

# VST Project

## Test Procedure in Europe

### Active Optics

Doc. no. : VST-PRO-OAC-AO-001

Date: 2006-10-26  
Issue: 1.0

*DRAFT VERSION*

|                    | <b>Name</b>                       | <b>Date</b> | <b>Signature</b> |
|--------------------|-----------------------------------|-------------|------------------|
| <i>Written by</i>  | M. Brescia, L. Marty, P. Schipani | 2006-10-26  |                  |
| <i>Checked by</i>  |                                   |             |                  |
| <i>Released by</i> |                                   |             |                  |

### Change Record

| Issue/Rev. | Date       | Section/Page affected | Reason/Remarks |
|------------|------------|-----------------------|----------------|
| 1.0        | 2006-10-26 | All                   | First Issue    |
|            |            |                       |                |
|            |            |                       |                |

## Index

|       |   |    |
|-------|---|----|
| 1     | Introduction .....                                | 4  |
| 2     | Scope of the tests .....                          | 4  |
| 3     | Applicable Documents.....                         | 6  |
| 4     | Test Conditions.....                              | 7  |
| 5     | Definition of tests .....                         | 8  |
| 5.1   | M1 Low-Level actuator test.....                   | 9  |
| 5.1.1 | Test principle .....                              | 9  |
| 5.1.2 | Collected data .....                              | 11 |
| 5.1.3 | Analysis .....                                    | 12 |
| 5.1.4 | Step by step procedure.....                       | 12 |
| 5.2   | M2 Low-Level hexapod test.....                    | 13 |
| 5.2.1 | Test principle .....                              | 13 |
| 5.2.2 | Collected data .....                              | 14 |
| 5.2.3 | Analysis .....                                    | 14 |
| 5.2.4 | Step by step procedure.....                       | 15 |
| 5.3   | AO Open Loop test.....                            | 17 |
| 5.3.1 | Test principle .....                              | 17 |
| 5.3.2 | Collected data .....                              | 17 |
| 5.3.3 | Analysis .....                                    | 17 |
| 5.3.4 | Step by step procedure.....                       | 17 |
| 5.4   | AO Closed Loop test.....                          | 19 |
| 5.4.1 | Test principle .....                              | 19 |
| 5.4.2 | Collected data .....                              | 19 |
| 5.4.3 | Analysis .....                                    | 19 |
| 5.4.4 | Step by step procedure.....                       | 20 |
| 6     | Definition of test matrices.....                  | 22 |
| 6.1   | M1 low level test matrix sample.....              | 22 |
| 6.2   | M2 low level test matrix sample.....              | 24 |
| 6.3   | Active Optics Open Loop test matrix sample.....   | 26 |
| 6.4   | Active Optics Closed Loop test matrix sample..... | 28 |
| 7     | Test results.....                                 | 30 |

## Reference Documents

| id. | Document code          | Title                                | Source | Date     | Issue |
|-----|------------------------|--------------------------------------|--------|----------|-------|
| RD1 | VLT-PRO-ESO-10200-1443 | Commissioning test procedures/report | ESO    | 03/06/98 | 1.0   |

## 1 Introduction

This document contains procedures to test the functional quality of the VST active optics (AO) system in Europe. Generally speaking, the test procedure should describe in detail all basic operations needed to perform a verification by plan. Therefore, the test procedure reports all the following information:

- Scope of the test: a theoretical description of the test, its scope e purpose, identifying also all the verification items involved;
- All the applicable documentation list (i.e. relevant documents to verify or referred for the test procedure);
- Test procedure: detailed list of all operations required to perform the functional tests;
- Test conditions: conditions of the system to perform tests, operation tools, environmental conditions, special requirements on tested items, calibration requirements;
- Test results: all the procedures to process raw data for the final presentation of the results;

## 2 Scope of the tests

Scope of the tests is to measure and verify telescope active optics functionality and performance in Europe, basically in terms of system functionality and reliability.

The term “active optics” is referred to the control system of all active devices employed to balance and correct primary (M1) and secondary (M2) mirror optical aberrations and misalignments basically induced by thermal gradient and altitude (ALT) axis inclination during exposures. The active system is composed of an actuator net under and around M1 surface (84 axial and 24 radial devices), able to correct local deformation of the thin mirror surface and of a double-stage hexapod able to correct de-focus, de-centering and tilt effects by moving M2. The first stage, based on a six-degree-of-freedom tilt platform will be used at the beginning of exposure sessions to align the optics, while the second stage, based on a six piezo-actuator system with sub-micron resolution, can be used during exposures to maintain aligned the optical system without affecting the image quality.



**Fig. 1 – the M1 axial and radial actuators**

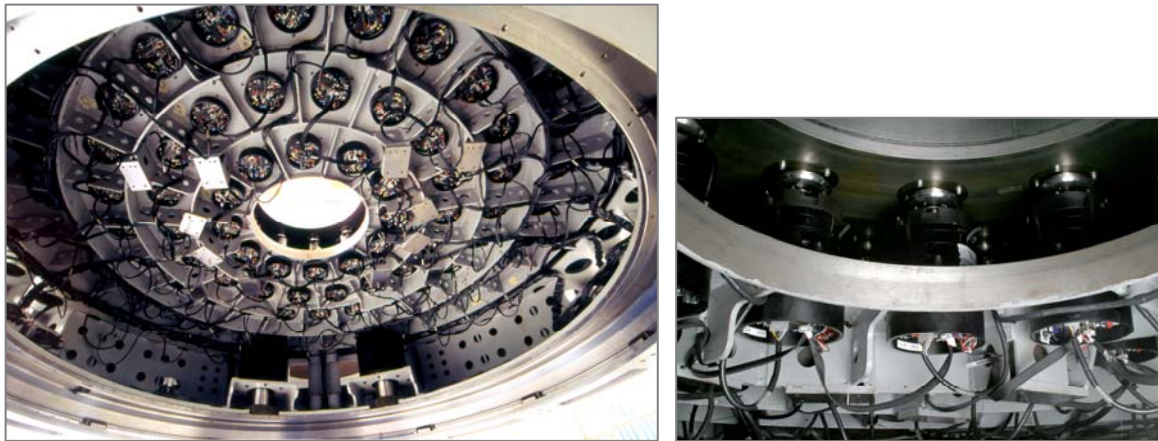


Fig. 2 – the positions of actuators on bottom and top of the cell



Fig. 3 – the M2 hexapod nr. 1 and nr. 2

Although the VST active optics system is based on the same concepts of the VLT one, there are some differences which make not possible the full reuse of the original modules (basically *actcon*). Therefore a new module (*vstactcon*) has been realized, reporting all the necessary modifications to fit VST requirements. Most of the modifications were needed because of physical differences between the VLT and VST AO systems:

