The VST Axes Control Software

M. Brescia, P. Schipani, L. Marty

INAF – Capodimonte Astronomical Observatory
Via Moiariello 16, I-80131 Napoli, Italy
http://www.na.astro.it

ABSTRACT

One of the tightest requirements to be respected for a telescope as the VST, hosted in one of the best astronomical sites as the ESO Paranal Observatory, is an excellent axes control, to obtain the best overall performance of the telescope that, otherwise, can be dramatically affected. The software strategy to control the VST axes (azimuth, altitude, rotator) is here described.

Keywords: azimuth, altitude, rotator, tracking, servo loop, control software, telescope

1. INTRODUCTION

The VST (VLT Survey Telescope) is a 2.61 m alt-az f/5.5 modified Ritchey-Chretien telescope. It is provided by one focus station in Cassegrain optical configuration, where there will be installed the 16Kx16K OmegaCAM imaging camera. The telescope is going to be installed at Cerro Paranal (Chile) as a powerful survey instrument for the ESO VLT. It is a joint project between the INAF - Osservatorio Astronomico di Capodimonte (OAC) and ESO. OAC has been committed to design and install the telescope, software included, but after the installation and commissioning VST will be managed by ESO. Therefore, in order to simplify the future maintenance of the software by ESO staff, it has been agreed to develop the VST software in the most “VLT-compliant” way. This constraint translates into a series of restricted requirements by OAC side, because many software utilities developed for VLT are available, but their migration to VST case requires a deep knowledge of all details, usually known to the original software developers only. In this context a reverse engineering approach, combined with the traditional software development for the VST specific subsystems, is the only solution.

High image quality performance is guaranteed by an active control of the optics, acting on both primary and secondary mirror. But in order to reach high tracking stability, a very careful approach must be also achieved for both azimuth (AZ), altitude (ALT) and rotator (ROT) axes motion control. Based on their past experience applied in many other telescope projects, the authors investigated a dual design and implementation approach, consisting in the development of an engineering control software tool used to optimize main axes control performances, finally porting the related software solutions into the official telescope control software released at the commissioning stage. This paper deals both with technological aspects and software engineering design and development strategies.

2. ARCHITECTURE

A key role in the control system design has been the basic requirement imposed by ESO to include the VST in the VLT instrumentation control environment working at Paranal Observatory. As the VLT is a large project involving many staff people for many years, a strong standardization has been introduced by ESO in the software development and hardware platforms, [3]. The VST software is therefore based on the same standard architecture, [2]: distributed workstations running HP-UX and VME-bus-based Local Control Units (LCU) running the VxWorks real time multitasking operating system, connected together via Local Area Networks. LCUs essentially play basic role as the real time interface with field electronics and electro-mechanics; the applications they run is a low level software, used to control and monitor the hardware devices, modularly distributed in processes and machines. At the other side, the workstations coordinate LCUs and run the Graphical User Interface (GUI) application software.
The workstation software has time execution requirements less urgent than the LCU’s one, and is the one the operators usually interface with. The basic ideas in the architecture are modularity and standardization. Each LCU is devoted to control a specific subsystem, so the hardware devices are driven independently by different LCUs, using whenever possible the same VME boards and the same software. The programming languages on HP-UX platforms are basically C++ for the core control modules, and Tcl/Tk for GUI development. The LCU modules are written in C. A key role in both workstation and LCU software architecture is played by an on-line Database, which allows the fast mechanism of information sharing between modules, monitoring the whole status of the system and of command execution. Between the operating system and the application layer there is another software layer developed and distributed by ESO for VLT and of course re-used for VST: the VLT Common Software. It provides a lot of utilities, tools and common services to the application programmer.

