

## BRINGING OLD TELESCOPES TO A NEW ROBOTIC LIFE

V. Suc<sup>1,2,3</sup>, S. Ropert<sup>3</sup>, A. Jordan<sup>1,3,4</sup>, and S. Royo<sup>2,3</sup>

### ABSTRACT

We present the development from its origin to the implementation of new hardware and software for telescope control. This system has been designed to be adaptable to old decommissioned telescopes. We will show how we applied a low-cost system to refurbish three telescopes belonging to the Pontificia Universidad Católica de Chile to allow their remote and automatic use with a web-based interface.

### RESUMEN

Presentamos el desarrollo desde su origen a la implementación de un nuevo hardware y software de control de telescopios. Este sistema puede ser utilizado para renovar equipos antiguos en desuso. Asimismo mostramos cómo un sistema a bajo costo pudo ser aplicado para renovar tres telescopios de la Pontificia Universidad Católica de Chile para permitir su uso remoto y automático mediante una interfaz web.

*Key Words:* telescopes

### 1. INTRODUCTION

Because of the continuous need and presence of an operator, institutions progressively decommissioned telescopes installed in the decades of the 60's to 90's, and set them away from their initial goal. Additionally, their hardware and software being obsolete, a significant proportion of these projects finally got to be merely abandoned because the development cost involved in upgrading their software is usually prohibitive or at least larger than purchasing new equipment.

However, some of these devices have been donated to universities and various institutions for outreach and teaching purpose. Typically these institutions could revive such instruments by adapting commercial kits which were initially designed for amateur astronomers and astrophotographers with hazardous success. From all the late experiences, developing a new Telescope Control System (TCS) easy to use and to implement on a wide variety of existing telescopes, appears like a primary need in the current scenario.

### 2. A NEW TELESCOPE CONTROL SYSTEM

Within an academical context, we were challenged to design and build a TCS for controlling a telescope with academic and scientific purposes. Our first considerations were that for a system intended for use by inexperienced students, the interface should be straightforward and easily controllable over the Internet. Additionally, the system should be precise enough to be tracking sidereal and non-sidereal objects for a period of 3 to 5 minutes without using autoguiding, keeping the image quality mostly limited by seeing. This system should also be kept compatible with existing commercial softwares, from which we considered the two leading families of widely distributed protocols already existing, the Internet Neutral Distributed Interface (INDI), for Linux based system, and ASCOM, for Windows ones. The last notable feature we kept in mind was that the complete TCS implementation cost should remain in the same order of magnitude as the typical commercial systems.

#### 2.1. Characteristics

With this set of specifications in mind, we developed a new Linux based TCS in which the computing power could be distributed on a set of Single Board Computers (SBC) interacting together. The first versions were implemented splitting the control system tasks into three well known Raspberry Pi single board computers (SBC). In this configuration, two SBCs were in charge of the axes speed control loop (one for the Hour Angle and one for the declination), while the third computer was running

<sup>1</sup>Instituto de Astrofísica, Facultad de Física, Pontificia Universidad Católica de Chile, Av. Vicuña Mackenna 4860, 7820436 Macul, Santiago, Chile.

<sup>2</sup>Centre for the Development of Instruments, Sensors and Systems, Universitat Politècnica de Catalunya, Rambla Sant Nebridi 10 E08222 Terrassa, Spain.

<sup>3</sup>ObsTech SpA, Nueva Providencia 1881 of 1620, Providencia, Santiago, Chile.

<sup>4</sup>Millennium Institute of Astrophysics, Av. Vicuña Mackenna 4860, 7820436 Macul, Santiago, Chile.

the position control, pointing models and the Graphical User Interfaces (GUIs). The GUI is a web-based interface which permits to operate the telescope remotely and from any OS.

### 3. THE SANTA MARTINA EXPERIENCE

We acquired our first experience with the new TCS with the Pontificia Universidad Católica de Chile (PUC) in their teaching observatory (ODUC), also named as Observatorio Santa Martina, in the outskirts of Santiago de Chile. The ODUc is located at long.=70:32:04, lat.=33:16:09 and 1450m above sea level, at about 22 km from the center of Santiago, Chile. Even if it is mostly a teaching purpose observatory, most of its equipment components are frequently used for scientific projects carried out by students despite the high light pollution from Santiago. Mean seeing conditions are close to 2 arc-seconds and the observatory benefits of around 150 clear nights per year.