- VST does not have a separate “passive system” like the VLT, to adjust the position of the M1 mirror. Therefore, all the communication with the specific LCU module (*mIpsServer*) has been removed
- VST does not have Nasmyth A & B foci like the VLT, but only the Cassegrain one
- VST has “active” radial actuators to support the M1 mirror. In order to let them work with the correct force, it is needed to send them forces update continuously depending on the altitude angle. This is a novelty with respect to VLT. The solution has been found attaching an event on the altitude position db attribute, so that a command with new radial forces is sent each time the altitude position changes (by at least one degree)
- VST has a different number of axial actuators to support the M1 mirror
- On the VST there is only one control LCU for M1 and M2 mirrors
- VST mirrors are different from the VLT ones


- Other main physical differences, like the M2 positioning device, are handled at level of LCU
- For the AO operations, while the original VLT scheme foresees the command sequence ONECAL and CYCLAO delivery, in the VST the software class prsAH\_AO provides the possibility to implement other two preset command sequences to be added to the default one:
  - ONECAL, CYCLAO (the original one): this is the configuration by default provided also on the VLT AO control scheme. It is composed by ONECAL, command to perform one-shot AO calibrated correction in open loop mode, i.e. based on pre-calculated aberration tables related to the current ALT axis inclination (see chapter 4 for details about open/close AO modes). With CYCLAO, it is then possible to perform cyclic AO corrections, in closed loop mode, i.e. with image analysis feedback
  - Only ONECAL: only one-shot AO calibrated correction in open loop mode
  - ONECAL, CYCLCAL: here CYCLCAL is used to perform open loop cyclic AO calibrated corrections.

Due to the complexity of the active optics system, two levels of test are needed. The first one is a low-level test, with the aim to investigate the functionality and reliability of hardware devices (actuators, hexapod, pinhole), by means of cyclic combined motions in all their functional ranges. The second one is an high-level test, with the aim to verify all AO coordinated functionalities, by means of correct action flow in all planned operation modes, starting from aberration and deformation calculation down to hardware device positioning command. Just blind active optics capabilities of the telescope can be measured and verified, because it cannot be used with real image feedback taken from TCCD in the Italy integration facility.

### 3 Applicable Documents

The following documents are considered relevant for the Active Optics system:

| AD | CODE                   | TITLE                            | SOURCE | DATE     | REL   |
|----|------------------------|----------------------------------|--------|----------|-------|
| 1  | VST-SPE-OAC-25000-1191 | VSTACT module description        | VST    | in prep. | 1.0   |
| 2  | VST-SPE-OAC-25000-1192 | VSTACTCAL module description     | VST    | in prep. | 1.0   |
| 3  | VST-SPE-OAC-25000-1193 | VSTACTCON module description     | VST    | in prep. | 1.0   |
| 4  | VST-SPE-OAC-25000-1194 | VSTACTGUI module description     | VST    | in prep. | 1.0   |
| 5  | VST-SPE-OAC-25000-1178 | VSTAXEI module description       | VST    | in prep. | 1.0   |
| 6  | VST-SPE-OAC-25000-1209 | VSTM1AS module description       | VST    | in prep. | 1.0   |
| 7  | VST-SPE-OAC-25000-1210 | VSTM1WS module description       | VST    | in prep. | 1.0   |
| 8  | VST-SPE-OAC-25000-1211 | VSTM2CO module description       | VST    | in prep. | 1.0   |
| 9  | VST-SPE-OAC-25000-1212 | VSTM2WS module description       | VST    | in prep. | 1.0   |
| 10 | VST-SPE-OAC-25000-1189 | VSTPH module description         | VST    | in prep. | 1.0   |
| 11 | VLT-MAN-ESO-17240-0672 | CCD control SW User Manual       | ESO    | 25/09/98 | 1.6   |
| 12 | VLT-SPE-ESO-17210-1173 | Active Optics Design Description | ESO    | 20/10/97 | draft |
| 13 | VST-SPE-OAC-21000-1028 | Final Optics Design Summary      | OAC    | 25/10/00 | 1.0   |
| 14 | VST-SW-OAC-M2-002      | HEX1 and HEX2 Test Report        | OAC    | 7/04/06  | 1.0   |

|  |  |   |
|--|--|---|
|  | Project: VST<br>Doc.VST-PRO-OAC-AO-001.DOC | Rev.: 1.0<br>Date: 2006/10/26<br>Pag. 7 of 31 |
|--|--|---|

## 4 Test Conditions

Basic condition for AO tests is, as obvious, the full operability of M1 actuators, M2 two-stage hexapod and pinhole axis, with M1 and M2 mirror dummies and astatic levers mounted on the telescope. In principle, before to perform AO activities, it is therefore suggested to test single devices separately. From the functional point of view, the AO control system is based on two operation modes:

- Open loop: the positioning of M1 actuators and M2 hexapod is based on pre-calculated lookup tables reporting M1 and M2 positions at various ALT angles. During tests in Italy dummy lookup tables can be used, because they need real sky observing conditions. With this configuration, the control system is able to correct mirror deformations induced by their weight only.
- Closed loop: the positioning of M1 actuators and M2 hexapod is calculated by the comparison between the real target image and a reference one, taken during exposures by means of a technical CCD (TCCD), a pinhole and the Shack-Hartmann (SH) wavefront sensor system. With this control configuration it is possible to perform a fine analysis and calculation of local aberrations and deformations of different order (polynomial coefficients) to be converted in M1 actuator forces and M2 hexapod combined motions to be applied. During tests in Italy it is possible to employ this control configuration by means of simulated images (TCCD running in simulation mode). But in this condition, at the end of a M1 and M2 correction, it will be possible to test only the accuracy of motion, without having a real image comparison feedback after re-positioning.

From the control point of view, the following basic equipment is directly involved, and their full functionality required:

- Wavefront sensor TCCD control LCU (environment ltviac) with TCCD in simulation mode
- Pinhole control LCU (environment ltvadc)
- M1 actuator and M2 hexapod control LCU (environment ltm12)
- ALT axis control LCU (environment ltvalt)
- TCS control WS (environment wtvst)

From the SW point of view, the following applications are usually employed, depending on the specific level of test/verification. They are partially derived/modified from VLT TCS SW package, so for most of them, their use and interface can be considered the same:

### Control tools:

- vstguiStatus: TCS mode switching control interface
- vstcangui: CAN bus low-level control engineering user interface
- vstcanTest: low-level M1/M2 device control engineering user interface (Fig. 5)
- vstcanActuatorTest: low-level M1 actuator test script
- vstm2Test: low-level M2 hexapod test script
- vstactguiEngineer: main AO control engineering interface (Fig. 4)
- vstactguiCoeffsDisplay: image analysis aberration coefficient display interface
- vstaxei: ALT axis control/diagnostic user interface
- vstguiTCS: main TCS control user interface

## Diagnostic tools:

- VLTSW engineering applications (logMonitor, ccseiDb, sampCtrl, etc.)

