Doctoral Thesis

Design of a Portable Observatory Control System

Vincent Suc

Centre for Sensors, Instruments and Systems Development
UNIVERSITAT POLITECNICA DE CATALUNYA
Shaping light to your needs

Doctoral Thesis

For obtaining the Doctoral Degree
in the field of Optical Engineering from the

Technical University of Catalonia
UPC-BarcelonaTech

Terrassa Faculty of Optics and Optometry FOOT

Design of a Portable Observatory Control System

Author:
Vincent Suc

Thesis Director:
Dr. Santiago Royo

Tribunal members:
Dr. Salvador Ribas
Dr. Salvador Bará
Dr. Marco Rocchetto

Presented in Terrassa, Spain, September 4, 2018

Acknowledgments

Time finally came to write these lines. Lines which have been in my mind all over the writing of this thesis and I would like to dedicate a few words to all the people who supported me along the years and made possible this achievement.

First of all, I would like to thank my wife Macarena for her constant help and presence along this extense work. There are no words able to describe her immense contribution, her sweetness, and moral support adequately. Not only she made it possible but gave me the strength to drive it along with my work and the development of the company born out of it. Thank you for the patience you had in driving me through this process.

Thanks to my family, my fathers, and sister who always supported my choices whatever they are. Thanks for being present and for your constant advice during all this process. Thanks also to my Uncle and aunt for indicating to me the path to science life.

I am sincerely grateful for the help of Santi, my advisor for trusting in this project, his patience, and advice. Without his directions and orientation, writing this work would not have been possible.

Thanks to Andrés for his revisions orientations and dedication from the very first moments of this project.

I would also add a thought to the ObsTech team Samuel and Rodrigo who showed to be much more than coworkers and made the applications of this work come true.

I also want to thank my dear friends Esther and Pepe for their long talks, the time they dedicated to reading it, their corrections and clarifications and their help which went so much beyond the technical point of view.

Thanks to my CD6 friends Francisco Miguel Jordi and Reza who participated in this project, the construction of its first prototype and the first and intense installation.

Finally, I would like to dedicate this document to my grandparents gone too early to see it but whom the sense of humor, positivism, sweetness, and professionalism, made most of the person I am today and gave me the tools to reach this moment.

Table of contents

List of figures

List of tables

Glossary

Application Programming Interface (API) An Application Programming Interface (API) is a particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another particular software program that implements that API. 79, 81

AStronomy Common Object Model (ASCOM) ASCOM is an open initiative to provide a standard interface to a range of astronomy equipment including mounts, focusers and imaging devices in a Microsoft Windows environment. 79, 80, 90

Electrically Erasable Programmable Read-Only Memory (EEPROM) Electrically Erasable Programmable Read-Only Memory (EEPROM) is a type of non-volatile memory used in computers, integrated into microcontrollers for smart cards and remote keyless system, and other electronic devices to store relatively small amounts of data but allowing individual bytes to be erased and reprogrammed. 54

Instrument Neutral Distributed Interface (INDI) The Instrument Neutral Distributed Interface is a distributed control system (DCS) protocol to enable control, data acquisition and exchange among hardware devices and software front ends, emphasizing astronomical instrumentation. 79, 80

Internet Communications Engine (ICE) The Internet Communications Engine, or Ice, is an open source RPC framework developed by ZeroC. It provides SDKs for C++, C#, Java, JavaScript, MATLAB, Objective-C, PHP, Python, and Ruby, and can run on various operating systems, including Linux, Windows, macOS, iOS, and Android. ICE implements a proprietary communications protocol, called the Ice protocol, that can run over TCP, TLS, UDP, and WebSocket. As its name indicates, ICE can be suitable for applications that communicate over the Internet, and includes functionality for traversing firewalls. 79, 80

Proportional–Integral–Derivative controller (PID) Proportional–Integral–Derivative controller (PID) is a control loop feedback mechanism widely used in industrial control systems and a variety of other applications requiring continuously modulated control. 55

Pulse-width modulation (PWM) Pulse-width modulation (PWM) is a modulation technique used to allow the control of the power supplied to electrical devices, especially to inertialloads such as motors. 55–57

Specification Langage for ICE (SLICE) The Specification Language for ICE (SLICE) is a simple definition object-oriented language in which we define member functions of every possible server object present in an ICE collection of peripherals. 80

1 Introduction

In this thesis, we synthesize the development of a new concept of operation of small robotic telescopes operated over the Internet. Our design includes a set of improvements in control algorithmic and hardware of several critical points of the list of subsystems necessary to obtain suitable data from a telescope.
We can synthesize the principal contributions of this thesis into five independent innovations:

An advanced drive closed-loop control We designed an innovative hardware and software solution for controlling a telescope position at high precision and high robustness.

A complete Telescope Control System (TCS) We implemented a light and portable software using advanced astronomical algorithms libraries for optimally compute in real-time the telescope positioning. This software also provides a new multiple simultaneous pointing models system using state machines which allows reaching higher pointing precision and longer exposure times with external guiding telescopes.

A distributed software architecture (CoolObs) CoolObs is the implementation of a ZeroC-ICE framework allowing the control, interaction, and communication of all the peripherals present in an astronomical observatory.

A patented system for dynamic collimation of optics SAPACAN is a mechanical parallel arrangement and its associated software used for active compensation of low-frequency aberration variations in small telescopes.

Collimation estimation algorithms A sensor-less AO algorithm have been applied by the analysis of images obtained with the field camera. This algorithm can detect effects of lousy collimation. The measured misalignments can later feed corrections to a device like SAPACAN.

Due to the constant presence of new technologies in the field of astronomy, it had been one of the first fields to introduce material which was not democratized at this time such as Coupled Charged Devices, internet, adaptive optics, remote and robotic control of devices.
However every time one of these new technologies was included in the field it was necessary to design software protocol according to the epoch's state of the art software. Then with the democratization of the same devices years after the definition of their protocols, the same communication rules tend to be used to keep backward compatibility with old - and progressively unused- devices. When using lots of cumulated software knowledge such as with robotic observing, we can dig in several nonsenses in the commonly used architectures due to the previously explained reasons.
The described situation is the reason why we will propose as follows a new concept of considering an observatory as an entity and not a separated list of independent peripherals. We will describe the application of this concept in the field or robotic telescopes and implement it in various completely different examples to show its versatility and robustness.

First of all, we will give a short introduction of the astronomical concepts which will be used all along the document, in a second part, we will expose a state of the art of the current solutions used in the different subsystems of an observing facility and explain why they fail in being used in small telescopes. The principal section will be dedicated to detail and explain each of the five innovations enumerated previously, and finally, we will present the fabrication and integration of these solutions. We will show here how the joint use of all of them allowed obtaining satisfactory outstanding results in the robotic use of a new prototype and on the adaptation on several existing refurbished telescopes. Finally, we dedicate the last chapter of this thesis to resuming the conclusions of our work.

2 Astronomical and Telescopes Mechanics

2.1 Celestial coordinates

In this section, we will give basic concepts of celestial mechanics which will be widely used in this thesis. It is important to notice that several standard symbols and notations will be repeated along the document. It is possible to refer to Tab.2.1.1 to find a summary and brief explanation of each one of them. An important number of introduction books have been written such as [2] or [3] in which basics of astronomical concepts and coordinates are defined. Regarding computational programming and optimization of such problems, one could refer to the J.Meuus's book[4].

Symbol	*Name*	*Description*
ϕ	Latitude	This is the latitude on the earth of the position of the observer.
λ	Longitude	This is the longtitude on the earth of the position of the observer. Positive values are East of Greenwich
	Celestial equator	The celestial equator is the projection on the terrestrial equator on the sky.
	Ecliptic	The ecliptic is the plane on whith the Earth rotates around the Sun it is tilted relatively to the celestial equator by an angle of $23^o26'$ approximately due to the inclination of Earth's rotation axis.
γ	Vernal point	The vernal point is one of the two intersections of the ecliptic with the celestial equator. By definition it is the direction of the position of the Sun at Sunrise on the first day of spring at Greenwich.
	Airmass	The airmass represents the relative quantity of athmosphere the target has to cross for the observer relatively to a target at zenith. Its value is 1 at zenith.
α	Right Ascention	This is the first coordinate of the target. It is measured in Hours minutes and seconds from the vernal point. It can be asimilated to the earth's meridians.
δ	Declination	This is the second coodinate of the observed target. It is measured in Degrees, minutes and seconds from the celestial equator. It can be assimilated to the earth's parallels.
h	Hour angle	The hour angle is the angle between the observer and the meridian viewed from the observer.
TT	Sidereal time	The sidereal time is given by the right ascention of the target crossing the meridian at a given instant for a given observer's position on Earth.
	Precession	The precession is the effect due to the periodic variation of the direction of the earth axis. As a result the vernal point does a complete rotation in approximateley 26.000 years.
	Nutation	The nutation is the periodic variation of the Earth's tilt rotation angle relative to the ecliptic. Its main periodicity is around 18 years and amplitude of 20".

Table 2.1.1: Important astronomical coordinate related terms and symbols.

2.1.1 Alt-Azimuthal coordinates

For a terrestrial human observer, the most intuitive way to refer a position in the sky is by using Alt-Azimuthal coordinates. Fig.2.1.1 represents an observer situated at 45^o north latitude on Earth. As we can see, the observer is placed in the center of the referential and any position in the sky can be identified by its azimuth and elevation. Azimuth is measured as a horizontal angle between

0^o and 360^o counted from the north direction positive clockwise for the observer. The elevation is measured vertically by an angle between 0^o for an object at the horizon to 90^o for a target at zenith.