ODUC's main telescopes are: a 40cm donated by Cerro Tololo Inter-American Observatory (CTIO), renamed then as PUC40, and a 50cm (ESO50) donated by ESO-LaSilla. When these telescopes were donated, their telescope control were included and outdated and needed to be upgraded.

#### 3.1. PUC40 Telescope

The PUC40 telescope is a 16" Ritchey-Chretien Telescope with an f/19.5 aperture equipped with a Sbig STL1001 commercial front-illuminated CCD. The square 24mm sensor gives a total field of 8.6'x8.6' and the 24 $\mu$ m pixels achieve a size on sky of 0.62 arc-seconds per pixel which is a suitable sampling for the available seeing on site. The telescope itself was built in 1963 and is hold by a Boller & Chivens mount which was renovated in 1980. This telescope was the first telescope installed at the Kitt Peak National Observatory (KNPO). One of its main tasks in there was doing measurements for the Landolt Variable Stars Catalog. Then it was transported into the CTIO where it got an upgrade.

#### 3.2. ESO50 Telescope

The ESO50 telescope is a 50cm Diameter Cassegrain at f/15 telescope on a fork mount. It was first installed at ESO-La Silla Observatory in 1969 and its operation ended in 1997. It was later donated to PUC in 2002 and translated to ODUc. At first, the control of the telescope was manually driven by an operator but a first automatization was performed in 1993, while it was still in use at La Silla. Only parts of the system were still operational when it was transported to Santiago.

A first upgrade was performed by Baffico et al. (2008). It kept most of the existing hardware from the 1993 upgrade and integrated new PIC Microcontrollers to act on the motor's speed. Communication with the control computer was handled with an RS232 port using the LX200 protocol. The observers used compatible commercial software to control the telescope.

The second software upgrade, performed by Shen et al. (2012), allowed the system a safer operation by monitoring the status of the drivers and integrating the system into the Internet Communication Engine (ICE). Even if this operation could make the interaction with the instruments, under development at the time, more flexible, the obsolete low-level hardware still in use did not allow a robust operation of the complete system.

#### 3.3. Initial Goal

The goal of Santa Martina is to give the students with no previous observational experience a first technical approach to Telescope Operations. As a consequence, the control system needed to be particularly simple and safe to prevent misunderstandings and possibly wrong or dangerous orders from the operator.

In order to give suitable observational results with a simple acquisition procedure, the telescope should be able to track and point precisely, including non-sidereal objects, through the use of a Pointing Model. For the sake of reliability and robustness, it was decided to make this TCS Linux based. One of the main characteristics demanded by the university was the possibility of allowing remote operation from the University Campus situated downtown. To achieve this, we decided that the Graphical User Interface (GUI) should be web-based, which could simplify the communication and also would allow operations independently of the observer computer's operative system. This TCS through its GUI would also allow a script based access, preferably in python, to achieve robotic operations. Also, as mentioned before, back compatibility with existing scheduling systems like RTS2 (INDI based) or ACP (ASCOM based) was considered.

Finally to mention that the upgrade budget needed to be kept the lowest possible since it was going to be performed using the standard operating costs of the observatory.

### 4. PUC40 AND ESO50 UPGRADE

#### 4.1. Hardware Modifications

The two telescope upgrades were at the end very similar. PUC40 used to be driven by three motors

per axis: the first one for slewing, the second one for centering and the third one for tracking. These motors were removed and replaced by a single DC motor per axis installed where the tracking motor used to be.

The speed control loop of each motor was closed using an 100000 steps/rev Gurley encoder placed on the worm. For the 50cm, since it was already using a single DC motor with a coupled relative encoder, we did not performed any hardware upgrade.

#### 4.2. Control Electronics and Software

In both, PUC40 and ESO50 telescopes, all the old electronics was removed. As motor drivers we used industrial DC Servo controllers IpecMot 48/10. Each one of these drivers are controlled through a TCP/IP protocol by Raspberry Pi. Also the position control is controlled through a Raspberry Pi. The communication between clients and servers was achieved using the ZeroC-ICE Framework. The web GUI is mounted in another Raspberry Pi which can be accessed through an Internet connection allowing remote control.