All these engineering tools can be launched from WS console in order to execute commands on AO system for all or single devices and/or to receive control and diagnostic information during test operations.

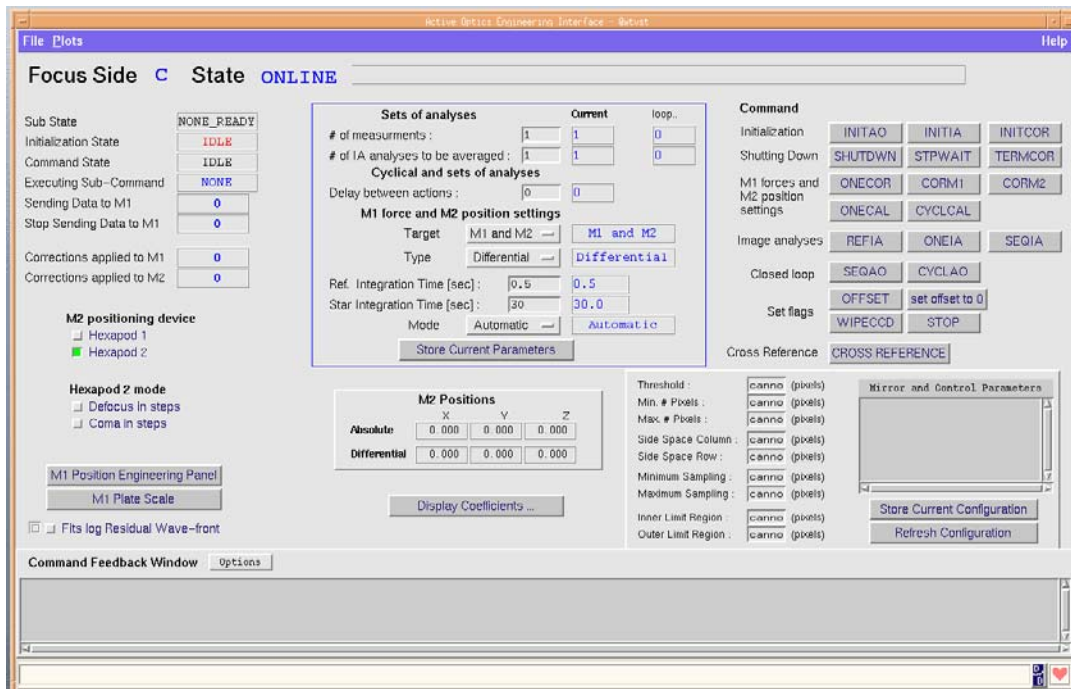


Fig. 4 – the *vstactguiEngineer* engineering user interface

## 5 Definition of tests


As introduced above, there are basically two kind of tests:

Low level device functionalities: motion of pinhole in/out of TCCD image axis, M1 axial and radial actuator force commands and readout, M2 first stage hexapod (called hex1) and M2 sub-micron second stage hexapod (called hex2) positioning commands and readout.

High level AO functionalities: open loop mirror weight balancing control, based on pre-calibrated correction lookup tables at different ALT axis angular positions; closed loop aberration and misalignment correction, based on image analysis, i.e. simulated target vs. reference image comparison through the wavefront sensor.

The procedures, described below, generate a map recording both reference and actual M1 and M2 active device forces/positions, based on specific command/reply sequences sent to the related subsystems. The devices involved are:

- M1 axial actuators
- M1 radial actuators
- M2 hex1

|  |  |   |
|--|--|---|
|  | Project: VST<br>Doc.VST-PRO-OAC-AO-001.DOC | Rev.: 1.0<br>Date: 2006/10/26<br>Pag. 9 of 31 |
|--|--|---|

- M2 hex2
- Pinhole axis
- SH wavefront sensor with TCCD in simulation mode

Regarding the analysis of data, resulting from tests without real observation in sky, in principle the only way to measure the correct positioning of M1 is to compare reference values of forces, issued by commands, with forces read from actuator load cells. This can be easily done because the involved processes store both information in the DB and it is straightforward to redirect these data in input/output log files to be analyzed off-line.

For the analysis of M2 positioning, the actual situation is the following:

- Hexapod nr. 1: the direct and inverse cinematic procedures are well known. By these formulas it is possible to obtain the values of decentering (X-Y plane), defocusing (Z plane) and tilt (rotations around X and Y) directly from the six leg linear motions. It is also easy to reverse the process. So far, by storing both reference and actual linear motions, it is possible to perform the functionality and reliability evaluation after tests, directly using the X,Y,Z,TILT data.
- Hexapod nr. 2: actually only the direct cinematic procedure is well known, i.e. the conversion from X,Y,Z,TILT values to linear motions of six piezo-actuators. The inverse process is not yet available, because it is not easy to calculate and is actually under investigation. Furthermore, actually the reference positions, issued by command, are not stored in the DB, so also this opportunity is under implementation. So far, at the moment it is possible to perform positioning analysis by acquiring linear motion values of piezo-actuators instead of X,Y,Z,TILT coordinates.
- Actually it is not foreseen a combined positioning command of M2 hexapods, i.e. with both hexapods involved in positioning actions at the same time. Basically this functionality should not be used during normal operations of the telescope (see section 5.2.1 of this document). Hence whenever a M2 positioning command is issued, it is necessary to pre-select which device must be involved.
- The hexapod nr. 2 has a restricted positioning range, lower than hexapod nr. 1, so it is better to use it only during AO open loop tests, in order to take under control the correct positioning commands.

## 5.1 M1 Low-Level actuator test

### 5.1.1 Test principle

To test both functional and reliability performances of axial and radial actuators. In particular to analyze:

- communication between LCU and single or groups of axial actuators, through CAN bus
- communication between LCU and single or groups of radial actuators, through CAN bus
- communication between LCU and all axial and radial actuators together
- stability of actuator force cyclic readout
- response to force step commands for single axial actuators
- response to force step commands by sampling groups of close axial actuators

All these tests should be repeated at different ALT axis angular positions (for example minimum operative angle 20°, medium angle 45°, maximum operative angle 90°).

In order to perform these tests, there are two possible WS engineering user panels or Tcl/Tk scripts, interfacing LCU with the actuators through CAN bus. The panels are *vstcangui* and *vstcanTest*. Both allow user to send single commands directly to single or groups of actuators, without the necessity to run high-level AO applications. The main difference between the two panels is that the latter directly offers the facility to run cyclic tests and to save results on WS disk.

