Mem. S.A.It. Suppl. Vol. 9, 469
© SAIt 2006

Memorie della
Supplementi

The active optics software for the VST telescope

P. Schipani , M. Brescia , L. Marty, and F. Perrotta

Istituto Nazionale di Astrofisica – Osservatorio Astronomico di Capodimonte, Salita Moiariello 16, I-80131 Napoli, Italy e-mail: `schipani@na.astro.it`

Abstract. The VST (VLT Survey Telescope) active optics software must basically provide the analysis of the image coming from the wavefront sensor (a 10x10 subpupils Shack Hartmann device) and the calculation of primary mirror forces and secondary mirror displacements to correct the aberrations of the optical system, instrinsic or originated for thermal and gravity reasons. After the telescope commissioning the VST will be operated by ESO. In this framework, INAF-OAC staff was committed to design and realize the software in a VLT-compliant way. This will smoothen the integration, operation and maintenance of the telescope in the Paranal observatory.

Key words. Active Optics - Image Analysis - VST - Telescope Control Software

1. Architecture

The active optics software is a part of the whole VST telescope control software package (Schipani et al. (2001), Schipani et al. (2004)). The network architecture of the active optics system involves different workstations and Local Control Units (LCU). The workstations are HP-Ux 11 based, while the LCUs are VME crate equipped with Motorola MVME2604 boards PowerPc boards, running VxWorks 5.4 operating system. The code is developed in C and C++, using Tcl-Tk for the GUIs as in the ESO VLT scheme. The active optics control architecture is shown in Fig.1. The programming environment is provided by ESO through the releases of the VLT Common Software. Programming practices must follow the standards set by ESO that define programming style, naming conventions for software objects, files, etc., directory structure for software modules, standard Makefile for module compilation and installation in integration environments.

2. Simulation code

Since the M1 and M2 mirrors and the overall active optics system were not physically available during the software development phase and so tests with the real system were impossible, the only validation of the computation of aberrations, forces and positions could come from a parallel simulation system. Therefore each key computation has been independently implemented in Matlab and compared with the results coming from the HP workstation + LCUs system. This has led to a complete toolbox for the simulation of an active optics system. It was developed in the framework of the VST project, but is general enough to be useful for many other active optics system. The toolbox deals with both off-line and on-line activities: it generates fictitious Shack-Hartmann fits images, supports different kinds