#### 4.3. Real-time Web Interfaces

In order to allow remote control of the devices, we decided to use standard HTML5/CSS3 Web-Interfaces. The communication between the interfaces and the control software is achieved using a web-sockets protocol which has more capabilities than other popular ones. The main reason for this configuration is that web-sockets allow to refresh the displayed values and execute JavaScript functions of the web-page several times per second without reloading the full webpage, which is fundamental for a real-time feedback. These refresh frequencies could not be achieved using a more popular web framework like AJAX. Java interfaces could achieve those frequencies, but they require the installation of plug-ins which may not always be easily kept up to date or compatible with all the possible user's platforms.

The mount control web-interface can be split into four main sections:

1. A GoogleSky API is reflecting the current respective position and targets of the telescope on the sky.
2. Real-Time feedback of the mount variables such as current equatorial and alt-azimuthal coordinates, hour angle, sidereal time telescope and pointing model offsets and distance to the current target.
3. This section allows all type of actions which can be done by the mount, such as pointing or synchronizing a target. Since target field is linked to the Aladin's Database, it is possible to enter any compatible target name to observe it.
4. This section represents a 3D model of the position of the telescope together with the airmass graph of the current target for the night.

On the instrument side, whilst the spectrograph at the 50cm was delivered with its web interface, the 40cm was equipped with an imager. It was then necessary to develop a similar observational interface for the imager using JS9, which is the JavaScript version of DS9 and which provides most of the functions of the desktop software embedded on a web-page.

### 5. UPGRADE OF THE ESO 1M TELESCOPE AT LA SILLA

#### 5.1. Initial Status

The ESO 1m telescope was the first telescope installed at La Silla in 1966. It was first used as a photometric telescope but then was decommissioned in 2005 and wasn't used ever until our upgrade. Since optics had not been cleaned or coated since then, the reflectivity was below 50% but came back to suitable levels after a realuminization. Concerning mechanics, both telescope drivers were running smoothly. The dome was still operational and just needed some lubrication.

#### 5.2. Telescope Upgrade

The dome mechanics and hardware were left unchanged, and the absolute encoder was changed by a relative encoder of 8000 steps per turn with an index signal. , and the control is performed by closing and opening relays placed in parallel of the movement buttons.

The focusing motor which was initially a DC motor which could move the secondary mirror was replaced by a stepper motor at the same position which has the advantage of giving simple feedback regarding its position when using standard Arduino-based control electronics responding to the well known Robofocus protocol.

As the telescope was equipped with DC motors on both drives since its origins, it was not necessary to change them. However, the encoders placed on the worm axis to close the motor speed control loop was replaced by Gurley relative encoders giving a final resolution of 100.000 steps per turn, equivalent to 0.03 arc-second per step on the sky.

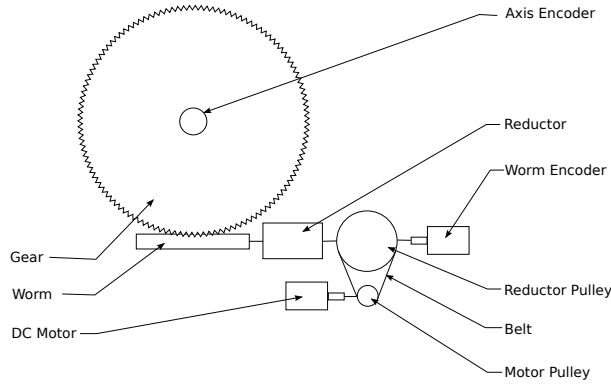


Fig. 1. Each of the axes of the telescope can be represented under this form. The major reduction is performed with a worm/gear system of 360 teeth and two additional reduction with gears and pulleys respectively. The motor is placed after the latest reduction system. One encoder is installed on the worm axis and the other one on the main axis.