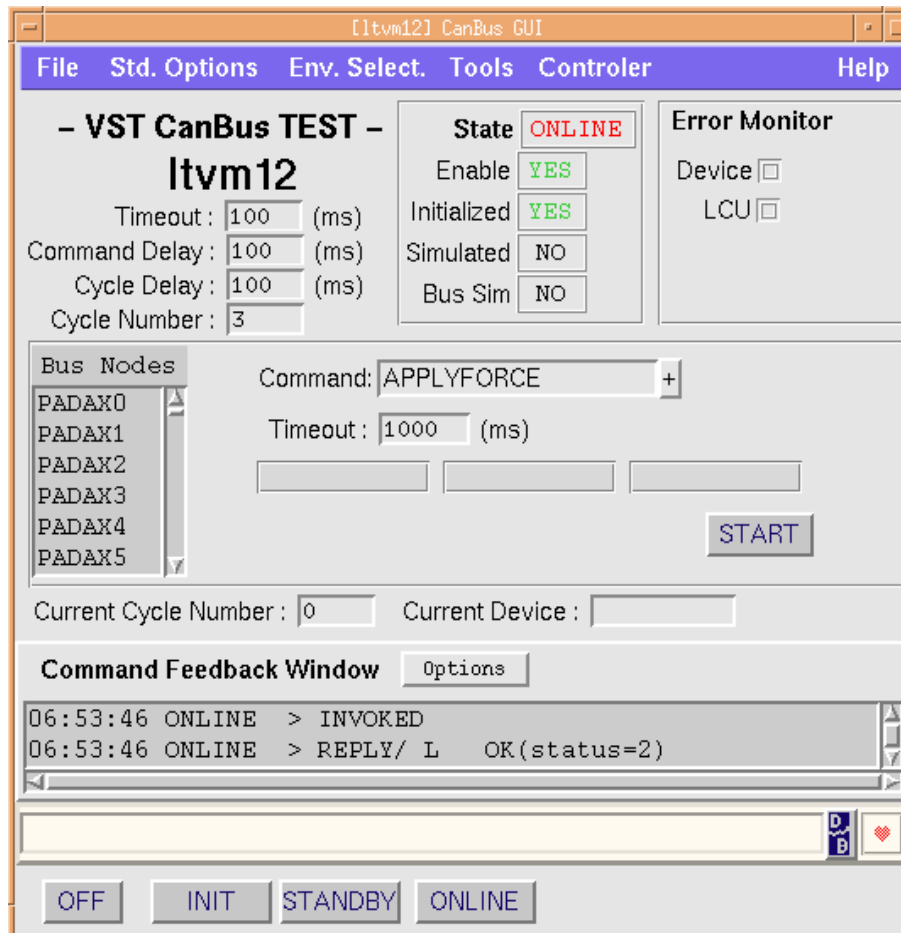


Fig. 5 – the *vstcanTest* engineering user interface

The limitation of using such user interfaces is that it is possible to run cyclic single command sessions, having as feedback the only command status (success/fail), without parameters.

In order to perform force setup commands with the force readout reply from actuators, it can be used a specific Tcl/Tk script, called *vstcanActuatorTest*, able to perform cyclic preset/apply/readout force sequences for single, groups or all actuators. The force commands are sent in values expressed by integers between 0 and 65535 (16-bit resolution), because the script runs at the lower possible level of interface software.

The conversion from these values to forces can be done by the following considerations:

- each axial actuator supports a weight in the range [0, 50] kg, corresponding to  $50 \cdot 9.81 = 490.5$  N. So far, the maximum resolution of 16 bit corresponds to a normalization factor of  $65535/490.5 = 133.61$ . Thus the final formula of force is  $\text{value} / 133.61$  [N]
- -For radial actuators the weight range is [-25, +25] kg with the same resolution. It must be also considered the mechanical demultiplying ratio of 4.56, thus obtaining the final force formula:  $(\text{value}/133.61 - 25 \cdot 9.81) \cdot 4.56$  [N]

### 5.1.2 Collected data

By assuming the use of *vstcanTest* GUI, the following data are collected:

- Report file in ASCII format reporting results after cyclic tests. The results display the command status (succeed or failed) for each command sent to actuators. Here there is an example:

Example of a communication test by sending the command GETFORCE to a group of axial actuators in a cycle of 10 iterations.

**Corresponding test result file:**

```
{VSTCAN TEST RESULT
-----
LCU Environment      : ltvml2
Tested Devices Number: 84
Tested Devices List  : PADAX0 PADAX1 PADAX2 PADAX3 PADAX4 PADAX5 PADAX6 PADAX7
Tested Command       : GETFORCE
Tested parameters    :
```

```
-----
TEST SETTING
-----
```

```
Timeout           : 0ms
Delay between commands: 0ms
Delay between cycles : 0ms
```

```
-----
TEST RESULT
-----
```

```
Start Time : 09-10-2006 09:43:19.
End Time   : 09-10-2006 09:43:30.
Duration   : 11 seconds.
Cycles Done : 10
```

| Device | Num.Cmd | Succeed | Failed |
|--------|---------|---------|--------|
| PADAX0 | 10      | 10      | 0      |
| PADAX1 | 10      | 10      | 0      |
| PADAX2 | 10      | 10      | 0      |
| PADAX3 | 10      | 10      | 0      |
| PADAX4 | 10      | 10      | 0      |
| PADAX5 | 10      | 10      | 0      |
| PADAX6 | 10      | 10      | 0      |
| PADAX7 | 10      | 10      | 0      |

By assuming the use of *vstcanActuatorTest* test script, the following data are collected:

- Force to be applied to selected actuators (in resolution values)
- Force read from load cell of actuator and converted in N (Newton) using above formulas.

By redirecting reference/output data in log files it will be possible to perform the test analysis.

### 5.1.3 Analysis

Data analysis is performed by comparing actual vs. reference force values for all actuators involved in the tests. See Tab. 1

### 5.1.4 Step by step procedure

By using *vstcanTest* the detailed procedure, step by step is the following:

1. activation of ALT axis:
  - a. move ALT to desired position
2. activation of M1 control system:
  - a. launch *vstcanTest*
  - b. select right environment (*ltvm12*)
  - c. verify scan link system is alive (if needed launch *vstBUILDScanLinks ltvm12*)
  - d. switch *vstcanServer* process to ONLINE state. After that the user interface is ready to communicate with the M1 actuator devices
  - e. On the top left side of the GUI it is possible to configure parameters for cyclic tests:
    - i. Timeout: reply timeout on command sent to actuators
    - ii. Command delay: delay between two consecutive commands
    - iii. Cycle delay: delay between two consecutive command cycles
    - iv. Cycle number: number of desired cycles (0 means infinity, i.e. to be manually stopped)
  - f. In the central part of the GUI, the listbox “Bus Nodes”, allows to select single or groups of the actuators (nodes directly controlled by the selected LCU). The actuators are named from PADAX0 to PADAX83 for the axial type and from PADRAD0 to PADRAD23 for radial type.
  - g. After actuator selection, the combo box named “command” allows to select the command to be sent, with related parameters.
  - h. By clicking on “START”, the test cycles start.
  - i. At the end of the cyclic test, an output panel will appear, displaying results. It will then possible to store results on files in ASCII format.
3. goto 1

