Acquisition of influence functions and flattening

Acquisition of influence functions

Acquisition of raw interferograms

Preliminary operations:

1. Verify that a fiber link is routed from the SwitchBCU to the Trigger Fiber to BNC Interface (small box in the lower left or right tree house); verify that the trigger interface is powered on (5V to 6V); verify that a BNC cable is routed from the trigger interface to the 4D trigger input.
2. Start 4Sight 1.8 on the Interferometer PC. For VNC connections, the IP's are the following: Arcetri 4D: 193.206.155.78 (password m1r.....); LBTO 4D: 10.144.0.92 (use the account PhaseCam, with password 4D).
3. launch the default script, that starts the Pyro server: this is done clicking the green arrow on the 4Sight panel (with a D).
4. open the python debug console (main menu-> tools) and type: from I4D import comm4d
5. to be sure reload the default configuration file by clicking File-> Open Config and select the file saved during the initial setup.
6. On the adsecdx (or adsecsx) machine login as AOeng and start the interferometer communication process by typing 'Interferometerctrl -i interferometerctrl' (verify that the process is up and running).
7. From adsecdx (or adsecsx) type "terminal" return, this will start the IDL Terminal communicating with the ASM Unit.
8. Verify that the corresponding (DX or SX) ASM Unit is ON and in AOSet mode.

Define the mode numbers and the mode amplitude to be sampled on the acquisition steps. you may refer to the following table:

Modes Numbers	N. of Iterations	Mode amplitudes
0-9	51	500 nm
10-250	3	150 nm
251-500	3	150 nm
501-671 (or less)	3	150 nm

Iterations over the wanted modes:

4D side

1. verify the alignment and refine it if the case (add more details later).
2. save the new configuration file with today's date. (e.g. 20060709.configuration.ini)
3. copy the new configuration file in D:\4D\Data\configuration.

4. in C:\Obelix_CVS\Supervisor\PyModules\I4D\Constant4d.py check the variable: 'ConfigurationFile', and update it with the new filename.
5. re-compile the script by typing on the Debug console: from I4D import Constant4d then reload(Constant4d) then reload(comm4d)
6. if the automatic trigger selection is used, then skip directly to "IDL side"
7. if the automatic trigger selection is not used (obsolete):
    - on the 4Sight GUI, open the Camera Settings panel and select the external trigger line. Arcetri: TTL0. LBTO: OPTO0
    - at this point the interferometer image will be black or frozen, this is nornal as an external trigger is expected. If a pop up window is asking for waiting the trigger, answer NO.
    - on the python console type: comm4d.capture(1000)
    - launch the command just 1 sec before launching the following IDL command.

IDL side

1. on the IDL Terminal (AdOpt>) launch the command:
    - print, i4d_opt_intmat(from, to,MANUAL=manual, KL=kl, average=average, amprms=amprms, simulation=simulation, tracknum=tracknum).
    - meaning of the parameters:
        - from, to: first and last mode to be sampled.
        - /manual: set this keyword if you want to launch manually the 4D scripts (obsolete).
        - /KL: set this keyword if you want to sample KL modes. if the case, verify the modal basis. it is linked in /adsec_utilities/make_modal_disturb.pro
        - average=number of +1, -1, +1, -1... application of modes to the mirror. if not provided the default value (from configuration file) is passed.
        - amprms= max mode rms for the application. such value is scaled as the mode number increases, to reduce the force required for the application.
        - simulation= enables the disturb and the SwitchBCU trigger with the shell ripped. the system behaviour is the same, with the only exceptions that the modes will not be applied and the trigger frequency is 16.6Hz. This mode can be used to test that the trigger is working.
        - tracknum: output, is the newly created tracking number of the dataset
    - in the log file of the idlctrl process, check for the successfull completion of the sampling and take note of the resulting tracking number.
2. repeat the steps 1 to 2 for the modes sets in the table.

The raw data are saved in the directory named after the tracknum in D:\4D\Data\Zcopy\.

Interferograms production

4D python interface

1. type: comm4d.produce(tracking_number). tracking_number must be passed as a string. the scripts takes care of assembling the data full path, here just the tracking number is required
2. for a multiple data reduction:
    - put the tracking numbers to be reduced in a string array: a=['track1','track2','track3']
    - for x in a:
        - comm4d.produce(x) this is an automatic python indentation; type return twice (until the process starts)
3. data are now stored in tracking_number/hdf5

IDL side

1. open a IDL terminal (there is no need for MSGDRTDB connection, so, simple aoidl and @startup at the prompt is enough)
2. print, if_redux(tracking_number, modal=modal, KL=kl)
    - /modal if mirror modes are sampled
    - /KL if the KL basis is used
3. data are now processed using the differential algorithm and stored in the folder ADOPT_MEAS/adsec_calib/ if_functions/tracknum, together with:
    - a copy of the adsec structure (including the FF matrix that produced that modes)
    - the info.txt file, copy of the i4d_opt_inmtat configuration file, containing the OVS, the template form, the decimation of the BCU trigger....
    - a copy of the 4D configuration.ini file
    - mask_intersection.sav file
    - tip_tilt vector on the data (tt_vector.sav)
    - in the ADOPT_MEAS/adsec_calib folder the file if4d.txt will be updated, and the mark "IF Redux done" indicates that data have been processed. on the same line, missing modes are highlighted (i.e. those modes not sampled because of laser intensity fluctuations.

Interaction matrix buidling

IDL side

1. look in the if4d.txt file and define the tracking numbers that compose the best set of data for your interaction matrix.
2. put them in a string vector, ordered by modes: track=['20100101_010101', '20100101_010200'....]. Note: the tracking numbers must span a range of mode number starting from 0 to the maximum desired without gaps.
3. print, build_int_mat(track): the script will read through the tracking numbers and build the cube of interferograms. The interaction matrix is stored as "flat_data_n_modes.sav" in the directory $ADOPT_MEAS/flat/tracknum[0]/

Flattening

IDL side

1. Choose an interaction matrix, selecting the corresponding tracking number (see above).
2. print, opt_int_mat_reduce2(nmodes, track=track, nmeas=nmeas,ZERN_NOT_FLAT = zern_mode2not_flatten, manual=manual):
    - meaning of the keyword:
    - nmodes: n of modes to apply
    - track= tracking number of the interaction matrix, stored in /flat/track
    - nmeas= n of 4D frames to be collected to calculate to opd to flatten
    - zern not flat= zern modes to not flatten (vector of index)
    - manua (obsolete): set this keyword to skip automatic 4D sampling of opd to flatten. if selected, do the following:
        - from the python console, print comm4d.flat(n): n default =10
        - the script will capture n images in the 4D folder Zcopy/img4d
        - IDL will search image inside this folder and use them as opd to flatten
3. The procedure will prompt if the flat should be applied and the resulting flattening data will be saved in the directory $ADOPT_MEAS/adsec_calib/ with the current tracking number. The flat commands can be retrieved from the ADSEC Control GUI at any later time.

-- ArmandoRiccardi - 22 Sep 2010

Attachments (1)

Attach files Show options

Procedure_for_AdSec672_Influence_Functions_sampling.doc (109.00K)
Version 1 uploaded by Runa Briguglio on 27 Sep 2010 - 21:03

Select all

Edit | Attach | Print version | History: r7 < r6 < r5 < r4 | Backlinks | View wiki text | Edit wiki text | More topic actions

Topic revision: r7 - 14 Sep 2012, RunaBriguglio