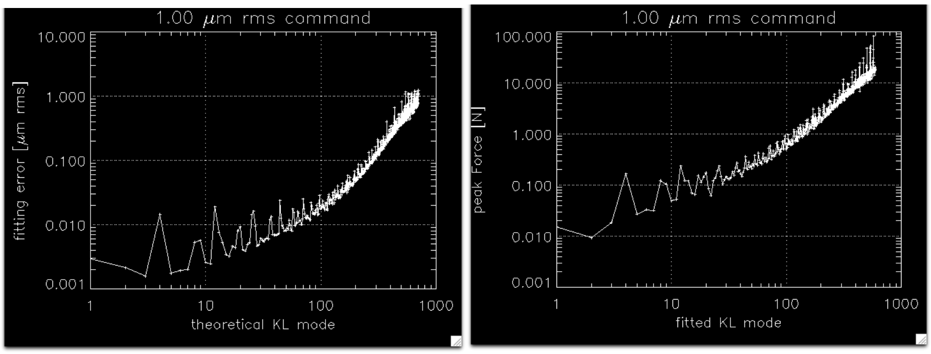


Figure 11. (Left) Fitting error of theoretical KL modes by the ASM IF set. (Right) Peak force required to fit the re-orthonormalized KL modes. As a rule of thumb, fitting error of low-order modes (other than focus, which is the KL#4) must be less than 0.005 µm RMS for a 1.0 µm RMS command.

Additional routines that can be handy to the user are also available in the klfit class. These additional procedures are listed below.

Command	**Notes**
IDL> `klf->`**`showme_the_modes`**`, KLtype, mode_num_idx=mode_num_idx, zoom=zoom, nrows=nrows, ncol=ncol`	Displays the 2D maps of the requested KL modes (see Figure 12). Required input: `KLtype`: Set it to either: `"THEO"`, `"FIT"`, or `"ORTHOFIT"` to show the theoretical, fitted, or ortho-fitted KL modes. For an explanation of all other input parameters see `mm->`**`showme_the_modes`**
IDL> print, `klf->`**`kl_theo_dir()`**	Shows where theoretical KL data is stored.
IDL> print, `klf->`**`kl_fit_dir()`**	Shows where fitted KL data is stored.

