GTCAO Software Control

J. Rosich*[a], J. Marco de la Rosa[a], J.A. Rodríguez Losada[b], O. Casanova [c], O. Tubío [a],
M. Núñez Cagigal [a], J.C. López[a], L.F. Rodríguez-Ramos[a], J. Patrón[a], R. L. López [a], M.
Puga Antolín [a], I. Montilla [a], R. Simoes [a], V.J.S. Béjar [a] , M. Reyes [a]

[a] Instituto de Astrofísica de Canarias, E-38200, La Laguna (S.C. Tenerife), [b] Gran
Telescopio de Canarias S.A. E-38200, La Laguna (S.C. Tenerife), [c] Wooptix S.L., E-
38200, La Laguna (S.C. Tenerife), SPAIN

ABSTRACT

The GTC Adaptive Optics is a classic ground layer post focal system that will initially use natural guide stars. The project is in manufacturing phase with several subsystems already integrated. This article shows an overview of the GTCAO mechanisms control software in charge of commanding the many mechanical devices, lamps and sensors present on the optical bench. The software is developed following the rules defined by GRANTECAN for its instruments, in order to be able to be integrated in the GTC Control System.

Keywords: GTCAO, GTC, GCS, adaptive optics, control system, software

1. OVERVIEW OF THE GTCAO INSTRUMENT

GTCAO is a post focal system situated on the Nasmyth platform B, which corrects the optical beam to feed the scientific instrument placed after it. GTCAO follows the classic design of an AO system with the use of two identical off-axis parabolas, maintaining the effective focal distance of the telescope. On Day 1, the system will provide a single deformable mirror conjugated to the telescope pupil and will use natural stars (NGS) for wavefront sensing.

The GTCAO system is expected to provide a corrected beam that will achieve a Strehl ratio of 0.65 in K-band with bright guide stars. The size of the transmitted field of view is 1.5-arcmin diameter and the optical layout of the system includes an atmospheric dispersion corrector (ADC) working up to 60º zenith angle in order not to degrade its performance with increasing zenith angle.
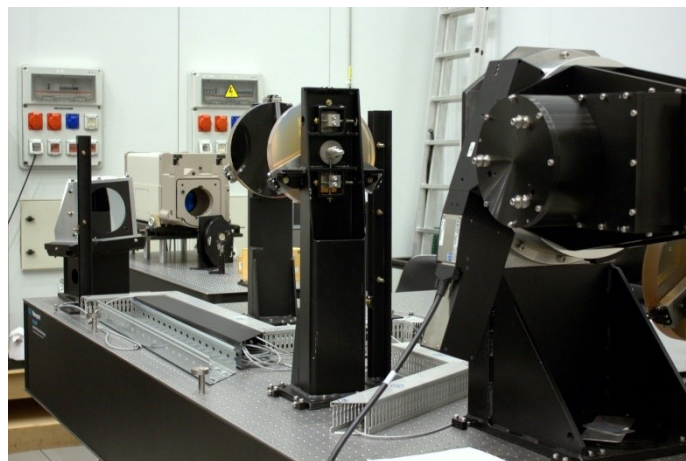

Figure 1. GTCAO in AIV tests.

2. BRIEF DESCRIPTION OF GCS

The control system of the GTC (GCS) operates the telescope and its instrumentation as a single entity. It is responsible for the operation and any computational needs of the GTC and consists of the necessary software and hardware to provide control and monitoring of the GTC, with the aim of optimally managing the scientific observations, as well as the automatic execution of them.

The GCS architecture has four layers:

-   Application Layer includes the subsystems, classes and packages which represent application-specific behaviour. This layer provides the final applications the users will see and that ultimately drive the GTC. The following application packages belong to this layer: *Observing Workshop* (containing the Inspector, the Observing Proposal Management Tool, and the Archive Management Tool), the *Operation Co-ordination Applications* (Data Factory, Sequencer, Scheduler, Observing Engine and others) and the *Equipment Control and Monitoring* (the M1 control system, M2 control system, the Main Axes control system, Science Instrument and others).

-   Domain-specific Layer includes the subsystems, classes and packages that represent model elements specific to the domain, but not limited to this particular application. This layer provides a set of services specialized for a telescope control system. Such services, in turn, are mainly built using the Domain-Independent Layer services. The following services have been placed in this layer: *GUI toolkit*, *data processing toolkit*, *control toolkit*, distributed application framework (*DAF*) and data access layer (*DAL*).