3. THEORETICAL DESIGN ASPECTS

VLT software has been made available to Capodimonte software staff, which was committed to re-use it as much as possible. This has been certainly an advantage because a lot of software parts could be recycled for VST. But obviously many other code sections have been replaced (basically most of the LCU software) or partially modified, [1]. It is well known that modifying a source code written by other people is often harder than rewriting it from scratch: thus a reverse engineering approach has been mandatory to let the old code run in the different (for telescope hardware, no. of machines, telescope functionalities) VST environment, to understand its operational aspects in terms of simple functionality or in deeper details when needed, to change small or large parts preserving the interlacing between the many applications, running on many different machines. Sometimes a lot of time has been necessary to change just a few lines of code. ESO assistance in this hard work solved only partially the problem, because generally the help in this kind of work is fully effective only if persons work in the same location and can be easily grouped around the same table (otherwise, even a trivial – for the original developer only! – problem can take a long time to be solved).
4. AXIS CONTROL SOFTWARE

4.1. FIRST STEP: A SPECIFIC ENGINEERING CONTROL SOFTWARE

Concerning the VST software team approach applied to the ALT-AZ-ROT control software design, it has been decided to realize as first step an engineering software tool from scratch, in order to be able to perform initial tests on main axes during the telescope integration phase, maintaining a complete control of actions from both hardware and software points of view. Basic requirement was, in fact, the capability to move, in the best secure and reliable way, single axis with several hardware configuration and in different control conditions, for example in single speed loop with one or two motors, during first hardware-software integration steps. This kind of software would be able to test single axis hardware subsystems, such as initialization and tuning of encoder signal, tuning of torque applied to single motors or the optimization of speed and position servo loop control filter parameters.

The main axes drive system engineering software is composed by two modules useful to test all axis drive system functionalities. From the user interface point of view it is basically composed by a series of Tcl/Tk panels directly interfacing the LCU module functionalities. It is possible to configure control filters and other drive system parameters, individually or in subsets, to activate axis startup/shutdown sequences and to move the single axis in position, speed and torque open/closed servo loops.

![Axis Control Engineering Panel](image)

The panel consists of following parts:

- axis LCU process control sub panel, always on top view, giving the possibility to handle the axis control state machine, open/close the speed loop, set a specific speed reference value if speed loop is in open state, and run safety stop and zero torque commands;
- axis motor selection sub panel, always on top view, giving the possibility to select how many and which motors are to be chosen for the current test/setup. In case of ROT axis, the choice will be forced by default to.
two preloaded motors, while in the AZ/ALT case it is possible to select between one, two or four motor configurations. The choice of three motors is considered an error because motors are coupled in two pairs;

- LCU process action log sub panel, always on top, displaying command and action log status at run-time;
- Axis setup/diagnostic panel, containing a series of sub panels selectable from its menu, referred to position/speed setup configuration and telemetry/diagnostic reports;
- speed, position and preload loop control widgets, containing fields to configure all control filter parameters and gains;
- widgets to enable/disable run-time graphical monitoring panels where the selected DB variables show their values along time, with a pre-selected refresh time frequency. This graphical monitoring is useful to have an immediate view of any axis control parameter status during the current test session;

4.2. SECOND STEP: THE SOFTWARE REVERSE ENGINEERING IMPLEMENTATION BREAKDOWN

As previously underlined, many efforts have been spent by VST software team to investigate the VLT telescope control software (TCS) architecture and to discover the optimal strategy to interface this big amount of software with specific software, designed and developed from scratch, controlling the VST subsystems. In particular a mix between VST and VLT control servo system solutions has been carefully considered, obtaining the control software strategy that can be summarized in the following.

The VST TCS follow the VLT TCS standard implementation that foresees a modular architecture where each functionality is composed by a set of software modules and related processes running on different LCUs and WS encapsulated in self-consistent environments. The ALT-AZ control subsystem is enclosed in a subset called Axis Control Module (ACM). The basic purpose of the ACM is to drive the main axes during tracking. For this reason the ACM was implemented as a set of software modules, each of them devoted to a particular axis control functionality. The ACM is composed by the following software modules:

- acm: a software module containing all common definitions used by other ACM modules. Also the interface between them is also defined here. In particular, acm does not contain code nor processes running during run time on the LCUs.
- axis: this module contains the control logic to move an axis using a position and digital speed controller. It also provides a command interface for all actions needed to start, stop and move an axis in tracking or absolute mode.
- iklida: this module is providing all necessary interface functions with the encoder subsystem in order to read, initialize and calibrate the encoder heads.
- altaz: this module is providing the access to the drive subsystem, the interlocks, vicinity limit switches, monitoring signals and all the other hardware components of the axis.

In order to be compliant with this software scenario, the VST reverse engineering process has been based on the complete re-use of first three original VLT TCS modules, described above, restricting the design and implementation from scratch of the last module, renamed “vstalaz”, requiring all the interfaces to the VST specific hardware. Referring to the most careful aspect of control scheme, the integration between VST and VLT TCS software functionalities has been performed following the criteria described below.

