AO Software Testing - I
Engineering tests with low level interfaces

Lorenzo Busoni, Luca Fini, Alfio Puglisi, Marco Xompero

CAN: 486f010a

INAF - Osservatorio di Arcetri
http://adopt.arcetri.astro.it/

ABSTRACT

This report contains a brief review of the AO Supervisor software testing performed in Arcetri in June 2009.
The testing activity reported here refers to the management of each individual hardware device by means of the related control programs and low level GUIs.

Contents

Glossary of terms and acronyms

AdSec. The Adaptive Secondary Mirror. In this context usually refers to the group of *AO-Supervisor* components controlling the hardware devices related to the secondary mirror.

ADAM. Ethernet controlled digital output. Used to enable various devices within the AdSec.

AdSec Computer. The workstation running the *AO-Supervisor* components which control the Adaptive Secondary hardware.

Adsec-Arb. The Adaptive Secondary Arbitrator. A component of the *AO-Supervisor* which executes commands related to the *AdSec* coordinating the operations of the hardware devices in the Adaptive Secondary. Commands to *AdSec-Arb* may come either from a specific GUI or from *AO-Arb*.

AO System. The hardware and software components of the LBT first light Adaptive Optics System. Includes the Wavefront Sensor, the Adaptive Secondary Mirror, the *AO Computer* and some auxiliary devices (such as networking hardware) and includes the *AO-Supervisor* and the real-time software.

AO-Arb. The AO Arbitrator. A component of the *AO-Supervisor* which manages the execution of high-level commands, coordinating the operations of *WFS-Arb* and the *AdSec-Arb*. Commands to *AO-Arb* may come either from a specific interface or from *AOS*.

AO Computer. The computer (or farm of computers) running the *AO-Supervisor*.

AO Console. The operator console of the *AO Computer*.

AO-Supervisor. The software system which manages all the components of the *AO System*

BCU. Basic Control Unit. Electronics board used as basic building block for most of the electronics in the AO System (see [1]).

BCU 47. The BCU used as frame grabber for the CCD 47.

C-BCU. Crate BCU. The six BCUs which controls the AdSec.

CCD 39. The CCD used for the Wavefront sensor.

CCD 47. The CCD used for the *TV*.

Copley. Motor driver for the Bayside stages.

Fastlink. The real-time data communication link between the WFS and the AdSec.

FLAO. First Light AO system for the LBT. The whole AO system for the LBT, including both hardware and software components.

Flowerpot. Auxiliary unit to control the calibration source and the related optics (cube beam splitter).

MsgD. Message Dispatcher, the *AO-Supervisor* message dispatching daemon.

RTDB. AO Real Time Database, the *AO-Supervisor* own variable repository. Its functionalities are supported by `MsgD`.

S-BCU. Switch BCU. The BCU operating as inpout source switch for the AdSec.

TTM. Tip-Tilt Mirror. A small mirror used to modulate the pupil image un top of the WFS pyramid.

TV. Technical Viewer. An auxiliary CCD camera used by the Wavefront Sensor to acquire the reference star.

WFS. The Wavefront Sensor. In this context usually refers to the software subsystem controlling the hardware devices related to the wavefront sensor.

WFS CCD. The CCD used in the *WFS* to measure the light wavefront deformation (also: *CCD 39*).

WFS Computer. The workstation running the *AO-Supervisor* components which control the Wavefront Sensor hardware.

WFS-Arb. The WFS Arbitrator. A component of the *AO-Supervisor* which executes commands related to the *WFS* coordinating the operation of the hardware devices of the WFS. Commands to *WFS-Arb* may come either from a specific GUI or from *AO-Arb*.

1 Introduction

This report describes the test activities related to the AO Supervisor software used as a standalone system to control the hardware devices which are part of the FLAO.

The testing activity is aimed to verify that all the hardware components of the system can be safely and effectively operated in the solar tower lab by means of the related software components. In this respect it must be clear that a successful test implies both the verification of the proper functioning of the software and the verification that the expected behavior and capabilities are observed in the hardware devices.