-   Domain-independent Layer provides a set of services, both platform-independent (i.e. provides insolation from the System Services level) and non-specific for a telescope control system. The following services are in this layer: distributed middleware (CORBA), GUI library, scripting, help tools, data processing libraries.

-   System Service Layer expresses the system services layer, subsystems, classes and packages that represent implementation-environment specific model elements. They provide hardware and operating system level facilities, i.e. the operating system calls, RTOS specifics, Java Virtual Machine and the hardware device drivers.
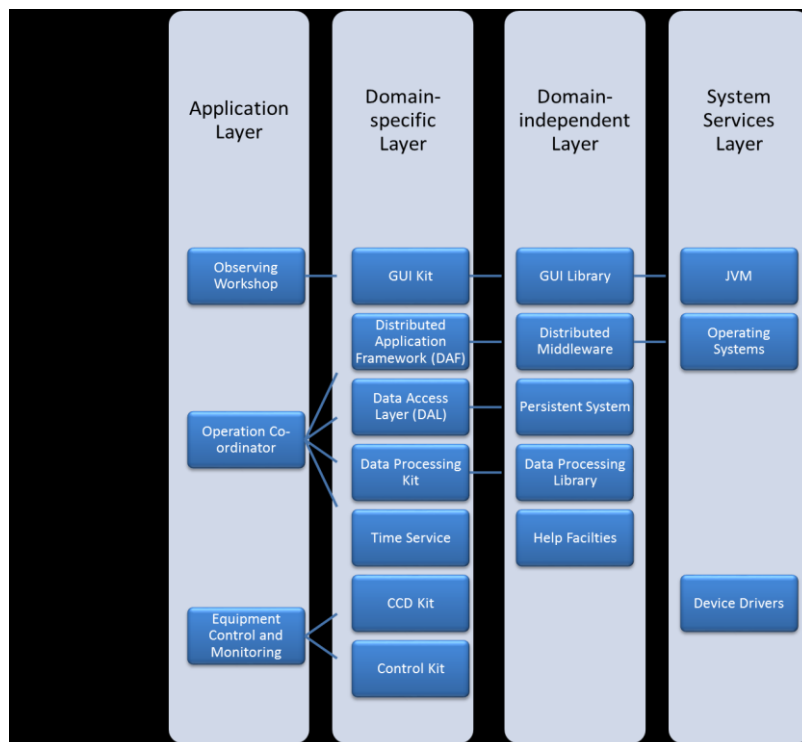

Figure 2. GCS Packages distributed in its four Layers.

3. GTCAO MECHANISMS CONTROL SYSTEM

GTCAO is composed of optical elements, which are mounted to and controlled by mechanical devices. These mechanisms will be actuated and monitored by the control software that is included in the package Science Instrument Control System that is found in the Application Layer of GCS. We identify the following subsystems, which contain both optical and electro-mechanical elements:

- Shutter
- Temperature sensors
- Light Sources
- Insertion mechanism: science ADC, focal plane unit and the Phase Screens
- Wheels: aperture wheel, filter wheel, lenslet array wheel and test camera filter wheel
- WFS Pupil positioner
- Laser lens focus mechanism
- ADC rotation mechanism: WFS and WFC ADCs
- K-system rotation mechanism
- WFS Positioner: consists on three axes
- Deformable Mirror (DM)
- WFS Camera
- Test Camera

From a software point of view, these components are independent but they share some common aspects due to the hardware used and the GCS architecture. Let us see the main classes, interfaces and instances grouped by its hardware.

3.1 Adam Controller Structure

There is an Adamlib library that consists on wrapping the low-level Modbus/TCP commands to the high level, which corresponds, to GCS components. Eight classes compose the Adam Library:

- AdamBase class that is the interface to all the ADAM family functionalities.
- AdamBaseHW and AdamBaseSim where it is implemented the real hardware control or simulator.
- ModbusManager class that packs and wraps the libmodbus (low-level library that implements the Modbus/TCP protocol in Linux) functions, providing capabilities as throwing exceptions when errors are detected.
- Adam6054, Adam6024 and Adam6060. An AdamBase object is instantiated as a private object, but only actually supported methods are used. In this way, each Adam6xxx class particularize the generic functionalities provided by Adam Base.