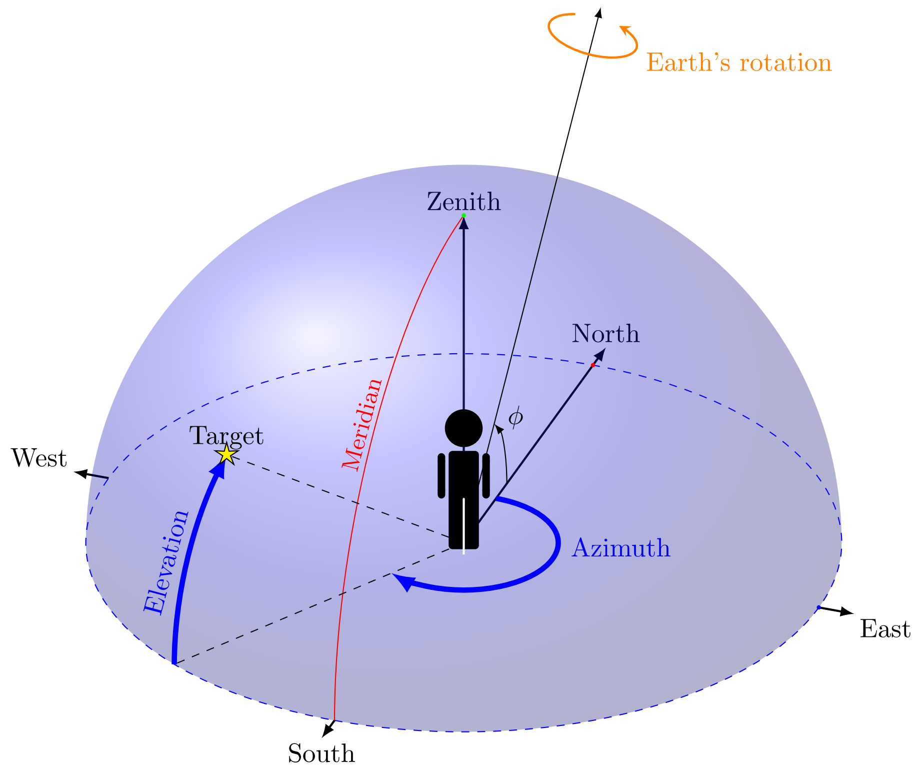


Figure 2.1.1: Aximuthal coordinates concepts.

Even is the Alt-Azimuthal notation is quite convenient for the earth referenced cases because the main axis is aligned with gravity vector which makes things very convenient for flexure or atmospheric related computations, it is not the most adequated for absolute referencing of stars position. When the Earth is rotating around the Sun and on its main axis, the path done by the stars along the night or the year in this referential is not straight-forward when expressed in this reference frame.

2.1.2 Equatorial coordinates system

The equatorial coordinates system it the mostly used coordinates system for referencing position of celestial objects. It is an earth center spherical coordinates system. Fig.2.1.2 represents the concepts necessary to express the equatorial coordinates of a target.
A reference plane is the projection of the terrestrial equator on the sky is called the celestial equator. A reference point of the celestial equator, called vernal point, corresponds to the direction of the sun at the first day of spring at Greenwich. The equatorial coordinates, which can be assimilated to the terrestrial latitude and longitude are then counted from this reference frame.

The equivalent of the terrestrial longitude called the Right Ascension δ is measured along the celestial equator as an angle between the vernal point and the target object. Declination is measured from -90 to +90 degrees as the angle to the celestial equator as the terrestrial parallels.
The right ascension is measured in Hours of time, corresponding at the time an observer must wait , observing from a fixed position on earth between the instant where the vernal point crosses meridian and the measured position crosses. As a result, Right ascention is measured from 0h00m00,0s to one sidereal day, the time in which earth is doing a full 360 degrees rotation or 23h56m4.1s approximately.

The biggest benefit of using an equatorial reference frame is that over long periods of time, (a few years) the coordinates of a distant star can be considered as constant.
However, the earth axis is affected by a periodic movement called precession of 26.000 years introducing a progressive drift of the vernal point over the years. The inclination of this same axis related to the ecliptic also suffers some oscillation called nutation.
Earth rotation also suffers from speed variations which implicates that some correction must be done to know the proper angle of the vernal point related to the earth.
As a result sky coordinates are always referenced to a given date and a set of correction must be performed to the instant we observe. As an example, a flowchart of corrections to be applied depending on the environmental conditions of the telescope which noticeable affect pointing quality are presented in Fig.2.1.3 where a typical sequence of corrections proposed by P.T Wallace in the SLALIB library[5, 6] from the catalog position to the real telescope position is presented.

2.2 Telescope mounts

Any optical component included on a telescope have to be mounted on a mechanical arrangement in order to be aimed a the target desired positions. As we can see in [7], telescope mounts can be generally categorized into three mechanical families, each with its own pros and cons. We will see how the configuration of the mount can benefit diverse kind of telescopes.
Since one of the purposes of this thesis is to obtain a generic control system compatible with any kind of telescope it is important to differentiate the particularities of each of the possible mechanical configurations.

2.2.1 Fork Equatorial mounts

Fork mounts have one of their mechanical axis arranged to stay parallel to the Earth's axis. The advantage of this configuration is to have, in an ideal arrangement, a single motor moving at a fixed speed in order to compensate for the Earth's movement in long exposures.
The problem is that these mounts must be very heavy to be rigid enough so the eventual unbalanced charges and flexures are avoided. The whole weight of the tube is making a big momentum on the fork and on the main axis. An example can be seen in figure 2.2.1.

2.2.2 German/British equatorial mounts

On German and British mounts the telescope is supported on a single side of the main axis. A counterweight stands on the opposite side. The advantage of these mounts is that the shape of the mount does not depend on the size of the tube, so it can support different types of optical arrangements.

British mounts were commonly used in professional telescopes before the use of fiber optics become commonplace to obtain a coudé focus (fixed focus), which can be obtained redirecting the beam of the telescope inside the same mount, allowing the use of heavy instruments (like old spectrometers).

The main disadvantage of British mounts comes from their control. For every position on the sky to be pointing at, there are two mathematically possible positions of the telescope, but only one of them is valid (the one with the tube over the counterweight), otherwise it touches the pillar of the mount. Heavy calculations are necessary in order to avoid this if we don't want to limit the range of action when we pass the meridian.

This means that the tube must be at East of the mount when the telescope is pointing West and on West of the mount when pointing East. That makes control and error correction more complicated. An example of this mechanical mount may be seen in figure 2.2.2. Another relevant problem to be taken into account is that pictures are flipped xy when the telescope is flipped in a

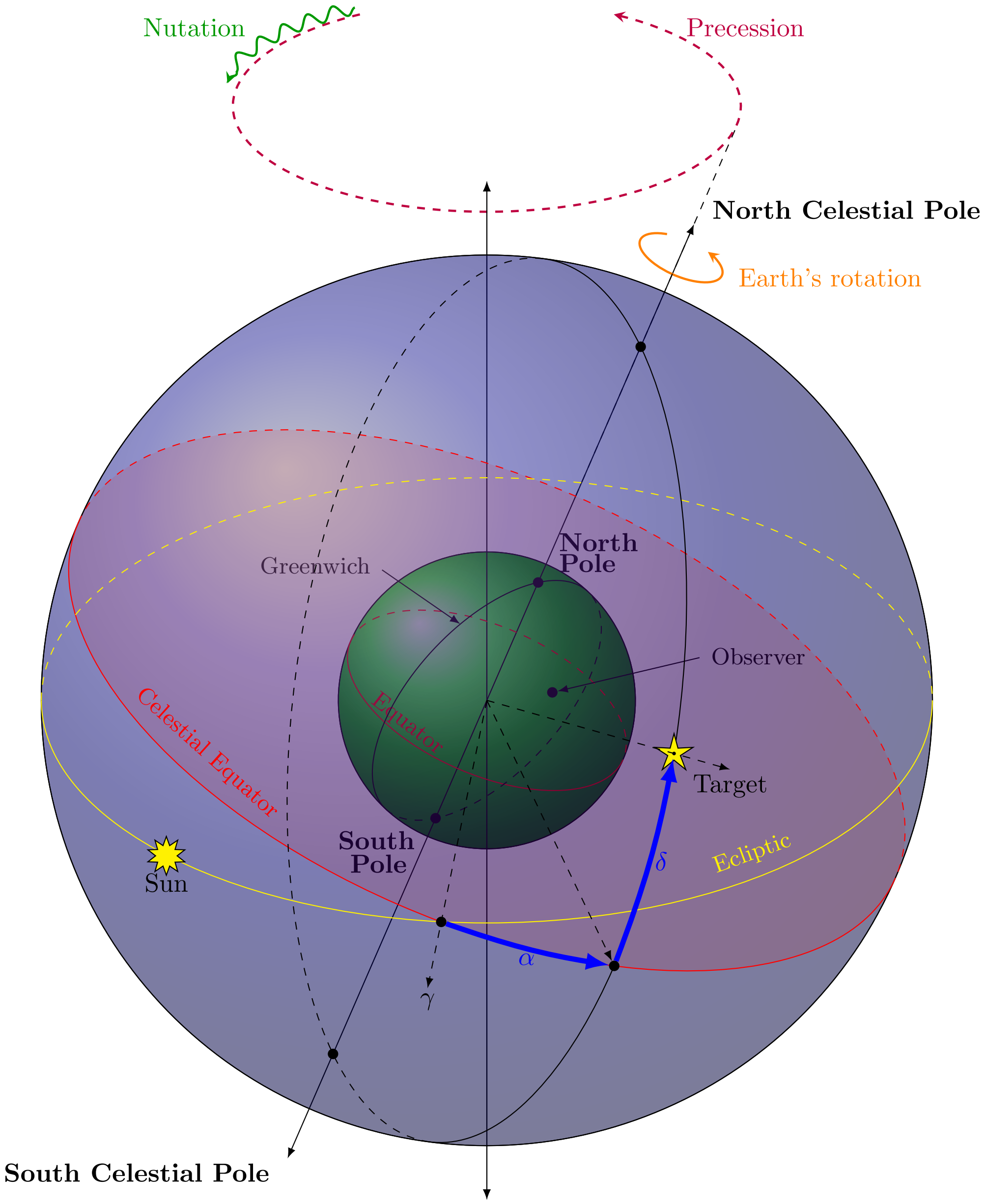


Figure 2.1.2: Equatorial coordinates concepts.