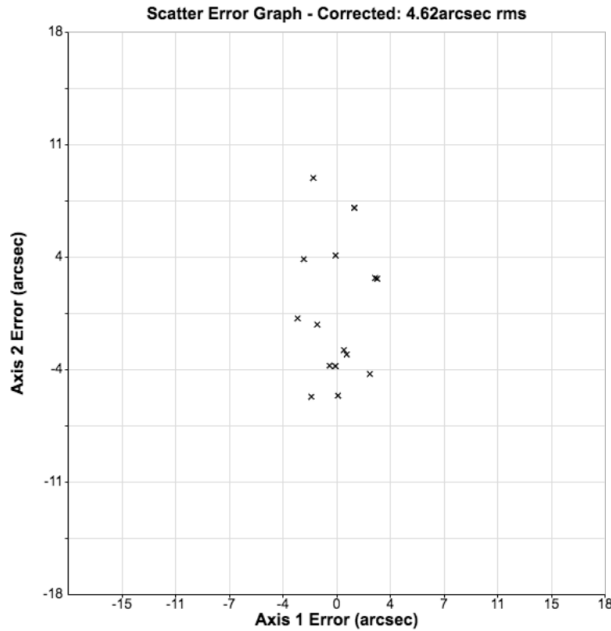


Fig. 2. Pointing residuals of the telescope control system. The graph shows the pointing error in Right Ascension versus the pointing error in Declination expressed in arcseconds for 15 bright stars randomly selected in the sky.

When these mechanical changes were performed, the same electronics box that was installed on the ODOC telescope was installed in the ESO1m as presented by Ropert et al. (2016). The main difference between the ODOC telescope is that rotative encoders of 1,000,000 steps/rev after interpolation were installed on the axes of the telescope. The final configuration of the right ascension axis and the declination axis is shown in Figure 1. For each axis,



Fig. 3. Example of adaptation on a telescopes control system on a commercial AZEQ-G mount. Original DC motors were replaced by stepper motors powered by a customized Arduino shield.

the final position feedback is obtained by fusing the values of the encoder installed on the axis and the encoder installed on the worm using an Extended Kalman Filter (EKF)

### 5.3. Results

After the installation at La Silla observatory, we proceeded with the encoders calibration and the PID control loop's parameters adjustment.

The initial pointing model was obtained with a set of 15 known stars uniformly spread in the sky. Figure 2 shows the pointing residuals which were below 5 arc-seconds RMS. The pointing errors observed during the following nights after the first light of the instrument FIDEOS were consistent with the pointing model residuals. FIDEOS is a Fiber-fed Dual Echelle Optical Spectrograph designed by Tala et al. (2014) with a resolution of  $R=43000$ .

Table 1 details the complete cost needed to refurbish the telescope from its cleaning to the calibration including man-made hardware and development. We can see that the overall project could be held in less than 25k USD.



TABLE 1  
SUMMARY OF PROJECT COSTS IN USD

Item	Human Resources	Material Cost	Total Cost
Computers & Electronics	0	10000	10000
Cables/trash removal	2000	500	2500
Software adaptation	7000	0	7000
Dome control adaptation	1000	500	1500
Transportation & lodging	0	3000	3000
Total	8000	16000	24000

## 6. FUTURE APPLICATIONS

We have shown that it is possible to build a professional grade Telescope Control System using low-cost hardware and electronics. Since the system is designed to be versatile and easily adapted to existing massive telescopes of all kinds and epochs, it can also be downscaled for smaller amateur-class telescopes. As a consequence we are currently implementing a variation of the TCS in an Orion AZEQ-G Mount as an experiment displayed in Figure 3. We are using a single Raspberry Pi Single Board Computer for control purposes, and we replaced the industrial DC Motor Controllers by an Arduino board with a customized shield equipped with two Pololu DRV8825 stepper motors drivers. In this project, we expect to significantly improve the mount's performance with a low budget.

**Acknowledgements:** V.S. acknowledges support from BASAL CATA PFB-06. A.J. acknowledges support from FONDECYT project 1171208, BASAL CATA PFB-06, and by the Ministry for the Economy, Development, and Tourism's Programa Iniciativa Científica Milenio through grant IC120009, awarded to the Millennium Institute of Astrophysics (MAS). S.R. acknowledges MINECO project FIS-2017-89850-R for partially supporting this project.

## REFERENCES

- Baffico, M., Avila, G., Baade, D., et al. 2008, SPIE, 7012, 20  
 Ropert, S., Suc, V., Jordán, A., et al. 2016, SPIE, 9912, 4  
 Shen, T., Soto, R., Revco, J., et al. 2012, SPIE, 8451, 1  
 Tala, M., Berdja, A., Jones, M., et al. 2014, SPIE, 9147, 89