In addition, in the high level it is developed the following GCS components that inherit from GCS's Device class to integrate the GTCAO software into GCS, and also have the availability of common services such as logging, alarms, configuration and monitoring services.

- GTCAOTempMonitor that is responsible for reading temperature of several hardware components, such as deformable mirror or natural guide star simulator. This component interfaces an Adam 6015 controller.
- GTCAODimmableLights that is responsible for controlling two dimmable lamps of GTCAO calibration system. This component interfaces an Adam 6024 controller.
- GTCAOOCam2Control that is responsible for switching on/off the Ocam2 camera and monitoring its temperature, humidity, coolant flow and compressed air. This component interfaces an Adam 6024 controller.
- GTCAOShutter that is responsible for opening and closing the shutter located at the entrance of the instrument. This component interfaces an Adam 6060 controller.
- GTCAOSwitchableLights that is responsible for switching on/off the three lamps of GTCAO calibration system. This component interfaces an Adam 6060 controller.
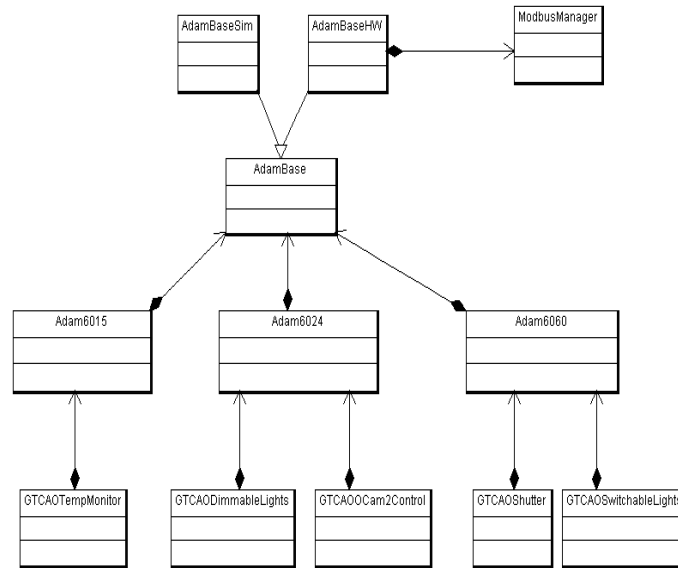
Figure 3. Adam controller structure.

3.2 CANOpen Controller Structure

Any controller mechanism based on a CANOpen can be implemented in the same way in the majority of aspects, such as controlling the engine power, the referenced, the parking, the stop, etc.... Only those positioning methods with different parameters, linear or rotary, require a specific interface and conversions. However, it is necessary to specify the type of motor controller used because when it is wanted to do tracking each one has its way of implementation. For example, the Servostar700 has a limitation in the interpolated mode because it has no position buffer. Due to that, it has been chosen implementing a speed control loop on the controller itself. While the IDM680 controller uses the interpolated mode in a time-sharing system.

The "Actuator List" class encapsulates the management of more than one drive associated to one axis, considering that operations are applicable to master drive and which to the slave drives. In the case of GTCAO always have the configuration of a single drive, and it is treated as a special case of the list, with one master drive and no slave drive.

When developing a new mechanism, on the one hand, it must inherit from the interface specific to CANOpenNMotorsController and secondly adding the corresponding actuator (Servostar700Drive or IDM680Drive), then all the implemented behavior from this generic structure is inherited and the specific functionality is added in this class.

For the GTCAOCS, mechanisms that use CANOpen are defined as follows:

- GTCAORotation: Rotary mechanism that uses the IDM680Drive. This class represents the objects Phase screens wheels.
- GTCAOIndexedWheel: GTCAORotation specialization that adds the functionality of choosing predefined positions. This class represents the objects Filter wheels, Apertures wheel, Lenslet wheel and Test Filter wheel.
- GTCAORotatorK: GTCAORotation specialization that adds the functionality of the AxisInterface in order to OE can operate with it.
- GTCAOAdc: Is composed by two GTCAORotation that have to synchronize them.
- GTCAOWFSAxis: Linear mechanism that uses the Servostar700.
- GTCAOWFSPositioner: Is composed by three GTCAOWFSAxis with the functionality of AxisInterface in order to OE can operate with it.
- GTCAOLinear: Linear mechanism that uses the IDM680Drive. This class represents the object focus laser lens.
- GTCAOInsertion: Linear mechanism that uses the IDM680Drive that only has two position insert/retract. This class represents the objects WFC ADC insertion, Phase screens insertion and focal plane insertion.
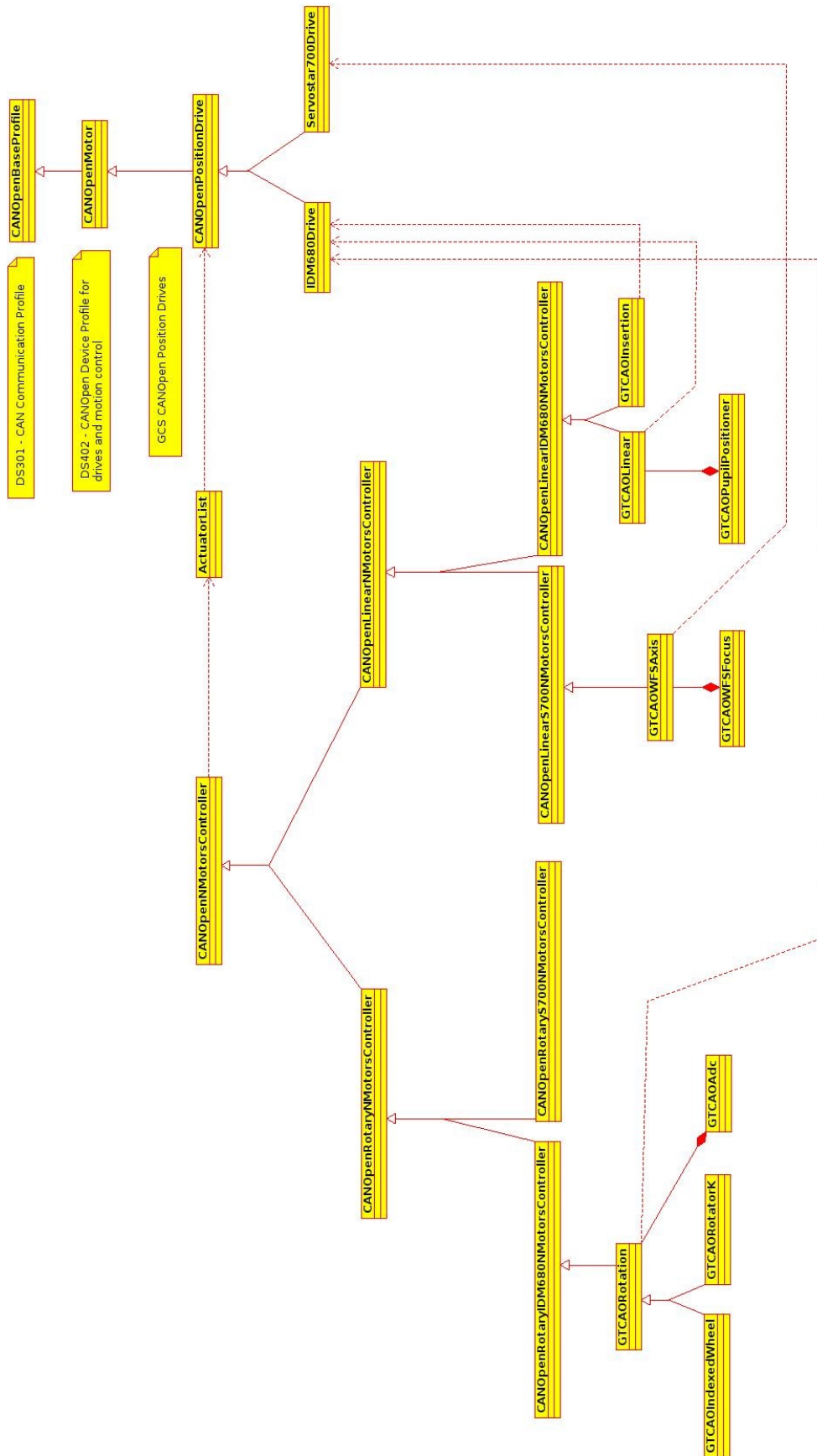- GTCAOPupilPositioner: Is composed by two GTCAOLinear.