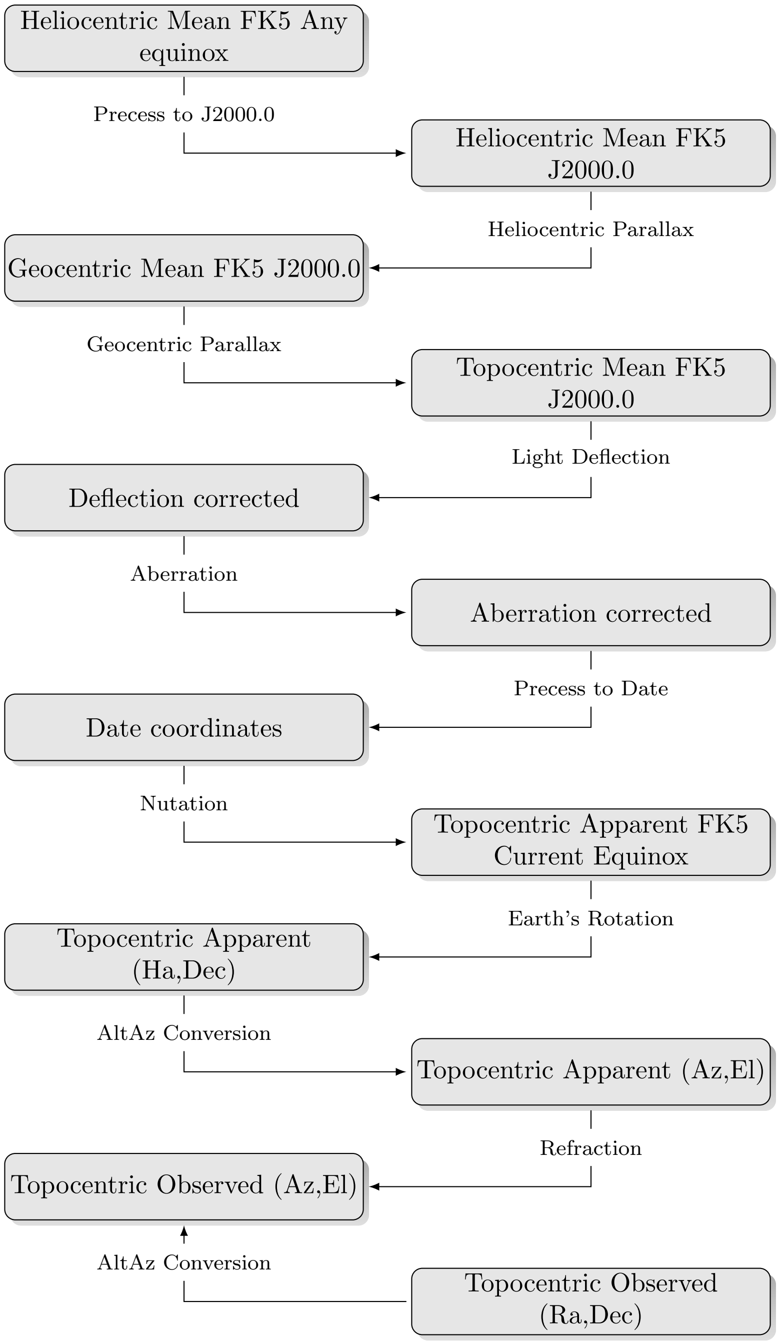

Heliocentric Mean FK5 Any equinox
Precess to J2000.0
Heliocentric Mean FK5 J2000.0
Heliocentric Parallax
Geocentric Mean FK5 J2000.0
Geocentric Parallax
Topocentric Mean FK5 J2000.0
Light Deflection
Deflection corrected
Aberration
Aberration corrected
Precess to Date
Date coordinates
Nutation
Topocentric Apparent FK5 Current Equinox
Earth's Rotation
Topocentric Apparent (Ha,Dec)
AltAz Conversion
Topocentric Apparent (Az,El)
Refraction
Topocentric Observed (Az,El)
AltAz Conversion
Topocentric Observed (Ra,Dec)

Figure 2.1.3: Most useful pointing machine states and transitions according SLALIB and to the Telescope Pointing Machine definitions.

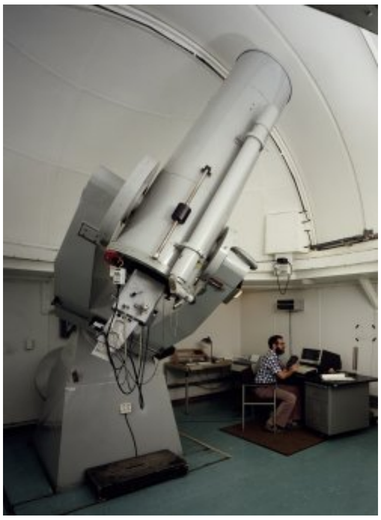


Figure 2.2.1: The Eso 50cm Fork equatorial design

long exposure.

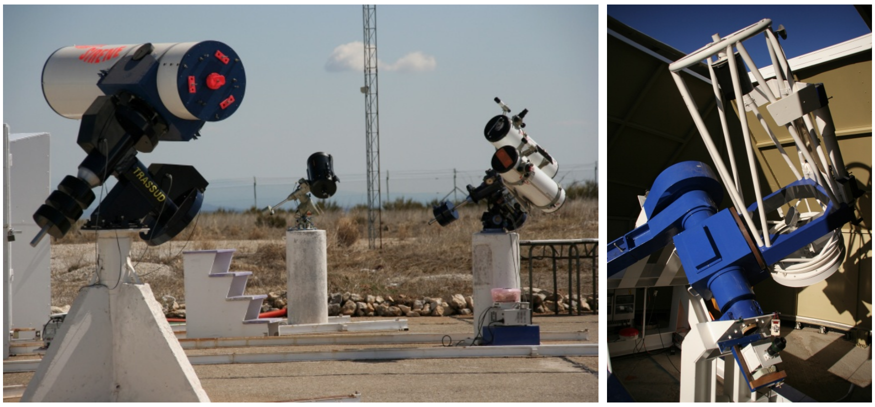


Figure 2.2.2: Telescopes on German and British Equatorial mounts

2.2.3 Azimuthal mounts

An Alt-Azimuthal mount includes a horizontal and a vertical axis. When the mount is installed the vertical axis has to be properly leveled.

- Altitude (Vertical) values vary from 0º (horizon) to +90º (zenith)
- Azimuth (Horizontal) values vary from 0º (North) to 360º clockwise

Even if they are much more simple conceptually, and from a mechanical point a view, controlling this mounts is a relevant challenge since for each pointing direction there are three moving axes (azimuth, elevation, and field rotation), which need very precise speeds and which can vary over an important range. On the other hand, the fact that the mirror is always supported on the same points of the structure gives the possibility of a design without hysteresis in the positioning of optics, allowing much more simple equations in the pointing model. This way it is much easier to have a balanced telescope regardless its position. An example can be seen in figure 2.2.3

Nowadays, in the design of very large and giant telescopes, (6-40m of diameter), all designs converge to azimuthal mounts, because of the difficulties implied by an equatorial mount in such

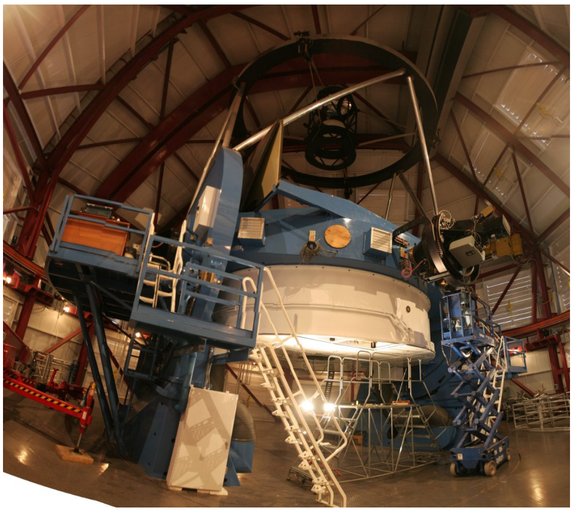


Figure 2.2.3: The Magellan Clay 6.5m AltAz telescope

large systems. The other big advantage of an azimuthal configuration in a large telescope is the possibility of having two Nasmyth foci in which we can install all required heavy instrumentation, which only moves around itself for field rotation. Heavy instrumentation can be used also on equatorial mounts using a coudé focus, although this implies a higher number of reflections and a lower throughput efficiency in the system.