By using *vstcanActuatorTest*, the detailed procedure, step by step is the following:

1. activation of ALT axis:
  - a. move ALT to desired position
2. activation of M1 control system:
  - a. boot *ltvm12* LCU
  - b. verify scan link system is alive (if needed launch *vstBUILDScanLinks ltvm12*)
  - c. switch *vstcanServer* process to ONLINE state. After that the interface is ready to communicate with the M1 actuator devices
  - d. launch *vstcanActuatorTest MinVal MaxVal Tolerance ActuatorNumList*, where:
    - i. *MinVal*: lower value (in 16-bit resolution) to reach (>7000)
    - ii. *MaxVal*: upper value (in 16-bit resolution) to reach (<57000)
    - iii. *Tolerance*:  $\pm$  accepted range of reached value for *MinVal* and *MaxVal*
    - iv. *ActuatorNumList*: list of actuators to be test
  - e. You can redirect in a log file the output or acquire force values by *SampCtrl*.
3. goto 2.d

In the following test definition matrix, all the procedure items are listed.

| Step no. | Action   | Actual Status / Value  |
|----------|--|--|
| 1        | Record date and time   | Date =<br>Time =   |
| 2        | Record software modules version  | vstcan:<br>vstm1as:<br>vstm1server:<br>vstm1ws:<br>vstBUILD:<br>vstgui:<br>vstmsw: |
| 5        | Record test conditions   | actual ALT axis position:  |
| 6        | Follows procedure in 5.1.4 using data in Tab. 1  |  |
| 7        | <ul style="list-style-type: none"> <li>• Start sampling of data to files</li> <li>• Collect data for desired minutes</li> <li>• Stop sampling of data to files</li> <li>• Write file name to Tab. 1</li> <li>• Move file to /vltuser/aoTests:<br/>mv <i>filename</i> /vltuser/aoTests</li> </ul> |  |
| 8        | If Tab. 1 entries are not finished, go to step no.6  |  |
| 9        | Produce report with the analysis of forces   |  |

## 5.2 M2 Low-Level hexapod test

### 5.2.1 Test principle

The VST secondary mirror needs to be positioned with very high accuracy in terms of x-y-z displacements and rotations around x-y axes. There are two devices foreseen to this purpose:


- Hexapod nr. 1, planned to be used for large excursions and not during the exposure
- Hexapod nr. 2, planned to be used during the exposure for small displacements

The purpose of the tests is a rough evaluation of the performance of hexapods. For each displacement or tilt, two kinds of tests can be performed:

- A ramp response test: the ramp is composed by steps of variable size, in order to investigate the positioning precision after step commands of different amplitudes, and also the absolute correspondence between commanded and executed displacement
- A repeatability test, i.e. displacements or tilts between two fixed positions or angles

The tests should also be performed at different angles of ALT axis.

For references related to accuracy of M2 hexapod positioning before and during the exposures, please refer to [AD14].

|  |  |  |
|--|--|--|
|  | Project: VST<br>Doc.VST-PRO-OAC-AO-001.DOC | Rev.: 1.0<br>Date: 2006/10/26<br>Pag. 14 of 31 |
|--|--|--|

In order to perform these tests, there are two possible WS engineering SW tools, interfacing LCU with the hexapods through CAN bus. The first is *vstcanTest*. It allows user to send single commands directly to single or both hexapods, without the necessity to run high-level AO applications. It also offers the facility to run cyclic tests and to save results on WS disk. The second tool is a specific Tcl/Tk script, *vstm2Test*, running at the lowest level, able as well to perform tests for single or both hexapods.

The data are recorded using sampCtrl sampling tool.

### 5.2.2 Collected data

The following data are collected:

- HEX1 x [mm] (both reference and reached values)
- HEX1 y [mm] (both reference and reached values)
- HEX1 z [mm] (both reference and reached values)
- HEX1 tilt-x [arcsec] (both reference and reached values)
- HEX1 tilt-y [arcsec] (both reference and reached values)
- HEX2 x [ $\mu\text{m}$ ] (reference position only)
- HEX2 y [ $\mu\text{m}$ ] (reference position only)
- HEX2 z [ $\mu\text{m}$ ] (reference position only)
- HEX2 tilt-x [arcsec] (reference position only)
- HEX2 tilt-y [arcsec] (reference position only)
- HEX2 x-leg (both reference and reached values)
- HEX2 y-actuator 1 (both reference and reached values)
- HEX2 y-actuator 2 (both reference and reached values)
- HEX2 z-actuator 1 (both reference and reached values)
- HEX2 z-actuator 2 (both reference and reached values)
- HEX2 z-actuator 3 (both reference and reached values)

### 5.2.3 Analysis

Data analysis is performed by comparing actual vs. reference positions for all subsystems involved in the tests. See Tab. 2

#### 5.2.4 Step by step procedure

By using *vstcanTest* the detailed procedure, step by step is the following:

1. activation of ALT axis:
  - a. move ALT to desired position
2. activation of M2 control system:
  - a. launch *vstcanTest*
  - b. select right environment (*ltvm12*)
  - c. verify scan link system is alive (if needed launch *vstBUILDScanLinks ltvm12*)
  - d. switch *vstcanServer* process to ONLINE state. After that the user interface is ready to communicate with the M2 hexapod devices
  - e. On the top left side of the GUI it is possible to configure parameters for cyclic tests:
    - i. Timeout: reply timeout on command sent
    - ii. Command delay: delay between two consecutive commands
    - iii. Cycle delay: delay between two consecutive command cycles
    - iv. Cycle number: number of desired cycles (0 means infinity, i.e. to be manually stopped)
  - f. In the central part of the GUI, the listbox “Bus Nodes”, allows to select single or groups of nodes (but on M2 no groups are required, because the hexapods are single units) The nodes selectable for M2 are:
    - i. PCHEXA0: the control Hexapod PC
    - ii. HEXAP1: the hexapod nr. 1
    - iii. HEXAP2: the hexapod nr. 2
  - g. After actuator selection, the combo box named “command” allows to select the command to be sent, with related parameters.
  - h. By clicking on “START”, the test cycles starts. Each time the cycle begins with a complete sequence of OFF-INIT-ONLINE commands sent to *vstcanServer* process
  - i. At the end of the cyclic test, an output panel will appear, displaying results. It will then possible to store results on files in ASCII format.
3. goto 1