The test is divided into two levels:

- **Single device operations**. Where all devices are operated in a standalone mode in order to exercise all the capabilities of each device.
- **Coordinated operations**. Where more complex tasks which imply the concurrent operations of different devices are tested.

2 Single device operations

2.1 Common components

2.1.1 MirrorCtrl

This controller manages the ethernet communications with all the BCUs. An instance of this process must reside on each workstation used for the AO System (currently two). Each instance operates all the related BCUs.

The name of the process is related to the mirror for historical reasons, but it is used both for the WFS and for the AdSec

Its main task is the control of all write and read requests to/from the unit. As an auxiliary task `MirrorCtrl` may operate as a watchdog for a set of devices by periodically sending messages to BCUs (see [2], par. 3, par. 8.3.3.a,b).

Because it is used by any controller which must communicate with a BCU it is indirectly tested as a result of the other tests described in the following sections.

2.2 WFS components

Here follows a brief description of tests performed on individual software components of the WFS. Both user interfaces and device controllers are included.

Note that most of the controllers have individual watchdog mechanisms implemented by periodically polling the related device electronics either via ethernet or via the serial line, depending on the specific communication technology used for each. The testing is done by severing the communication line and verifying that the interruption is detected within the proper time frame.

2.2.1 chUI

Command line based user interface. It may be used for simple tasks instead of any GUI, and allows to write scripts to perform complex tasks.

As a basis for all the scripts used for the control of WFS devices it is indirectly tested in some of the individual tests described below.

2.2.2 WFS Engineering GUI

This is the low level GUI to operate all the devices included in the WFS. It is essentially a collection of all the individual GUIs for the WFS devices. Each sub-GUI is related to a specific device and allows to operate it. Most of the tests described below are performed by means of these GUIs.

2.2.3 CCD Viewer

Specific GUI to display the CCD images. Three instances of this process are used, respectively, to display images from the CCD 39, the CCD 47 and to display slopes. The following tests have been performed:

1. Binning 1,2,3,4, for CCD 39 and 1,2,4,16 for CCD 47.
2. Automatic switch of binning in the display when CCD binning is changed
3. Slopes display. This is tested by applying an artificial tilt by adding bias to a sector of the CCD.

2.2.4 SlopeComputerCtrl

This controller configures and operates the Slope Computer BCU. The following items have been tested:

1. BCU watchdog functionality.
2. Pixel lookup tables. Tested by generating a known pattern.
3. Background subtraction. Tested by replicating the slope computation algorithms in IDL and applying on the same set of input data.
4. Slopes calculation. Tested as above.
5. Slopes transmission to the AdSec. Tested by reading back values from the AdSec BCU.

Tests have been repeated to verify the capability to support binning at 1,2,3 and 4 pixels.

2.2.5 TechnicalViewerCtrl

This controller configures and operates the Technical Viewer BCU. The following items have been tested:

1. CCD controller watchdog.
2. Pixel lookup tables.

Tests have been repeated to verify the capability to support binning at 1,2,4 and 16 pixels.

2.2.6 RelayCtrl

This controller operates the main power switch, box fans switches and reset/program switches of BCUs and manages the 10 temperature sensors in the two electronic boxes. It also configures the over temperature threshold for emergency stop. The following items have been tested:

1. Watchdog mechanism
2. Each individual switch operated from the GUI. The feedback value is then checked in the RTDB.
3. For each temperature sensor measures have been recorded while stopping the cooling system.
4. The cooling system was stopped until the over temperature hardware mechanism stopped the system.

2.2.7 PowerCtrl

This controller operates the on/off switches (operated by BCU) of most other devices (stages, motors, CCDs, etc.). The following item have been tested:

1. BCU watchdog mechanism
2. Each individual switch operated from the GUI. The feedback value is then checked in the RTDB.

2.2.8 PicCtrl

This controller operates the on/off switches of flowerpot devices: source switch, and two switches for the beam splitter, a dimmer for the source and reads two temperature/humidity sensors placed close to the CCD 39 (to evaluate the dew point). The following items have been tested:

1. Watchdog mechanism
2. Cube beam splitter switches.
3. Source switch. Tested by inspecting CCD images.
4. Source dimmer. Tested as above.
5. Temperature readouts.