2.3 Usual optical configurations

Possible configurations for the optics of a telescope are numerous. We will present in this section the most commonly used as described in [8] which also corresponds the ones the system will have to be compatible with.

2.3.1 Newtonian

In a Newtonian telescope, the primary mirror is parabolic and there is a 45 degrees folded flat secondary mirror between the prime focus and the primary mirror that sends the focal plane outside the tube. This design is very simple but needs a corrector if we want to observe wide fields because of the importance of field coma (fig. 2.3.1).

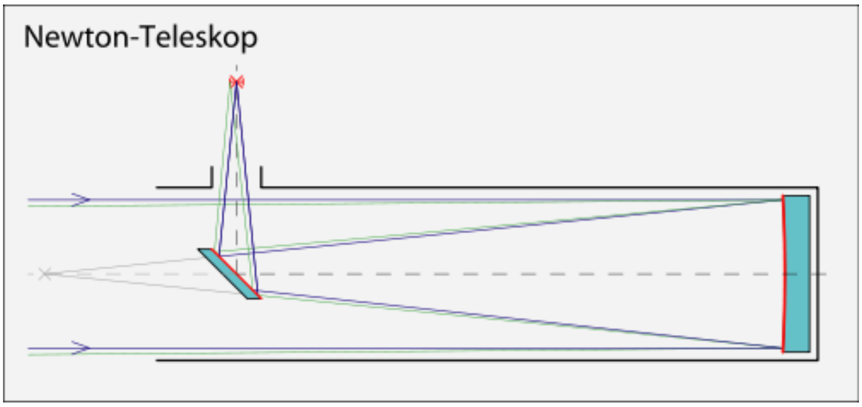


Figure 2.3.1: The Newtonian configuration

2.3.2 Cassegrain

In a Cassegrain telescope, there is a concave parabolic primary mirror and a convex hyperbolic secondary mirror which sends the focal plane through the primary mirror, in which we need a hole. This configuration has the advantage of being very compact and is a classical solution for amateur astronomers who need a transportable solution. In this configuration, the effect of field coma remains strong and these telescopes still need a strong coma corrector in order to give proper images as soon as you try to observe away from the optical axis (fig.2.3.2).
This configuration along with the Newtonian one are the most common configurations among the amateur astronomers. Newtonian one is more stable in terms of vibrations and parts stability while the Cassegrain one is giving the advantage of a smaller tube.

2.3.3 Ritchey-Chrétien

The Ritchey-Chrétien telescope is a variation of the Cassegrain telescope where both primary mirror and secondary mirrors are hyperbolic and the shapes of the surfaces are adapted to each other. This configuration gives better results far away from the optical axis (fig.2.3.2).

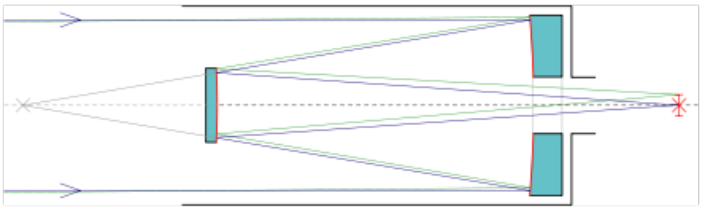


Figure 2.3.2: The Cassegrain or Richey-Chretien configuration

2.3.4 Gregorian

In a Gregorian telescope, the optical configuration is equivalent to that of a Cassegrain one but for the fact that the secondary mirror is concave too, and is placed after the prime focus. This has the advantage of giving a real pupil between the primary and the secondary mirror when we look from the detector position. It can be used to install a removable flat field screen usable for spectroscopic calibrations, avoiding that way a screen of the same size of the primary mirror (fig.2.3.3).

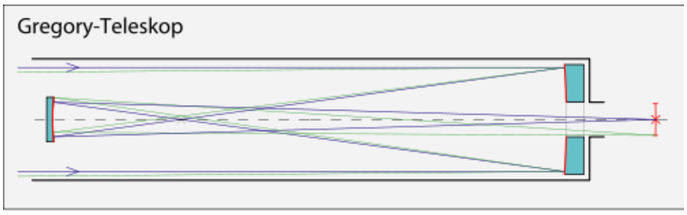


Figure 2.3.3: The Gregorian configuration

2.3.5 Prime focus

In a prime focus telescope, the detector is directly installed in the focal of the primary mirror. This pushes to have a single science instrument in the telescope, and in addition, it must be small in order to avoid large aperture obstruction. On the other hand, it is a setup which only has one reflector (the primary) so there is a single surface to polish making the system cheaper in some cases (here we should notice that as a consequence of the configuration the tube must be longer and that implies a bigger dome (fig.2.3.4). This is the design has been selected for the first prototype in the present project.

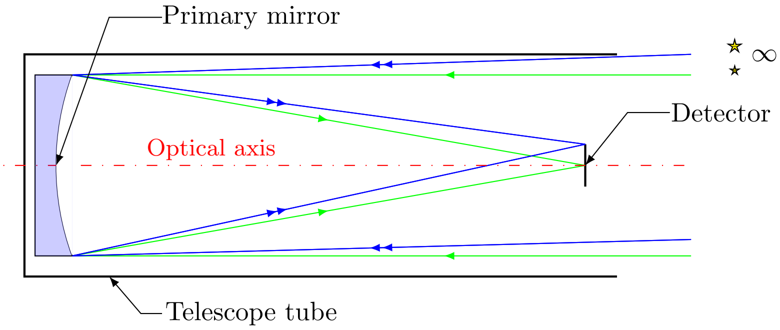


Figure 2.3.4: Lateral view of a prime focus telescope. Chief (green) and marginal (blue) rays coming from stars at infinity are represented.

3 State of the Art

In the field of telescope design, a long path have been done since the 1960's but most of the software, or hardware solutions which are used in modern profesional telescopes are based on solutions and restrictions which were established several decades ago.

For this reason we are making in this chapter a review of the existing methods or hardware solutions used to facilitate the obtention of automatic data. Our goal is to analyse way large profesional telescopes adquire data from the point of view of several steps of the adquisition chain. This mean from the software and hardware point of view of movement of the telescope to the data analisis and their retro alimentation in order to get better image quality.

3.1 Applied algorithmics to robotic observing

In this section we will present 4 main lines to improve the image quality obtained through a telescope. In the first two sections we will analyse the techniques used for obtaineing a better axes control will allow to keep the telescope to track at the right speed and maintain the stars in the same position of the detector. The last two steps will refer to what kind of litterature can be found in the litterature itself, first we will see what has been used to better align the optics of a telescope dependign on its position and then we will have a view on the most ommonly used algorithms to measure the optical wavefront aberrations in order to provide better knowledge and correciton of it.

Along this study, we will analyse the status or feasability of their possible porting to small telescopes.

3.1.1 Telescope axes control

The end of the XXth century is characterized by the birth and growth of computers and their respective capacities.
The complexity of astronomical movements over the sky and the extremely high precision needed by the astronomers to obtain the best data from the stars rapidly created the need to introduce the use of computers and algorithms in order to compensate tracking errors which could hardly be taken into account by the mechanics of telescopes.
As soon as 1976, the AAT (Anglo Australian Telescope) [9] at Siding Springs Observatory was equipped by numerical controllers in order to improve its pointing and tracking.
Closed control loop algorithms such as the Proportional-Integral-Derivative algorithm (PID) rapidly impose themselves in this aplication which implicates controlling both axes of the telescope to track a moving target which is the star or sky coordinates we want to aim the telescope at.

As described in the previous chapter, the earth is moving over its axis at a constant speed of 15 arcseconds per second and as first approximation this is the needed tracking speed on the right ascention axis of an equatorial telescope. The tracking precision needed by a telescope is usually driven by the size of a star on the telescope imager which must be rigourusly kept at the very same position of the detector while it is integrating the light of the target. on the other side, the athmosferic turbulence degrades the image quality and make that the best achivable star point spread function on the detector is hardly better than 0.3 to 2 arcseconds in diameter depending on the quality of the observing site.

As a result, for telescopes which are note equipped with adaptive optics the limitation is driven by the athmospheric quality but the maximum acceptable tracking error during the observation if we want to keep correct shapes of the stars still remains below a fraction of an arcsecond which is still mechanically hard to achieve.

As a result, if we want to keep the star within 0.3arcseconds of its nominal position while tracking at sidereal speed (15.04 arcsec/s approximately) it is necessary to choose a control time sampling δt so that the angular distance done by the star would be less than 0.3arcseconds at this speed. As a consecuence, it was early defined that the axes control loop need to have a time sampling inferior to a thenth of the maximum acceptable error which would be around 20ms.

Since this time sampling was not sufficient for the epoch's computers existing at this time to proceed to astronomical computation and correction Wallace proposed to split the positioning control loop of a telescope into two loops running at different speeds. The fist one would control blindly the motor speed closing the loop with a tacometer or an encoder while the second one would control the position of the telescope at a lower rate allowing astronomical correction computation.

Computing performance and the availablility of closing a position control loop at a sampling period inferior to 100ms also allowed the design of the first alt-azimuthal telescopes available to give similar image quality thant the equatorial ones. The multi mirror telescope which starte operating in 1979 is one of the first big size telescopes to use an alt-azimuthal design. The advantage of this configuration is a more compact mechanics and dome which allows to build a bigger telescope for a fixed budget and less mechanical restrictions in terms of flexures and mechanical tensions. The counterpart of this it the fact that it becomes absolutely necessary to recompute the transformation from equatorial coordinates to mount coordinates at each loop iteration in order to have a proper tracking since both axes would have to track to different and non-constant speeds all the time in order to keep the star in the field of the sensor.