Figure 4. CANOpen controller structure.
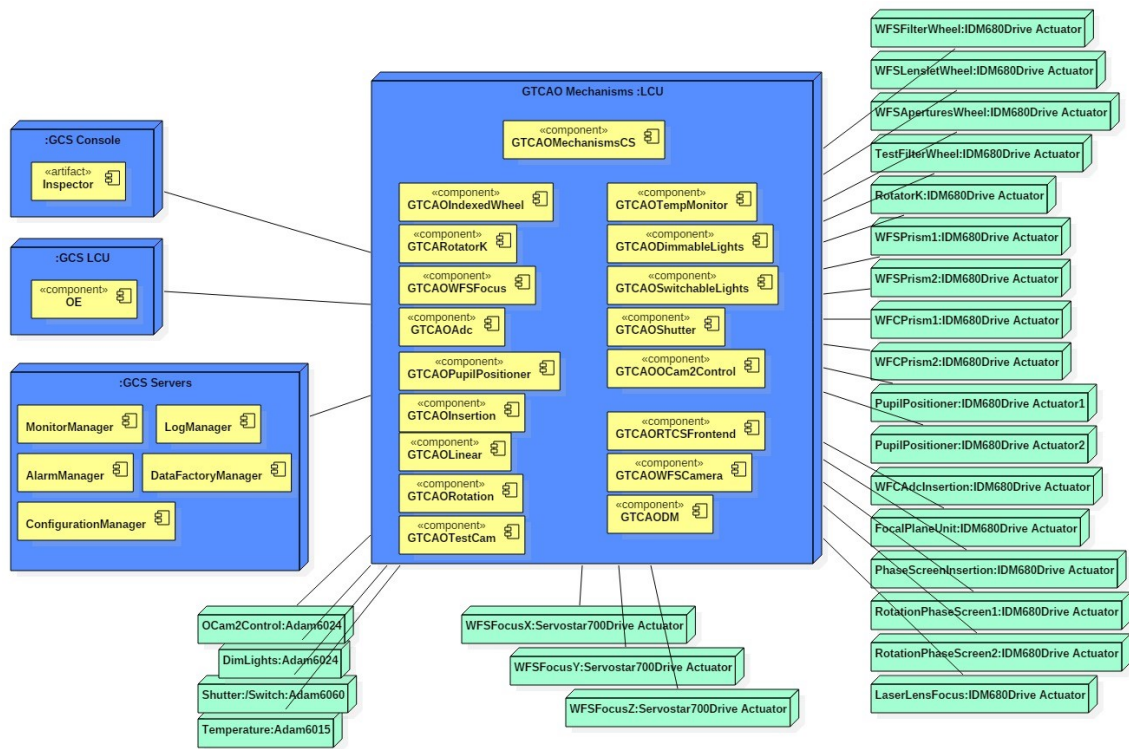
3.3 Mechanisms LCU Deployment

Figure 5. GTCAO Mechanisms deployment.

This figure shows the nodes of the GCS (consoles, servers and LCUs) and which components can be executed in each node. The GTCAO Mechanism Control System (MCS) and the GTCAO Real Time Controller run in separate LCUs. Here is the deployment of Mechanisms LCU.

4. GTCAO INSTRUMENT LAYER CONTROL SYSTEM

The GTCAO Instrument Layer is the part of the GTCAO Control System responsible for the coordination of the different mechanisms of the instrument and real time control system in order to correct images from atmospheric aberrations. The GTCAO Instrument Layer is part of the Sequencer and it is included in the Operation Co-ordination Applications GCS package. For this reason the GTCAO ILCS architecture is restricted by GCS architecture and we have to follow some GTC guidelines and inherit some models.