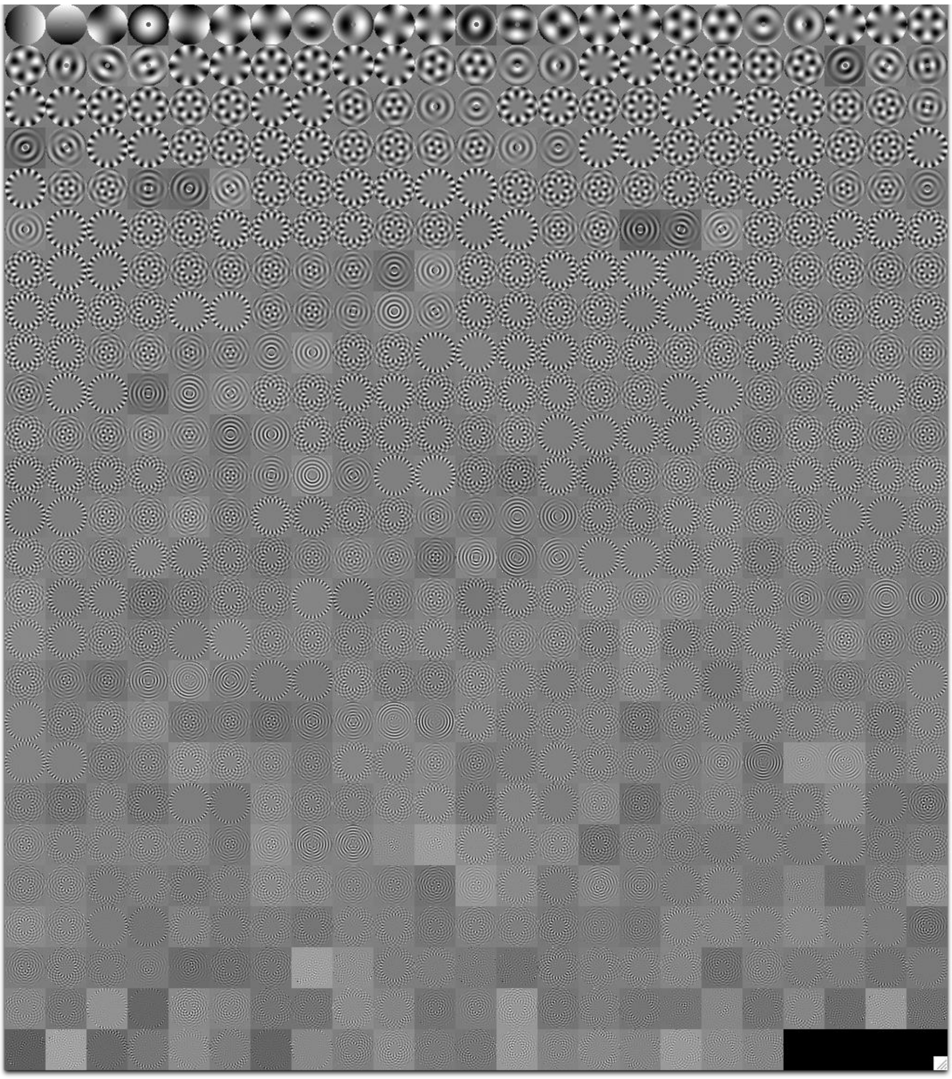


Figure 12. Example of a final set of re-orthonormalized fitted KL modes visualized with the command: `klf->showme_the_modes, 'ORTHOFIT', ncol=23, zoom=0.2`

The final product of the whole "definition of the AO modal basis" process is the so-called **Modes-to-Commands matrix (M2C)**, which was saved in the file `m2c.fits` located in the `klf->`**`kl_fit_dir()`** directory. The user will have to copy manually this file to a specific directory, as indicated below.

Also, the final IF data set and KL data set must be saved for use in other applications (e.g. `elab_lib` uses these files to load the modal shapes and reconstruct the WF for a given AO dataset acquisition).

The commands and actions to accomplish the final data savings and transfers are listed below.

Commands and actions	**Notes**
`$ make_m2c.py KL_NAME`	Run this python script to create a set of directories that will be uniquely associated with a new KL modal basis. Required parameter: `KL_NAME`: Name of the AO control basis. By convention, the name should be like: `KL_vXX` where `XX` is a number. For example, at the time of writing this report, the last KL basis on the SX system (FLAO2 system) was `KL_v18`.
`$ cp m2c.fits [destination]`	Copy the file `m2c.fits` from the `klf->`**`kl_fit_dir()`** directory to the directory created with the command above. *Note 1*: The `make_m2c.py` script will display the name of the directory to which the file `m2c.fits` must be copied.
IDL> `mm->`**`save_MMmatrix`**`, sys_id, mask_id=mask_id, slave_id=slave_id, fname_extra_tag=fname_extra_tag`	Save the final IF matrix data to a .sav file in the `mm->`**`save_dir()`** directory. Inputs: `sys_id`: System identifier. By convention, these identifiers have been previously used: `FLAO1, FLAO2, MAG585`, to refer to the DX, SX, and Magellan ASM, respectively. Please continue to use them. `fname_extra_tag`: String to be appended to the .sav file. *Note 1*: The procedure will display the full name of the .sav file.

IDL> `klf->`**`save_KLmatrix`**`, kl_id, fname_extra_tag=fname_extra_tag`	Save the final ortho-fitted KL matrix data to a .sav file in the `klf->`**`kl_fit_dir`**`()` directory. Inputs: `kl_id`: Choose the same `KL_NAME` selected above (e.g. KL_vXX). `fname_extra_tag`: String to be appended to the .sav file. *Note 1*: The procedure will display the full name of the .sav file.
`$ cp MMmatrix_....sav /local/phase_maps/` `$ cp KLmatrix_....sav /local/phase_maps/`	Copy manually the .sav files to the `/local/phase_maps/` repository. *Note 1*: Contact your local `elab_lib` representative for further information on the use of these files.

-oOo-

Doc_info_start
Title: LBT672 optical calibration procedure
Document Type: Technical Report
Source: INAF-Osservatorio Astrofisico di Arcetri
Issued by: A. Riccardi
Date_of_Issue: 19 Jan 2016
Revised by:
Date_of_Revision:
Checked by:
Date_of_Check:
Accepted by:
Date_of_Acceptance:
Released by:
Date_of_Release:
File Type:MS-WORD
Local Name:
Category: 600
Sub-Category: 640
Assembly: 640
Sub-Assembly:
Part Name:
CAN designation: 640f038
Revision: A-Draft03
Doc_info_end