By using *vstm2Test*, the detailed procedure, step by step is the following:

1. activation of ALT axis:
  - a. move ALT to desired position
2. activation of M2 control system:
  - a. boot *ltvm12* LCU
  - b. verify scan link system is alive (if needed launch *vstBUILDScanLinks ltvm12*)
  - c. switch *vstcanServer* process to ONLINE state. After that the interface is ready to communicate with the M2 hexapod devices
  - d. launch *vstm2Test Hexapod Test inputFile [monitor]*, where:
    - i. Hexapod: device selected (H1 or H2)
    - ii. Test: axis to be tested (X,Y, Z,TILTX, TILTY)
    - iii. inputFile: name of text file containing the sequence of positions to be issued (in mm for HEX1 and  $\mu$ m for HEX2)
    - iv. monitor: optional. No actions if empty. If enabled (value=1), it indicates that the inputFile contains DB paths for each specified position command.
  - e. You can redirect in a log file the output or acquire position values by *SampCtrl*.
3. goto 2.d

In the following test definition matrix, all the procedure items are listed.

| Step no. | Action   | Actual Status / Value   |
|----------|--|---|
| 1        | Record date and time   | Date =<br>Time =  |
| 2        | Record software modules version  | vstcan:<br>vstm2co:<br>vstm2ws:<br>vstBUILD:<br>vstgui:<br>vstmsw:            |
| 5        | Record test conditions   | <ul style="list-style-type: none"> <li>• actual ALT axis position=</li> </ul> |
| 6        | Follows procedure in 5.2.4 using data in Tab. 2  |   |
| 7        | <ul style="list-style-type: none"> <li>• Start sampling of data to files</li> <li>• Stop sampling of data to files</li> <li>• Write file name to Tab. 2</li> <li>• Move file to /vltuser/aoTests:<br/>mv <i>filename</i> /vltuser/aoTests</li> </ul> |   |
| 8        | If Tab. 2 entries are not finished, go to step no.6  |   |
| 9        | Produce report with the analysis of positions  |   |

## 5.3 AO Open Loop test

### 5.3.1 Test principle

As already described, the AO open loop test consist of calibrated corrections, based on pre-calculated aberration tables related to different ALT angles. Open loop thus means no image analysis feedback. In Italy this is the only AO test that can be applied in real mode, with the exception due to the use of empty calibrated tables (these tables must be filled at Paranal with appropriate entries).

The open loop AO test should be repeated at different ALT angles, in order to test functionality and reliability of M1 and M2 control devices. In this configuration only the mirror weight balancing is corrected and not for real aberrations.

Basic WS application involved in AO tests is *vstactguiEngineer*, where all AO commands and conditions can be set and issued. It is also possible to select single or both hexapod devices to involve in current tests.

The data are recorded using sampCtrl sampling tool and/or redirecting them in log files.

### 5.3.2 Collected data

The following data are collected:

- Current ALT angle
- Command and reached positions of M2 hexapods
- Reference and actual forces for axial M1 actuators
- Reference and actual forces for radial M1 actuators

### 5.3.3 Analysis

Data analysis is performed by comparing actual vs. reference positions and forces for all subsystems involved in the tests. See Tab. 3


### 5.3.4 Step by step procedure

The detailed procedure, step by step is the following:

1. activation of main axis control system:
  - a. HBS online
  - b. AZ/ALT/ROT online
2. activation of M1/M2 control system:
  - a. ltm12 booted
  - b. vstcan, vstm2co, vstm2ws, vstm1ws, vstm1as, vstm1Server online
3. start AO activities (through application *vstactguiEngineer*):
  - a. preset ALT axis to a specific angle
  - b. select M1/M2 target devices to test
  - c. select ONECAL to perform one-shot AO calibrated correction
  - d. select CYCLCAL to activate cyclic AO calibrated corrections
  - e. monitor and store on WS (using sampCtrl) positions of M2 and forces for M1
  - f. stop AO cyclic correction when you want
4. goto 3

In the following test definition matrix, all the procedure items are listed.

| Step no. | Action   | Actual Status / Value   |
|----------|--|---|
| 1        | Record date and time   | Date =<br>Time =  |
| 2        | Record software modules version  | vstaxis:<br>vstiklida:<br>vstalaz:<br>vstrot:<br>vstco:<br>vstBUILD:<br>vstIO:<br>vsthb:<br>vstcan:<br>vstm1as:<br>vstm1ws:<br>vstm2co:<br>vstm2ws:<br>vstactcon:<br>vstactcal:<br>vstgui:<br>vstmsw: |
| 5        | Record test conditions   | <ul style="list-style-type: none"> <li>• current ALT angle=</li> </ul>  |
| 6        | Follows procedure in 5.3.4 using data in Tab. 3  |   |
| 7        | <ul style="list-style-type: none"> <li>• Start sampling of data to files</li> <li>• Stop sampling of data to files</li> <li>• Write file name to Tab. 3</li> <li>• Move file to /vltuser/aoTests:<br/><i>mv filename /vltuser/aoTests</i></li> </ul> |   |
| 8        | If Tab. 3 entries are not finished, go to step no.6  |   |
| 9        | Produce report with the analysis of positions and forces   |   |

|  |  |  |
|--|--|--|
|  | Project: VST<br>Doc.VST-PRO-OAC-AO-001.DOC | Rev.: 1.0<br>Date: 2006/10/26<br>Pag. 19 of 31 |
|--|--|--|

## 5.4 AO Closed Loop test

### 5.4.1 Test principle

As already described, the AO closed loop test consist of corrections sent to M1 actuators and M2 hexapods, based on the image analysis process, which consists of a TCCD, a pin hole device and a wavefront sensor controlled by a specific LCU environment (*ltviac*). Closed loop thus means to have as aberration calculation feedback the comparison between reference (obtained by pin hole in the image plan of the TCCD) and current target images. In Italy this AO test mode is in principle not applicable (no real target images can be used). But it is possible to simulate the TCCD operations and to use dummy target images in order to perform the required actions of image analysis.

The closed loop AO test should be repeated at different ALT angles, in order to test functionality and reliability of M1 and M2 control devices.

Basic WS application involved in AO tests is *vstactguiEngineer*, where all AO commands and conditions can be set and issued. It is also possible to select single or both hexapod devices to involve in current tests.

A particular attention must be paid in case of selection of Hex2 device, due to its intrinsic very limited positioning range. So far, in Italy the use of Hex2 is only recommended with AO tests in open loop.