2.2.9 JoeCtrl

This controller manages the configuration and operations of both the CCD 39 and and CCD 47. Two instances of the same controller are used. The following items have been tested for both instances:

1. Serial watchdog mechanism
2. Set binning
3. Set Readout speed
4. Set frame rate.
5. Set bias
6. Start/Stop integration.
7. Temperature readout.

2.2.10 TTCtrl

This controller manages (via BCU 47) the analog waveforms generator for the TTM. The following items have been tested:

1. Watchdog mechanism
2. Set frequency (both free frequency and locked to CCD framerate)
3. Set modulation amplitude
4. Set X-Y offset.

Correctness of results have been optically verified by using an auxiliary CCD.

2.2.11 SimpleMotorCtrl

The AO Supervisor includes seven instances of this controller. Each instance controls the position of a single motor. The controlled devices are: Filter wheel 1 & 2, ADC motors 1 & 2, Pupil rerotator motor, Cube rotator motor, cube stage motor. The following items have been tested:

1. Watchdog mechanism
2. Position calibration. Verified with direct measures.
3. Positioning control. Verified with direct measures.
4. Homing repeatability. Verified with external measure
5. Limit switch on/off. Verified from feedback variables in RTDB.
6. Configurable movement speed. Movement speed variation verified by visual inspection.

2.2.12 CopleyCtrl

Three instances of this controller manage the three Copley drivers for the XYZ stages. The following items have been tested:

1. Watchdog mechanism
2. Position control
3. Homing
4. Brake
5. Limit switch on/off
6. Temperature and current readout
7. High precision movement (0.1 microns)

2.2.13 CameraLensCtrl

This controller operates the analog lines for the piezoelectric camera lens stage. Because the operation of the stage depends on previous operations performed by `TTCtrl`, a check of the correct functioning of the latter must be performed. The following items have been tested:

1. TTCtrl up and running
2. Camera lens movement (optically verified)

2.2.14 WFS MirrorCtrl

See section 2.1.1.

2.3 AdSec components

2.3.1 AdSec MirrorCtrl

For a general description see section 2.1.1.

This process in the AdSec incarnation is also used as watchdog mechanism. The following items have been tested:

1. ADAM watchdog
2. C-BCU watchdog
3. S-BCU watchdog

2.3.2 Housekeeper

This controller has the purpose to retrieve slow rate diagnostic data from the AdSec and check the health of the mirror. The set of monitored variables is fully described in [1], par. 17.2, and in [3], par. 5.2 and par 7.2.

When a potentially dangerous condition is detected, the process must initiate proper emergency procedures (see [3]) and provide visual and audible alert in the Housekeeper GUI.

The following items have been tested:

1. Read out of all system temperatures (cooling and electronics)
2. Read out of cooling flux-meter
3. Read out of all status flags
4. Read out of system voltages and currents (both at the crate level and at the actuator level)

5. Safety procedures for temperatures out of ranges (call `FunctEmergencyTemp`)
6. Safety procedures for no cooling flux (call `FunctEmergencyFlux` – TBD)
7. Safety procedures for dew point alarm (call `FunctEmergencyDew` – TBD)

Because it is not always possible and/or safe to generate the dangerous conditions in the controlled hardware, the tests are performed by dedicated IDL procedures which simulates the conditions by modifying the warning and alarm thresholds for each controlled variable.

2.3.3 MasterDiagnostic

This process collects diagnostic data from many sources (actuator positions and currents, slopes, data from accelerometers and some counters) and organizes them for further analysis.

The following items have been tested:

1. Data collection at 350 Hz. No data loss.
2. Data collection at 1000 Hz. With some data loss. *Solutions to be investigated*[1].

2.3.4 OptLoopDiagnostic

This process extracts from data sets gathered by `MasterDiagnostic` a subset of data needed to evaluate quality parameters of the optical loop. Extracted data are stored on disk for subsequent analysis.

During the tests the correctness of stored data sets have been verified.