A similar control loop was implemented at the Multi Mirror Telescope (MMT) from its first initial design [10] where the speed control loop runs at 4kHz while the position control loop runs at 50Hz. For the new design of this telescope by the Smithsonian Institution for Astropysics (SAO) and the University of Arizona, High resolution on-axis encoders coupled to specific control algorithms have been used.

A PID consist of a control loop that send a set point speed to the motor. An encoder measures the position or speed error and the PID corrects the set point speed of the motor by a sum of three terms. The first term is proportional to the last measured error, the second to the sum of all the previous errors, and the third to the mean of the previous errors. This specific algorithm attacks DC servomotors that have their own high resolution PID control embedded. The PID parameters are actively tuned using the algorithmics when the speed of the motor changes, allowing to get a better response of the motor at every speed. This strategy is going to be adapted to the active telescope we are proposing.

Fig.3.1.1 is a representation of the organization of the control loops of the elevation azimuth axis of the MMT (the azimuth control loop is very similar and thus not displayed). First of all we will remark the presence of a GPS timing board for a precise timing of the absolute position which is indisociable with a precise pointing. Then we can note that the motors are controlled by a standard PID which receives the signal from a high resolution relative encoder. The reference to the encoders is given by an absolute encoder of lower resolution and the control computer unit send the PID parameters to the PID controller according to the current speed of the telescope.

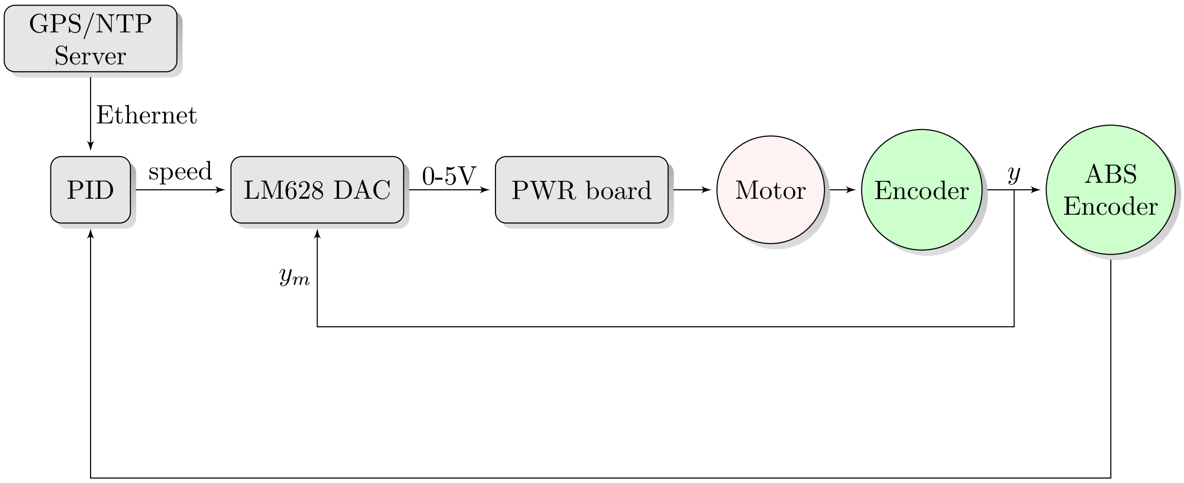


Figure 3.1.1: The MMT elevation control loop diagram

However, controlled close loop tracking has been for a long time reserved to the bigger telescopes where the latest technologies were implemented and remained that way out of accessibility of small scientific projects or amateur class telescopes. Implementing such a control usually implicated the use of a full rack of computers and custom designed control boards which cost and maintenance were prohibitive for the 20cm-1m class telescopes.

Embedded microcontrollers and microcomputers got progressively democratized within this class of telescopes and the beginning of the XXth century saw the massification of GoTo systems which allowed to point equatorial telescopes and allowed control from personal computers.

However these systems implemented only basical coordinates transformations or mount corrections running in open loop on the axes since the computing capabilities of these massified microcontrollers didn't allow such corrections.

Shen & al.[11] proposed a distributed arquitecture we can see in Fig:(3.1.2) where the fastest and simplest operations could be run in an embeeded control loop in a standard PIC16F876 microcontroller while positioning could be run in an embedded computer and most complex operation in a standar personal computer. This arquitecture allowed to export the most coslty operations to more powerfull system while the fastest operation could be run inside dedicated real time hardware. This system does not offer close loop control of the positioning but allows to distribute the power and the speed of the system inside a low cost overall arquitecture.

3.1.2 Telescope pointing

In a telescope mount, more particularly when applied to alt-azimuthal systems, it is very important to have a very precise pointing algorithmics. In such configuration, each position in the sky the telescope will need different tracking speeds in each of its three axes (altitude, azimuth, and field rotation). If we have a small offset between theoretical and actual pointing, the telescope will track the object of interest with speeds which correspond to a slightly different sky position, and stars will start to form trails on the sensor when long exposures are used. As a consequence, precise pointing has a very relevant role to ensure the position where the telescope is actually aiming at.
An algorithm was proposed in 1979 [12] to compensate drift due to flexures at given position of the telescope reajusting the polar alignment as a function of where the telescope is pointing. At the time computational power wasn't sufficient to recompute speeds in realtime inside a control loop. This is why it was first proposed to add a polar alignment error giving a known aberration to balance other non dynamically correctible effects.
TPoint is a classical pointing model software developed by P.T.Wallace[13, 14].
It was at first implemented for the Anglo-Australian Telescope in 2002. Afterwards, it was also

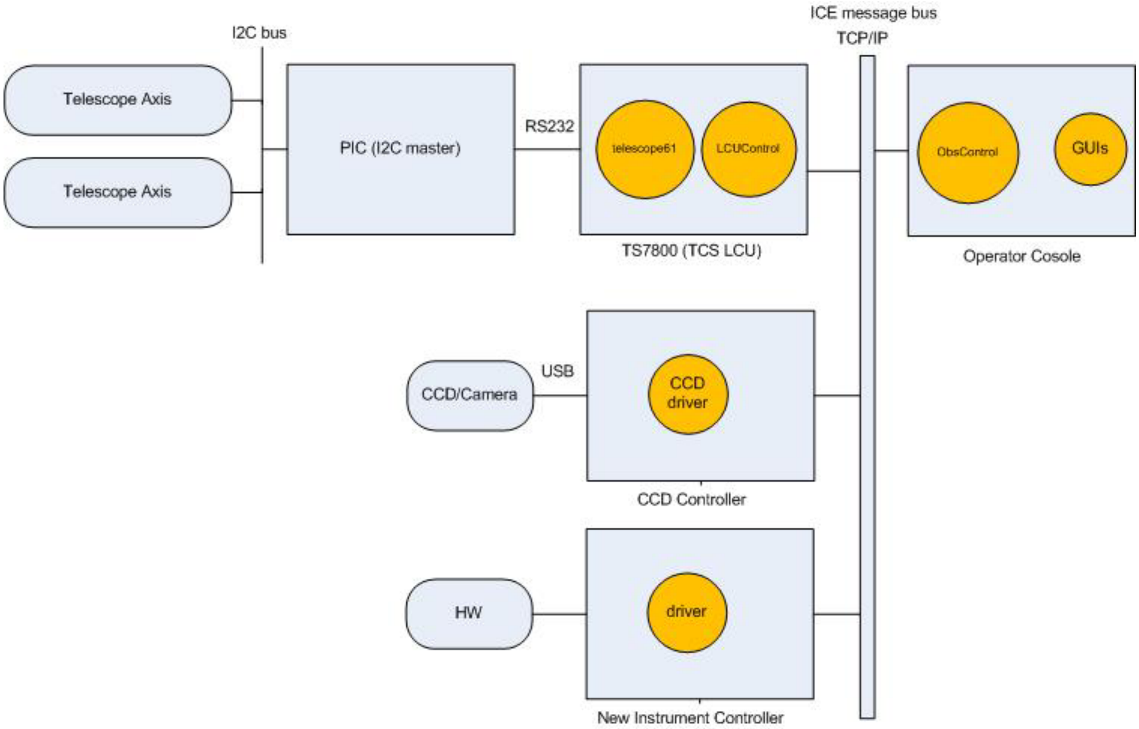


Figure 3.1.2: The Eso50cm telescope control software architecture

adopted for many other professional telescopes such as VLT Keck, Gemini, or Magellan. The goal of TPoint is to set up a correction by software of a majority of the mechanical aberrations of the telescope, and to improve the pointing accuracy by a very important factor. A version of TPoint has also been implemented for amateur applications by the company BisqueSoftware, which can be interfaced with the hardware product TheSky6 [15].
From a general point of view, the principle of a pointing model is to assume that there is an unavoidable difference between the ideal coordinates of the object we are pointing at, obtained from a star catalogue, and the real coordinates the telescope is in fact pointing at. This difference may have a number of sources of different types, some of them dependent exclusively on the environmental conditions and can be computed in advance by the SLALIB or TPM library . These are mostly depend on the date, time, position on earth and coordinates of the target object. These effects are listed in fig 3.1.3.

The following list of effects, however, are only dependent on the mechanism of the telescope itself, and must be calibrated using the same instrument and weight balance to be used in the final observations. They also have a dependence on which position the telescope is pointing at, and may affect severely the position of the object we try to observe on the science camera.