GTCAO Instrument Layer is in charge of translating the high level commands, into smaller operations understandable by these subsystems, sending them to the mentioned subsystems, ordering the concurrent and sequential execution of the operations between the different GCS Subsystems involved, and finally returning the results to the emitter of the command. This is the way to write sequences that are employed for describing Observation Modes.
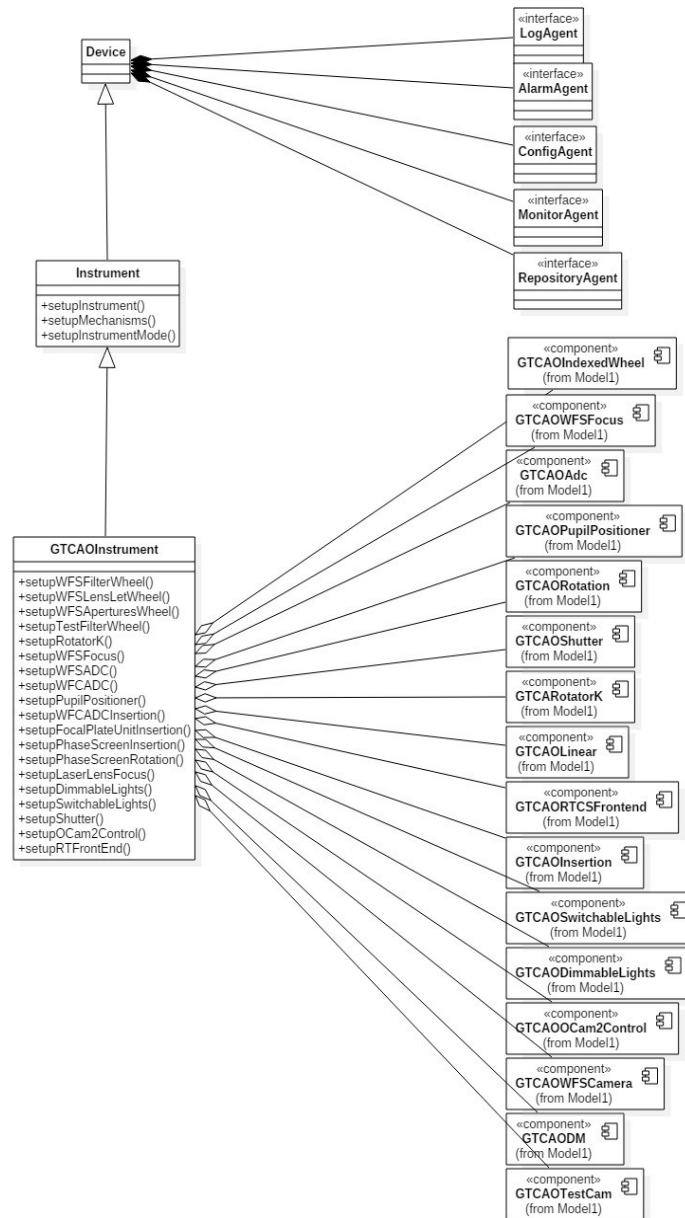
Figure 6. GTCAO Instrument Layer Control System Logical View.

The Observation Mode defines the operations that will be performed on the received data. The GTCAO has the following Observing Modes:

- Closed-loop observation mode with the GTCAO calibration mode
- Open-loop observation mode with the GTCAO calibration mode
- Target acquisition on the WFS
- GTCAO on-sky closed-loop mode
- GTCAO on-sky open-loop mode

In addition, the TestCam Observing Modes will be:

- Target acquisition
- Stare image

5. ACKNOWLEDGEMENTS

REFERENCES

[1] Hardy, J.W., "Adaptive Optics for Astronomical Telescopes", Oxford University Press, 1998.

[2] "L10: Adaptive Optics." [Online]. Available: http://slittlefair.staff.shef.ac.uk/teaching/phy217/lectures/telescopes/L10/.

[3] Pi I Puig, M., "AOCS. Use Case Model Survey" (2011).

[4] Filgueira, J.M., Peñataro, R., Pi i Puig, M., González Bergés, M., Gomez Cambronero, P., Osinde, J., Suárez, M., "GTC Control System Software Architecture" (2000).

[5] Rodríguez Losada, J.A. "Folded Cassegrain. Software Architecture" (2016).

[6] Casanova González, O., "Software Components for Advantech ADAM Devices Involved in the Mechanism Control System" (2015).

[7] Philippe Kruchten, "Architectural Blueprints—The "4+1" View Model of Software Architecture", IEEE Software 12 (6) Nov 1995, pp. 42-50.