2.3.5 FastDiagnostic

This process has the purpose to analyze diagnostic data collected by `MasterDiagnostic` and to compute mirror health parameters derived from fast changing data (capacitive sensor positions, actuator currents, etc.). The content of diagnostic frames are described in [1], par. 17.3.5. The following items have been tested:

1. Diagnostics computation at 80 Hz.
2. Positions out of ranges (call `FunctDisableCoils`)
3. Coil currents out of ranges (call `FunctDisableCoils`)
4. Safety skip frames detection (call `FunctNotifySkip`, TBD)

Note: the same process will provide information needed for mode offload. This part is currently in development.

[1]Problems related with data loss at 1KHz are not directly dependent on software issues. From the point of view of software testing, this test is to be considered as successful.

2.3.6 IDLCtrl

This process is a wrapper encapsulating the IDL based procedures which manage the AdSec. Each command defined corresponds to an IDL procedure which is executed.

The following items have been tested:

1. Correct standard input and output redirection
2. Correct routing from AdSec-Arb to IDL function and parameters.

A "Dummy" configuration has been defined which allows to test the correct routing of commands from AO-Arb to `IDLCtrl`.

2.3.7 InterferometerCtrl

This process controls the 4D interferometer. The following items have been tested:

1. Remote Python object control for acquiring single measures
2. Remote Python object control for acquiring burst measures

3 Coordinated operations

This part of the testing activity has the purpose to asses the capability of the AO Supervisor modules to coordinate the operations of many devices in order to perform a complex task.

In this set of tests coordinated tasks are not managed by automatic procedures but are performed by skillful operators controlling the two main parts of the system (i.e.: the WFS and the AdSec) by mean of the low level GUIs of the various controllers.

The following procedures have been successfully tested:

1. Adaptive loop closed
   (a) All needed WFS devices are switched on.
   (b) The AdSec is switched on and set to default precalibrated shape.
   (c) The WFS is re-centered and the focus is adjusted. The effect is verified optically.
   (d) The precalibrated reconstructor matrix and gain vector are sent to the AdSec.
   (e) The AdSec is set in wait mode.
   (f) The close loop command is sent to WFS

   Tests have been repeated with various conditions up to a loop closed at 1 KHz with 600 modes controlled.

2. Reconstructor matrix calibration
   (a) As above
   (b) As above
   (c) As above
   (d) The calibration procedure is started. It consists in applying a known mirror shape history while recording the corresponding set of slopes. The reconstructor is then computed off-line and stored in a FITS file to be used.

4 Conclusions

The results from testing activities reported in the above pages demonstrated the capability of the software components tested to provide all the functionalities required in any phase of the development of the FLAO system, from the managing of individual hardware devices, to the verification of the functionalities of the devices, to the coordinated operations of the AO subsystems of which the tested devices are components.

The tested components which are the basic builkding blocks of the AO SUpervisor software, are thus well capable to support all the required functionalities.

References

[1] Roberto Biasi, Mario Andrighettoni, and Daniele Veronese. LBT672 Design Report: Electronics. Technical Report 641a006, Microgate, May 2008.

[2] Armando Riccardi, Marco Xompero, and Daniela Zanotti. LBT672a acceptance test: performance and high-level safety. Technical Report 640f016, INAF-Arcetri, March 2008.

[3] Roberto Biasi, Mario Andrighettoni, and Daniele Veronese. Adaptive Secondary Control System. Control System RT Software User Manual. Technical Report 641a022, Microgate, May 2008.

Doc_info_start
Title:
Document Type: Specification
Source: Osservatorio di Arcetri
Issued by: L. Busoni, L. Fini, A. Puglisi, M. Xompero
Date_of_Issue: 26 June 2009
Revised by:
Date_of_Revision:
Checked by:
Date_of_Check:
Accepted by:
Date_of_Acceptance:
Released by: L. Fini
Date_of_Release:
File Type: PDF
Local Name:
Category: 400
Sub-Category: 480
Assembly: 486 Adaptive Optics Software
Sub-Assembly:
Part Name:
CAN Designation: 486f010
Revision: a
Doc_info_end