- The telescope is not properly aligned towards the pole (for an equatorial mount) or the zenith (for an azimuthal mount)

- The axes of the mount are not perpendicular

- The zero positions of the encoders used to control the motor position have some offset relative to the true zero position

- The tube or the camera holder have flexures. This flexure changes as a function of the position where the telescope is pointing at, so this effect will require a specific calibration to characterize the error introduced depending on the position

- Some optics can slightly move in their respective cells, or the cell itself may be deformed depending on the position of the telescope

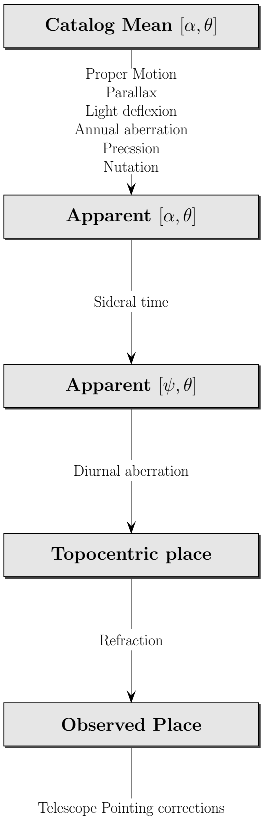


Figure 3.1.3: Corrections to be considered between the catalogue position and the telescope position. α is the Right Ascension, δ is the Declination and h is the Local Hour Angle.

At a larger or minor extent, these errors in pointing are unavoidable and need be corrected. The philosophy of the equations defining a pointing model is presented next. It is a general problem which can be reduced to a minimum square matrix inversion. We will define three matrices which will represent a vector containing the parameters of interest, a matrix describing the equations in the considered pointing model, and a vector containing the pointing error to be calculated. Let us call:

- $p_1..p_n$ the n parameters of the model defining the mount
- (Ψ, Θ) the position of the mount (Local hour angle and declination)
- $f_1(\Psi, \Theta)..f_n(\Psi, \Theta)$ the contribution function of every parameter on the Ψ axis
- $g_1(\Psi, \Theta)..g_n(\Psi, \Theta)$ the contribution function of every parameter on the Θ axis
- $(d\Psi, d\Theta)$ the position error on every axis of the mount at a given position.

We can assume the following relationships for a given position assuming errors are linear:

$$d\Psi = f_1(\Psi, \Theta)p_1 + ... + f_n(\Psi, \Theta).p_n \tag{3.1.1}$$

$$d\Theta = g_1(\Psi, \Theta)p_1 + ... + g_n(\Psi, \Theta).p_n \tag{3.1.2}$$

We can express these relations in a vectorial way:

- Let's call $\vec{F}$ the vector $[f_1..f_n]$

- We call $\vec{G}$ the vector $[g_1..g_n]$
- We call $\vec{P}$ the vector $[p_1..p_n]$

$\vec{F}$ and $\vec{G}$ are vectors containing constant values characteristic of the arrangement of the telescope optics and mount. Then the two previous relationships can be expressed as

$$d\Psi = \overrightarrow{F_{(\Psi,\Theta)}}.\vec{P} \tag{3.1.3}$$

$$d\Theta = \overrightarrow{G_{(\Psi,\Theta)}}.\vec{P} \tag{3.1.4}$$

We now can observe a number m of theoretical positions (Ψ_i, Θ_i) using our telescope; for every of these positions the telescope will observe the real position $(\tilde{\Psi}_i, \tilde{\Theta}_i)$, which may present differences with the theoretical ones. The real position is deduced using a common technique called an astrometrical reduction of the stars present on the field of the CCD [16]. It becomes obvious that the vector

$$[d\Psi_i, d\Theta_i] = [\Psi_i - \tilde{\Psi}_i, \Theta_i - \tilde{\Theta}_i] \tag{3.1.5}$$

Now we can construct the following relation which is a matricial expression of the previous one:

$$\begin{bmatrix} f_1(\Psi_1,\Theta_1) & \cdots & f_n(\Psi_1,\Theta_1) \\ g_1(\Psi_1,\Theta_1) & \cdots & g_n(\Psi_1,\Theta_1) \\ \vdots & \ddots & \vdots \\ f_1(\Psi_m,\Theta_m) & \cdots & f_n(\Psi_m,\Theta_m) \\ g_1(\Psi_m,\Theta_m) & \cdots & g_n(\Psi_m,\Theta_m) \end{bmatrix} . \begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix} = \begin{bmatrix} d\Psi_1 \\ d\Theta_1 \\ \vdots \\ d\Psi_m \\ d\Theta_m \end{bmatrix} \tag{3.1.6}$$

So if we have series of observations we can fill the first and the third matrices of the previous expression and deduce the parameter vector $\vec{P}$ using a least squares solution of the matrix inversion as follows:

$$\begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix} = \begin{bmatrix} f_1(\Psi_1,\Theta_1) & \cdots & f_n(\Psi_1,\Theta_1) \\ g_1(\Psi_1,\Theta_1) & \cdots & g_n(\Psi_1,\Theta_1) \\ \vdots & \ddots & \vdots \\ f_1(\Psi_m,\Theta_m) & \cdots & f_n(\Psi_m,\Theta_m) \\ g_1(\Psi_m,\Theta_m) & \cdots & g_n(\Psi_m,\Theta_m) \end{bmatrix}^{-1} . \begin{bmatrix} d\Psi_1 \\ d\Theta_1 \\ \vdots \\ d\Psi_m \\ d\Theta_m \end{bmatrix} \tag{3.1.7}$$

TPoint is able to solve a model based on linearised equations like the ones in Tab. 3.1.1 and Tab.3.1.2.

Term	*Description*	Δh	$\Delta\delta$
IH	h index error	IH	
ID	δ index error		ID
CH	Collimation error	CH $\sec\delta$	
NP	h/δ non-perpendicularity	NP $\tan\delta$	
MA	Polar axis left-right misalignment	-MA $\cos h \tan\delta$	MA $\sin h$
ME	Polar axis vertical misalignment	ME $\sin h \tan\delta$	ME $\cos h$

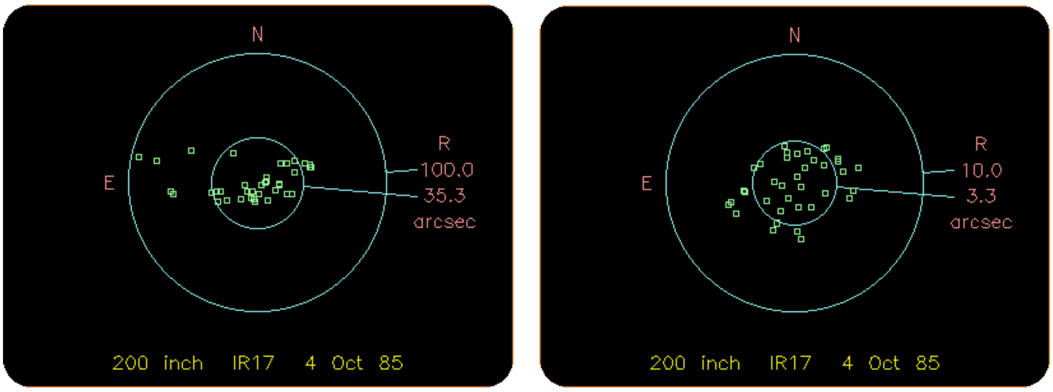
Table 3.1.1: Model of the six primary parameters for TPoint correction

The obtained parameters model the telescope-dependant effects. The possibilities of applying a pointing model can be seen in figures 3.4(a) and 3.4(b). In these graphs we can see that without applying any advanced correction, a high quality mecanism is pointing with a 35 arcsecond rms precision and this value can be divided by a factor 10 with an apropriate mathematical model.

With the increase of computational possibility, TPoint was then implemented not only for correcting pointing but also to include dynamic tracking speeds as a function of where the telescope

Term	*Description*	Δh	$\Delta\delta$
TF	Tube flexure	TF $\cos\phi\sin h\sec\delta$	TF $(\cos\phi\cos h\sin\delta - \sin\phi\cos\delta)$
FO	Fork flexure		FO $\sec h$
DAF	δ axis flexure	$-$DAF$(\cos\phi\cos h + \sin\phi\tan\delta)$	

Table 3.1.2: Advanced TPoint parameters



(a) Offsets correction only

(b) Correction of flexures, misaligments and non perpendicularities

Figure 3.1.4: The HALE 200 inches telescope with two kinds of corrections

is pointing. Latest generation big telescopes such as Keck[17] or Gemini[18] all use this functions in their tracking control loop. Using this method, the telescope can track longer without drift due to flexures effectos or mechanical misalignments.

Terret et al. [19] proposed in 2006 a c++ class library for large scale telescopes layered on TCSpk and SLALIB that implements virtual telescope objects for generating mount and rotator position and speeds. It also permits the interaction with TPOINT[13].

Percival and al [1] proposed the Telescope Pointing Machine Library (TPM). TPM is less complete than the previous one but popose multi lengage backends such as python.

As we can see in fig. (3.1.2) The telescope is considered as a state machine and transitions of coordinates is considered as a state transition. This concep makes more versatile the change from one state to another and its further implementation in a custom control loop.

3.1.3 Active collimation control

Whenever the telescope changes its position, the mechanical mount is affected by several kinds of flexures which may affect the optics, and the optics support too. Movement or deformation of the optics severely affects the calibration frames, and the vignetting area of the spider holding the secondary becomes displaced across the field of view. Image quality is also affected, especially in Cassegrain or Ritchey-Chrétien designs where the secondary mirror has a magnification effect. In modern design of large telescopes (apertures larger than 5 meters) the primary mirror usually lays on hundreds of actuators and a wave front sensor is set on the focal plane to send corrections to the primary mirror. Even if we could correct higher order aberrations, we cannot compensate for the misalignment introduced in the optics due to flexures only using hte primary mirror. This misalignment include strong field aberrations that must be compensated by direct repositioning of the secondary mirror of the telescope, so the optical axis of both main mirrors keeps on being conjugated.