The command CYCLAO, normally foreseen as main cyclic AO closed loop correction command, cannot be used during AO tests in Italy, where image analysis is performed in simulation mode. The reason is that with no real target image acquired during cycles, the comparison is always done between a reference image and the same target image, causing no corrections (actuators and hexapods do not move). It is recommended to perform AO closed loop tests with a different sequence of actions, described in the step-by-step procedure reported below.

The data are recorded using sampCtrl sampling tool and/or redirecting them in log files.

### 5.4.2 Collected data

The following data are collected:

- Current ALT angle
- Pin hole positioning status
- Command and reached positions of M2 hex1
- Reference and actual forces for axial M1 actuators
- Reference and actual forces for radial M1 actuators

### 5.4.3 Analysis

Data analysis is performed by comparing actual vs. reference positions for all subsystems involved in the tests. See Tab. 4

#### 5.4.4 Step by step procedure

The detailed procedure, step by step is the following:

1. activation of main axis control system:
  - a. HBS online
  - b. AZ/ALT/ROT online
2. activation of image analysis system
  - a. Itviac booted
  - b. TCCD in simulation mode
  - c. Itvadc booted
  - d. vstphServer online (pin hole control module)
  - e. provide a set of simulated target object images
3. activation of M1/M2 control system:
  - a. Itvm12 booted
  - b. vstcan, vstm2co, vstm2ws, vstm1ws, vstm1as, vstm1Server online
4. start AO activities (through application *vstactguiEngineer*):
  - a. preset ALT axis to a specific angle
  - b. select M1/M2 target devices to test
  - c. select INITAO command button to initialize image analysis control devices
  - d. select ONECAL to perform one-shot AO calibrated correction
  - e. select REFIA to analyse the reference image (this command automatically put pin hole in the image plane, by sending command REFIN to vstphServer)
  - f. select a dummy image as simulated target image acquired by TCCD
  - g. select SEQIA to analyse the dummy target image (this command automatically put pin hole out of the image plane, by sending command REFOUT to vstphServer)
  - h. select ONECOR to apply corrections to M1 and M2 devices
  - i. goto 4.e to continue with current test session or continue to finish actions
  - j. stop AO correction when you want
5. goto 4

In the following test definition matrix, all the procedure items are listed.

| Step no. | Action  | Actual Status / Value  |
|----------|---|--|
| 1        | Record date and time  | Date =<br>Time =   |
| 2        | Record software modules version   | vstaxis:<br>vstiklida:<br>vstalaz:<br>vstrot:<br>vstco:<br>vstBUILD:<br>vstIO:<br>vsthb:<br>vstcan:<br>vstm1as:<br>vstm1ws:<br>vstm2co:<br>vstm2ws:<br>vstactcon:<br>vstactcal:<br>vstact:<br>vstph:<br>vstgui:<br>vstmsw: |
| 5        | Record test conditions  | <ul style="list-style-type: none"> <li>• actual ALT angle=</li> </ul>  |
| 6        | Follows procedure in 5.4.4 using data in Tab. 4   |  |
| 7        | <ul style="list-style-type: none"> <li>• Start sampling of data to files</li> <li>• Stop sampling of data to files</li> <li>• Write file name to Tab. 4</li> <li>• Move file to /vltuser/aoTests:<br/> mv <i>filename</i> /vltuser/aoTests</li> </ul> |  |
| 8        | If Tab. 4 entries are not finished, go to step no.6   |  |
| 9        | Produce report with the analysis of positions   |  |

## 6 Definition of test matrices

In this chapter test matrices for all test kinds are reported. Basically they represent samples of test maps to be filled during test in Italy. So far, their fields are at the moment intentionally left blank. Their quantities and details will be decided when tests will be performed.

### 6.1 M1 low level test matrix sample

#### LEGENDA

- **ALT pos:** actual ALT axis angular position (in deg)
- **Actuator target:** this field contains details about how many and which actuators are involved in the test.
  - GROUPS: number of actuators used (ax is for axial type; rad is for radial type)
  - ALL: all actuators involved
- **Reference Force Range:** min and max reference force values (in Newton) to be commanded
- **Tolerance:** tolerance (in Newton) applied to min and max reference force commands
- **Command Status:** this field contains details on how many successes or failures have been obtained on commands sent
- **Cycles Nr.:** number of command cycles issued for current test session
- **Output File:** the name and location of output log file
- **Remarks:** comments on particular conditions during tests

| No. | ALT pos [deg] | Actuator Target  | Reference Force Range [N] | Tolerance [N] | Command Status | Cycles Nr. | Output File | Remarks | Executer |
|-----|---------------|--|---------------------------|---------------|----------------|------------|-------------|---------|----------|
| 1   | 90            | GROUPS<br>ax act. nr.<br>ax act. nr.<br>ax act. nr.<br>... | MinVal:<br>MaxVal:        |               |                |            |             |         |          |

|   |    |   |                    |  |  |  |  |  |  |
|---|----|---|--------------------|--|--|--|--|--|--|
|   |    | rad act. nr.<br>rad act. nr.<br>rad act. nr.<br>...   |                    |  |  |  |  |  |  |
| 2 | 45 | GROUPS<br>ax act. nr.<br>ax act. nr.<br>ax act. nr.<br>...<br>rad act. nr.<br>rad act. nr.<br>rad act. nr.<br>... | MinVal:<br>MaxVal: |  |  |  |  |  |  |
| 3 | 20 | GROUPS<br>ax act. nr.<br>ax act. nr.<br>ax act. nr.<br>...<br>rad act. nr.<br>rad act. nr.<br>rad act. nr.<br>... | MinVal:<br>MaxVal: |  |  |  |  |  |  |
| 4 | 90 | ALL   | MinVal:<br>MaxVal: |  |  |  |  |  |  |
| 5 | 45 | ALL   | MinVal:<br>MaxVal: |  |  |  |  |  |  |
| 6 | 20 | ALL   | MinVal:<br>MaxVal: |  |  |  |  |  |  |