The design of the ACM assumes following characteristic: the controlled system requires a desired value (astrometry reference position) which correspond to the target object position and provides a measured feedback signal corresponding to the actual telescope position. The sequenced position and speed controllers are both software units, executed periodically as part of the axis servo loop with a fixed sampling frequency of 500 Hz, Fig. 4-2, whereas the position references are delivered to the trk tracking module (external to the ACM but running on the same LCU), from the tracking coordination module trkws, running on the control workstation, with a typical frequency of 20 Hz. Therefore an interpolation is required that calculates the position set-points between two sequential references. The position controller is internally based on a software-type control filter switching system that, depending on fixed thresholds, stored in the database, determines the type of control filter setup applied in order to minimize the current servo loop R.M.S. error and to set the appropriate speed and hence the corresponding torque to be applied to the motors. The schematic view of such a controller is therefore based on the following algorithm, [4], [5]:


SPIE USE: ____ DB Check, ____ Prod Check, Notes:
if R.M.S. servo loop error is included between 0 and first threshold, then a basic Proportional-Integral (PI) filter is applied
else if the servo loop error is between first and second threshold, then a polynomial filter is applied
else if the servo loop error is higher than the second threshold, a $k_b \cdot \sqrt{\text{error}}$ based filter is applied

**Fig. 4-2** the ALT-AZ servo control system block diagram (ESO courtesy)

**Fig. 4-3** the axis position control working areas (ESO courtesy)

The digital speed controller is running with the same sampling frequency as the position controller and it is based on a PI-type software controller with several configurable low-pass and notch filters in series, while an hardware tachometer system represents its feedback reference mechanism.

4.3. PRELIMINARY PERFORMANCE TESTS

During first integration of the telescope, a preliminary performance test session has been done on main axes control systems, in a “degraded” mechanical configuration (telescope only partially mounted). As basic requirements, [6], included in the error budget approved at the project Final Design Review stage, in terms of main axis tracking positioning error, there must be considered 0.05 arcsec R.M.S. for ALT and AZ, while for ROT axis, the error can be much more relaxed up to more than 5 arcsec RMS, by considering the ratio between ROT angular displacement and its effect on the image plane, taking into account also the telescope instrument scale of 0.21 arcsec/pixel. As can be seen through following tracking test reports (Fig. 4-4 to Fig. 4-9), actually the three axis performances can be considered basically aligned with theoretical requirements. This is extremely encouraging for final performance results on site, where the telescope will be tested fully mounted and equipped.
Fig. 4-4 the AZ axis current position during tracking test session

Fig. 4-5 the AZ axis current speed during tracking test session
Fig. 4-6 the ALT axis current position during tracking test session

Fig. 4-7 the ALT axis current speed during tracking test session
Fig. 4-8 the ROT axis current position and speed during tracking test session

Fig. 4-9 the ROT axis position RMS error during tracking test session
5. CONCLUSIONS

At the moment, most of the software is written and debugged and it is ready for performance tests with the telescope finally integrated. Since this software interface with many external devices, it is foreseen a period of intensive tests with real hardware in order to let the overall control system (software + electronics) working at its best. After the next summer, when the preliminary acceptance in Europe will be completed, the telescope will be shipped in Chile at the VLT site and integrated in its dome together with optics and focal plane instrumentation. Here, after a commissioning phase of several months done in collaboration between ESO and VST team, it will be released to the scientific worldwide community for normal use.

6. ACKNOWLEDGMENTS

We would like to thank the whole VST project staff, M. Capaccioli, G. Sedmak, D. Mancini and the TWG group for helpful and daily support during all telescope design and development phases. We are also grateful to ESO staff, in particular J. Spyromilio and S. Sandrock, for providing fundamental information at several points.

7. REFERENCES

4. VLT-SPE-ESO-11320-0763, Alt and Az LCU control Sw Func Spec, Issue 1.0, ESO VLT Technical Documentation, Garching (Munich), 1994
5. VLT-SPE-ESO-17130-1210, Axis Control Module Design description, Issue 2.0, ESO VLT Technical Documentation, Garching (Munich), 2001

__oOo__