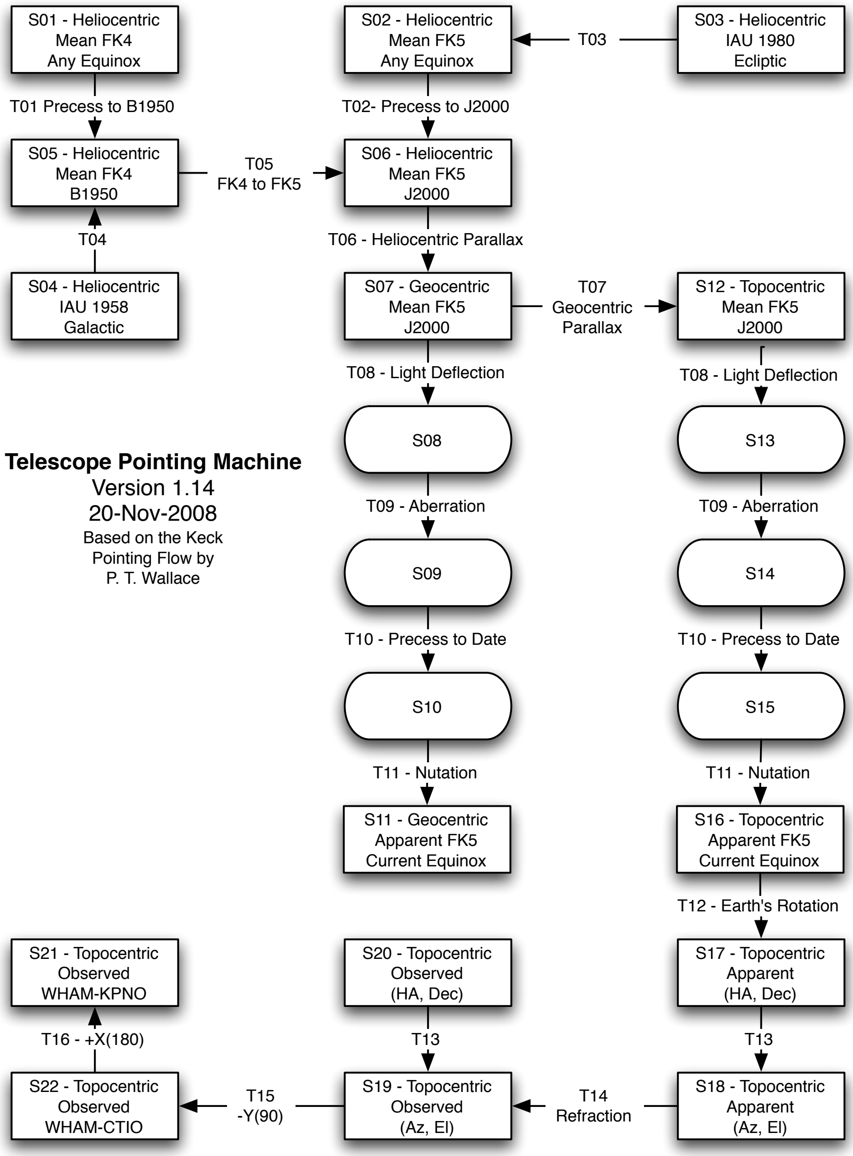

S01 - Heliocentric Mean FK4 Any Equinox
S02 - Heliocentric Mean FK5 Any Equinox
T03
S03 - Heliocentric IAU 1980 Ecliptic
T01 Precess to B1950
T02- Precess to J2000
S05 - Heliocentric Mean FK4 B1950
T05 FK4 to FK5
S06 - Heliocentric Mean FK5 J2000
T04
T06 - Heliocentric Parallax
S04 - Heliocentric IAU 1958 Galactic
S07 - Geocentric Mean FK5 J2000
T07 Geocentric Parallax
S12 - Topocentric Mean FK5 J2000
T08 - Light Deflection
T08 - Light Deflection
S08
S13
Telescope Pointing Machine
Version 1.14
20-Nov-2008
Based on the Keck Pointing Flow by P. T. Wallace
T09 - Aberration
T09 - Aberration
S09
S14
T10 - Precess to Date
T10 - Precess to Date
S10
S15
T11 - Nutation
T11 - Nutation
S11 - Geocentric Apparent FK5 Current Equinox
S16 - Topocentric Apparent FK5 Current Equinox
T12 - Earth's Rotation
S21 - Topocentric Observed WHAM-KPNO
S20 - Topocentric Observed (HA, Dec)
S17 - Topocentric Apparent (HA, Dec)
T16 - +X(180)
T13
T13
S22 - Topocentric Observed WHAM-CTIO
T15 -Y(90)
S19 - Topocentric Observed (Az, El)
T14 Refraction
S18 - Topocentric Apparent (Az, El)

Figure 3.1.5: The tpm state machine

As soon as image misalignment becomes more important than the image of the star affected by athmospherical turbulence (the seeing disk), all telescopes become sensitive to this effect, so approaches use din the case of larger telescopes could be interesting to analyze, and to extend to smaller telescopes. When the MMT (Multiple Mirror Telescope) [20] was changed from a segmented seven-mirror primary arrangement to a 6.5 monolithic one, conceptors were confronted to the positioning of the f/5 secondary mirror issue. The secondary mirror was then placed onto an hexapod (a positioning system with six degrees of freedom) and a devoted control algorithm was used to recenter it depending on the position of the telescope in order to compensate misalignments due to flexures. In a technical memorandum[21] of MMT, a method is defined in order to compensate the mirror positioning as a function of the telescope position. The method consist of thee main steps:

- We place a camera at the prime focus and replacing the secondary mirror by a doughnut shaped counterweight of equal mass.
- We point the telescope at several altitudes and register the variation of the position of the

aimed star related to the encoder position and store it in a flexure model that can be interpolated by a simple polynomial function.

- When the telescope is in normal operation we compensate the flexure tilting and shifting the secondary mirror around its cardinal zero coma point in an open loop mode.

It has been shown that the telescope mount behavior can be assumed as repetitive and how such a correction can drastically improve the general image quality of the instrument. A similar correction can be applied in a quite straightforward way on a smaller telescope like ours, and the model could be done with the same science camera in case we use it in a prime focus configuration.

3.1.4 Optical aberrations sensing in small telescopes

On large aperture telescopes, the simplest and fastest way for wavefront sensing is to use a Shack-Hartmann (SH) sensor or equivalent system at focus, on one side of the scientific sensor. But on small telescopes we cannot be that generous in terms of space or light quantity, which must be carefully managed for several reasons:

- It is hard to install a microlens array or a collimating lens at the same focal plane because we physically don't have enough space. Moreover, these sensors must be close to the optical axis in order to get small out of axis aberration and measure wavefronts undistorted by the sensor arrangement.

- When the Shack-Hartman system can be moved around the science detector in order to find a proper wavefront sensing star, it is very useful to have a very small aperture of the SH (pinhole) in order to reduce the contribution of the sky background and be sure of having a single star in the aperture. Of course the problem is the same lack of practicable space in small telescopes, so we cannot install moving stages close to the focus.

- We cannot use any beam splitter or any device that introduces light losses to the science detector. Light cannot be wasted in instruments where the light gathering capability is limited by the aperture size.

- In a Shack-Hartmann system, the light of a single star is divided into many sub-pupils, thus reducing the number of stars available for wavefront sensing procedures. On a small telescope we could become limited very quickly by the low number of bright enough stars.

As a result a Shack-Hartmann sensor may not be the right solution for such an instrument. That is the reason why we searched a proposed solution in comparable optical configurations where Shack-Hartmann sensors are rather difficult to operate, such as in microscopy. Using this strategy we could isolate several sensorless adaptive optics methods. In our application, these techniques are expected to retrieve the aberrations of the incoming wavefront out of the image registered in the science field by making use of some algorithmics and some measurement strategy.

We will now present a brief overview of the main techniques which could be applied to our system with this purpose is presented in the following.