**Tab. 1 – M1 low-level actuator test map sample**

**6.2 M2 low level test matrix sample**

**LEGENDA**

- **ALT pos:** actual ALT axis angular position (in deg)
- **Target device:** which device is involved in the test (HEX1 or HEX2).
- **Reference Axis and pos:** reference axis involved and position to be set (mm for X,Y,Z of HEX1,  $\mu\text{m}$  for X,Y,Z of HEX2, arcsec for TILTX and TILTY of both devices)
- **Command Status:** this field contains details on how many successes or failures have been obtained on commands sent
- **Cycles Nr.:** number of command cycles issued for current test session
- **Output File:** the name and location of output log file,
- **Remarks:** comments on particular conditions during tests

| No. | ALT pos [deg] | Target device | Reference Axis and pos  | Command Status | Cycles Nr. | Output File | Remarks | Executer |
|-----|---------------|---------------|---|----------------|------------|-------------|---------|----------|
| 1   | 90            | HEX1          | X – pos =<br>Y – pos =<br>Z – pos =<br>TILTX – pos =<br>TILTY – pos = |                |            |             |         |          |
| 2   | 45            | HEX1          | X – pos =<br>Y – pos =<br>Z – pos =<br>TILTX – pos =<br>TILTY – pos = |                |            |             |         |          |
| 3   | 20            | HEX1          | X – pos =<br>Y – pos =<br>Z – pos =<br>TILTX – pos =                  |                |            |             |         |          |

|   |    |      |   |  |  |  |  |  |
|---|----|------|---|--|--|--|--|--|
|   |    |      | TILTY – pos =   |  |  |  |  |  |
| 4 | 90 | HEX2 | X – pos =<br>Y – pos =<br>Z – pos =<br>TILTX – pos =<br>TILTY – pos = |  |  |  |  |  |
| 5 | 45 | HEX2 | X – pos =<br>Y – pos =<br>Z – pos =<br>TILTX – pos =<br>TILTY – pos = |  |  |  |  |  |
| 6 | 20 | HEX2 | X – pos =<br>Y – pos =<br>Z – pos =<br>TILTX – pos =<br>TILTY – pos = |  |  |  |  |  |

**Tab. 2 – M2 low-level hexapod test map sample**

### 6.3 Active Optics Open Loop test matrix sample

#### LEGENDA

- **ALT pos:** actual ALT axis angular position (in deg)
- **M2 Target device:** which device is involved in the test (HEX1 or HEX2).
- **Reference M1 forces M2 positions:** reference forces for axial and radial actuators and reference positions of M2 hexapod selected. These tables will be stored in an input log file reported in the column entry.
- **Actual M1 forces M2 positions:** reached forces for axial and radial actuators and positions of M2 hexapod selected after command execution. These tables will be stored in an output log file reported in the column entry.
- **Command issued:** this field reports the command issued (ONECAL or CYCLCAL)
- **Remarks:** comments on particular conditions during tests

| No. | ALT pos [deg] | M2 Target device | Reference M1 forces M2 positions <sup>1</sup> | Actual M1 forces M2 positions              | Command issued | Remarks | Executer |
|-----|---------------|------------------|---|--|----------------|---------|----------|
| 1   | 90            | HEX1             | name of input file where info are stored:     | name of output file where info are stored: | ONECAL         |         |          |
| 2   | 90            | HEX1             | name of input file where info are stored:     | name of output file where info are stored: | CYCLCAL        |         |          |
| 3   | 45            | HEX1             | name of input file where info are stored:     | name of output file where info are stored: | ONECAL         |         |          |
| 4   | 45            | HEX1             | name of input file where info are stored:     | name of output file where info are stored: | CYCLCAL        |         |          |
| 5   | 20            | HEX1             | name of input file where                      | name of output file where                  | ONECAL         |         |          |

<sup>1</sup> Actually the reference positions of M2 hex2 piezo-actuators are calculated by vstm2co module, but not stored in the DB. It is under development the implementation of this issue.

|    |    |      | info are stored:                          | info are stored:                           |         |  |  |
|----|----|------|---|--|---------|--|--|
| 6  | 20 | HEX1 | name of input file where info are stored: | name of output file where info are stored: | CYCLCAL |  |  |
| 7  | 90 | HEX2 | name of input file where info are stored: | name of output file where info are stored: | ONECAL  |  |  |
| 8  | 90 | HEX2 | name of input file where info are stored: | name of output file where info are stored: | CYCLCAL |  |  |
| 9  | 45 | HEX2 | name of input file where info are stored: | name of output file where info are stored: | ONECAL  |  |  |
| 10 | 45 | HEX2 | name of input file where info are stored: | name of output file where info are stored: | CYCLCAL |  |  |
| 11 | 20 | HEX2 | name of input file where info are stored: | name of output file where info are stored: | ONECAL  |  |  |
| 12 | 20 | HEX2 | name of input file where info are stored: | name of output file where info are stored: | CYCLCAL |  |  |

**Tab. 3 – Active Optics open loop test map sample**

#### 6.4 Active Optics Closed Loop test matrix sample

##### LEGENDA


- **ALT pos:** actual ALT axis angular position (in deg)
- **M2 Target device:** which device is involved in the test (HEX1 or HEX2).
- **Reference M1 forces M2 positions:** reference forces for axial and radial actuators and reference positions of M2 hexapod selected. These tables will be stored in an input log file reported in the column entry.
- **Actual M1 forces M2 positions:** reached forces for axial and radial actuators and positions of M2 hexapod selected after command execution. These tables will be stored in an output log file reported in the column entry.
- **Command issued:** this field reports the command issued (INITAO, ONECAL, REFIA, SEQIA, ONECOR)
- **Remarks:** comments on particular conditions during tests

| No. | ALT pos [deg] | M2 Target device | Reference M1 forces M2 positions <sup>2</sup> | Actual M1 forces M2 positions              | Command issued           | Remarks  | Executer |
|-----|---------------|------------------|---|--|--------------------------|--|----------|
|     |               |                  |   |  | INITAO                   |  |          |
| 1   | 90            | HEX1             | name of input file where info are stored:     | name of output file where info are stored: | ONECAL                   |  |          |
| 2   | 90            | HEX1             | name of input file where info are stored:     | name of output file where info are stored: | REFIA<br>SEQIA<br>ONECOR | these commands are sent two times, with dummy target image changed between first and second time |          |
| 3   | 45            | HEX1             | name of input file where info are stored:     | name of output file where info are stored: | ONECAL                   |  |          |

<sup>2</sup> Actually the reference positions of M2 hex2 piezo-actuators are calculated by vstm2co module, but not stored in the DB. It is under development the implementation of this issue.

|   |    |      |   |  |                          |  |
|---|----|------|---|--|--------------------------|--|
| 4 | 45 | HEX1 | name of input file where info are stored: | name of output file where info are stored: | REFIA<br>SEQIA<br>ONECOR | these commands are sent two times, with dummy target image changed between first and second time |
| 5 | 20 | HEX1 | name of input file where info are stored: | name of output file where info are stored: | ONECAL                   |  |
| 6 | 20 | HEX1 | name of input file where info are stored: | name of output file where info are stored: | REFIA<br>SEQIA<br>ONECOR | these commands are sent two times, with dummy target image changed between first and second time |

**Tab. 4 – Active Optics closed loop test map sample**

|  |  |  |
|--|--|--|
|  | Project: VST<br>Doc.VST-PRO-OAC-AO-001.DOC | Rev.: 1.0<br>Date: 2006/10/26<br>Pag. 30 of 31 |
|--|--|--|

## 7 Test results

This section will be filled after tests.

**\_\_oOo\_\_**