---

Send offprint requests to: P. Schipani

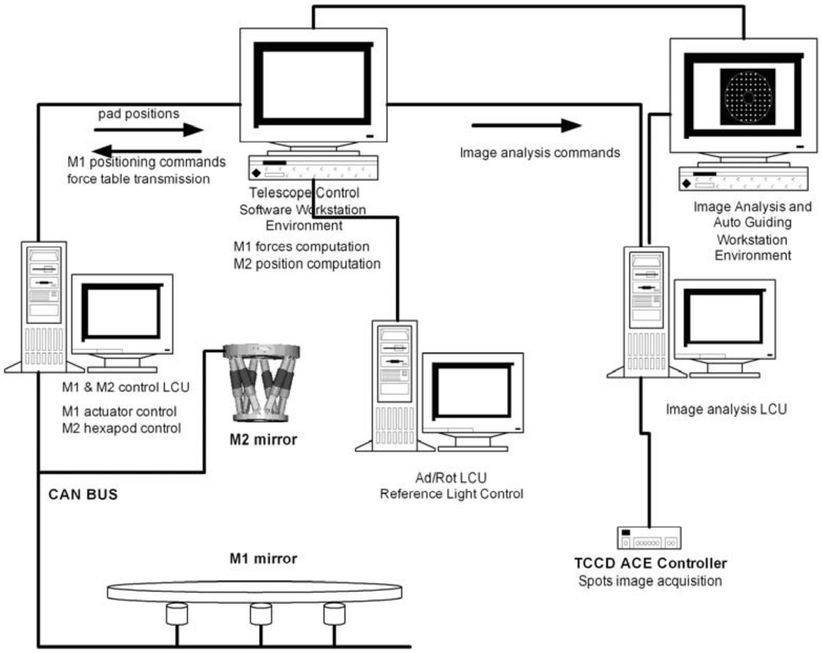

Fig. 1. The VST active optics control software runs in a distributed architecture. The image analysis LCU estimates the incoming wavefront through the Shack-Hartmann array of spots and computes the aberrations; the image analysis workstation environment provides real-time display capabilities; the telescope control workstation environment computes the forces to apply to the M1 actuators and the new position of M2 in order to compensate for the measured aberrations; the M1/M2 LCU computes the positions of the M2 single leg actuators implementing the kinematics of the hexapod device and communicates with the distributed M1 and M2 controllers through a dedicated CAN field-bus; the adapter/rotator LCU controls the reference light needed to produce Shack-Hartmann reference images. The OmegaCam consortium will provide its own image analysis system based on a curvature sensor. This work deals just with the VST image analysis system.

of polynomials (Zernike, quasi-Zernike, normalized Zernike, minimum energy modes) to fit the wavefront data, and performs calculation of primary mirror actuator forces and secondary mirror position corrections.

3. Implementation

The official VST code for the active optics system can be logically divided in a computational part and in an actuation one. The actuation part is spread in a network of distributed controllers: the control loops for the forces of the M1 actuators and the positions of the M2 mirror are closed in these local controllers, described elsewhere (Mancini et al. (2001)). What is reported in the following is related to the whole system but the distributed controllers.

3.1. Computational part

The computational part of the active optics code requires a deep knowledge of the active optics computation theory as described in Noethe (2002) and Noethe (1991). The computations have been checked against the results of the simulation toolbox whenever possible. In the VST, as in the VLT, the main computations are shared between LCUs and WS. The image analysis LCU re-builds the wave-front from the Shack-Hartmann images and produces the aberration coefficients of the modes used in VST. They represent just some low order aberrations, dominant in telescopes with an aspect ratio similar to the VST one (e.g. NTT). It is remarkable that unlike the Zernike polynomials, the elastic modes used in VST depend on the specific mirror. So they must be computed off-line to fill the database with the correct parameters to fit the wave-front. Of course the calibration forces to compensate the measured aberrations depend on the VST modes and must be computed off-line too. The M1 actuator forces and the M2 positions are calculated on the WS, where the measured aberrations are converted from the probe system (polar coordinates, because of the VST specific physical characteristics of the probe arm) to the M1 reference system and the field aberrations are subtracted from the measurement.

3.2. Commmunication with distributed controllers

The actuation part in the VST software resides on the M1/M2 LCU. The LCU modules receive the commands from the WS (basically M1 forces and M2 position settings) and communicate to the local distributed controllers via a CAN field-bus. Of course, since the hardware to be controlled and the bus used for the communications are different, the LCU code is completely different from the VLT one. Nevertheless, in order to maintain the compatibility with the WS control processes, the interfaces are the same. Another remarkable VST specific solution is the control of the secondary mirror by an hexapod based system. The computation of the parallel device leg lengths corresponding to a given amount of displacement of the secondary mirror (e.g. a rotation around the center of curvature to correct the coma) depends of course on the M2 support system.

3.3. Graphical User Interfaces

The whole VST telescope control software (TCS), including the active optics software, is designed to behave as a slave of the Observation Software as usual in the ESO software concept. Therefore the whole TCS is designed to work without the intervention of the operator, simply reacting to external commands. Nevertheless for engineering purposes it is necessary to work directly on the TCS system; so engineering GUIs are provided wherever needed. The VST GUIs have the same look of the VLT ones, in order to hidden to the user the underlying differences in the two systems.

References

Mancini, D., Schipani, P., Mazzola, G., Marty, L., Brescia, M., Cortecchia. F., Perrotta, F., Rossi, E. 2001, Proceedings of ICALEPCS 2001, 179

Noethe, L. 1991, Journal Of Modern Optics, 38, 1043

Noethe, L. 2002, Progress in Optics, 43, 3

Schipani, P., Brescia, M., Mancini, D., Marty, L., Spirito, G. 2001, Proceedings of ICALEPCS 2001, 179

Schipani, P., Brescia, M., Mancini, D., Marty, L., Spirito, G. 2004, Proceedings of ADASS XIII, 697