3.1.4.1 Definition of a modal sensor

Neil & al[22] define a modal sensor using the following concept. A modal sensor is oriented to observe different effects (simultaneously or sequentially) of an image affected by the same unknown wavefront error plus or minus a known bias we include in a certain mode. The observed output related to a certain mode is the difference of the aberrated wavefront plus an offset in the considered mode, and the one minus the offset in the same mode. The output of the wavefront sensor is the difference of the two point source images aberrated by these offsets. Repeating this operation in several modes can give an estimate of the actual wavefront under test. A standard modal sensor configuration is presented in figure 3.1.6

Let's call $\Psi(\vec{r})$ the aberration wave front phase, $\Phi(\vec{r})$ the included offset and $F\{\}$ The Fourier transform operator . We obtain the two following measurements I_1 and I_2 on the sensor:

$$I_1 = F\{exp[j\Psi(\vec{r}) + j\Psi(\vec{r})]\} \tag{3.1.8}$$

$$I_2 = F\{exp[j\Psi(\vec{r}) - j\Psi(\vec{r})]\} \tag{3.1.9}$$

These aberration can then be expressed using a decomposition of orthogonal functions f_n with a or b defining the influence of the mode n and we have:

$$I_{1,2} = \left| \iint\limits_A exp(jaf_k \pm bf_i)\, dA \right|^2 \tag{3.1.10}$$

And the output of the sensor can be expressed as the difference of intensities of the same sensor affected by the positive and negative offset aberration:

$$\Delta I = I_1 - I_2 \tag{3.1.11}$$

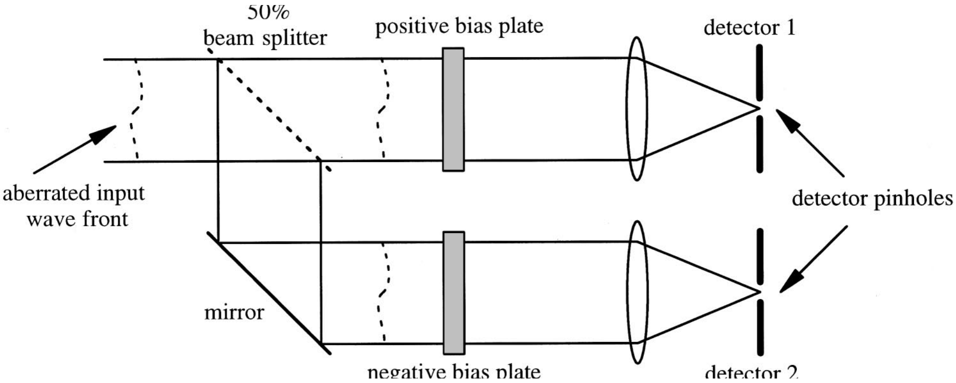


Figure 3.1.6: A standard modal sensor

Then, in order to be able to close the loop properly , we must know which is the sensitivity of a mode n to an actuation of a mode m in order to be aware of any eventual crossover degeneracy. In fact, depending on the modes we use to decompose the signal in these orthogonal functions we may not assure that an actuation on a mode does not affect another one. A sensitivity matrix can be generated getting the gradient of the output of the sensor on a mode n with respect to the input aberration when this aberration a is close to zero.

$$S_{n,a} = \left.\frac{\partial \Delta I_n}{\partial a}\right|_{a=0} \tag{3.1.12}$$

3.1.4.2 Using defocused offset images to retrieve low-order Zernike coefficients

For the design of the MEGACAM Imager McLeod & al.[23] decided to use two guiding sensors on each side of the science sensor array. The guide sensors are used so as to get always a window around a single star in its field, making the readout very fast. Typically, the readout time is 25s in binning 1:1 for the science array and only about a tenth of a second for a windowed box round a guide star. That way we can take high rate guiding frames on both guiding arrays sending corrections of pointing and rotation to the telescope at the same frame rate while the science array is exposing for several minutes.
MEGACAM is initially loaded with Sloan u',g' r', i',z' filter set. Each filter covers the full science array while each guide sensor is covered by a transparent compensation plate as can be seen in Fig. 3.1.7. Both compensation plates are designed for each filter in order to have a small defocus aberration relative to the science array, one of them in the positive sense and the other one in the negative sense.

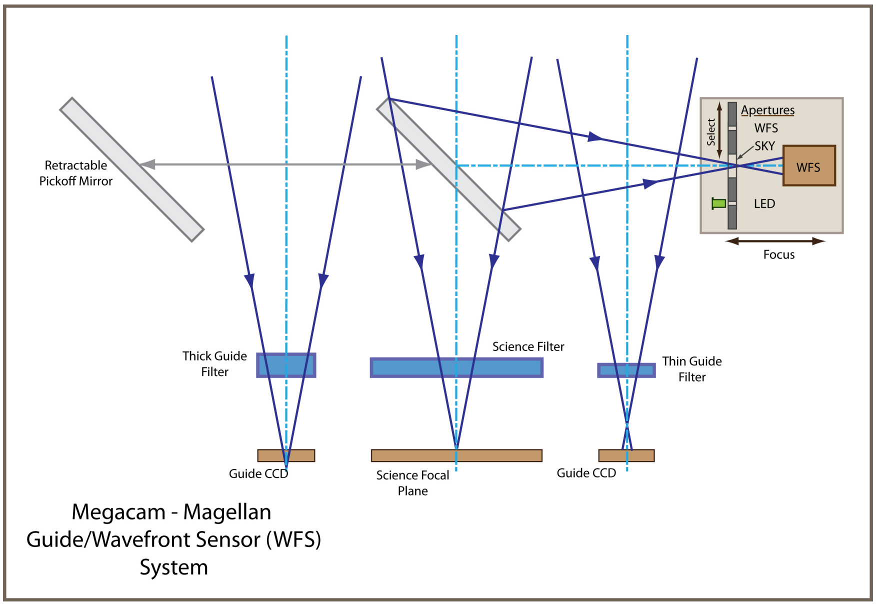


Figure 3.1.7: Wavefront sensing on Megacam focal plane

This becomes very helpful when we want to use the guide chips as modal sensors, as described in the previous subsection. We only need to scale both guide stars, and to make the difference of the two windows we obtain the same output as defined in equation 3.1.11. Figure 3.1.8 shows the output of the modal sensor when the optical path is affected by several types of basic aberrations. It shows that we can easily measure low-order Zernike coefficients, from defocus to astigmatism, coma or sphericity, that could be used to send corrections of these modes to the secondary mirror positioning system, and some primary mirror actuators. Defocus information can basicaly be retrieved from the sum of the difference image while the higher order aberrations must be fitted using the shape of the difference image.
In practice, in the MEGACAM imager it appears that we only need to use the defocus guiding information because the telescope has an embedded Shack-Hartman wavefront sensor which runs hourly coupled to the primary mirror server, which interpolates the astigmatism with a FEM analysis of the primary mirror deformation due to changes in its inclination. This approach yields good enough results, although it is hardly applicable to a small telescope.

3.1.4.3 Including specific known aberrations

D.Debarre[24] showed that it was possible to close an adaptive optics loop based only on the information we got directly on the imaging sensor. That means aberrations can be determined without any a priori knowledge related to the image, or to pinhole measurement data.

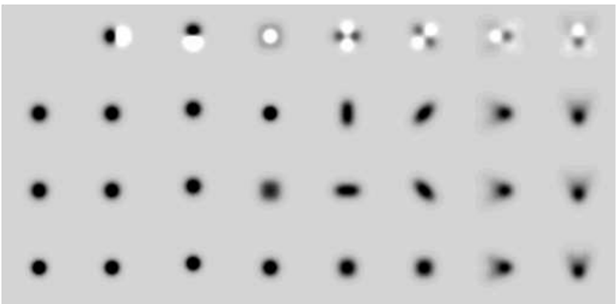


Figure 3.1.8: Using the Megacam Guide sensors as modal sensors to retrieve information about astigmatism coma and sphericity

As shown in Fig 3.1.9 Debarre's set-up is quite simple. It consists of re-imaging the pupil on a deformable mirror and then sending it to the sensor. The imaging sensor is therefore affected by the unknown aberration that arrives at the entry of the system plus the known aberration included in the deformable mirror.

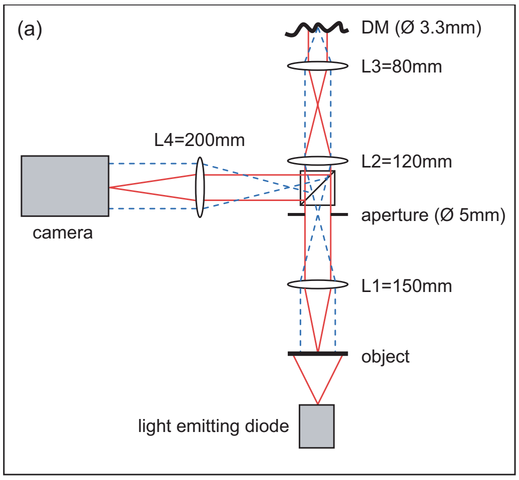


Figure 3.1.9: D. Debarre setup for optimizing a wavefront based on direct image measurement and a deformable mirror

D. Debarre showed that if the included modal offsets by the deformable mirror are decomposed in Lukosz modes instead of classical Zernike modes, their sensitivity related to the low frequencies of the image becomes decoupled. In other words, the computation of the sensitivity matrix of the equation 3.1.12 gives a diagonal matrix.
Or, in other words, if we take in mind the three following restrictions:

- The actuation on the deformable mirror follows a Lukosz decomposition
- We consider only the lowest frequencies of the Fourier transform of the obtained images
- We try to maximize the sum of these lower frequencies using a bilinear interpolation of one start point with two known offsets on each mode

In that case we can observe the following relations:

$$S_{n,a}\big|_{n=a} = 1 \tag{3.1.13}$$
$$S_{n,a}\big|_{n\neq a} = 0 \tag{3.1.14}$$

meaning that every mode is linearly independent of the others in the linear decomposition so they do not contaminate each other. Making 2N+1 measurements based on a start point plus 2 known offsets in each mode we want to maximize, we can make a bilinear interpolation and thus find the N-dimensional point where the sum of the lower frequencies is maximum. This N-dimensional point represents the values to be used for the set of modes we want to optimize. In other words, if we simplify the problem to the optimization of one single mode, we can call the base position affected by un unknown aberration G_0, and the same position affected of a known offset b in this mode are called G_+ G_-. The optimized G_{Best} position in this mode will be so far:

$$G_{Best} = b\frac{G_+ - G_-}{2G_+ - 4G_0 + 2G_-} \tag{3.1.15}$$

An example of this particular case is shown Fig.3.1.10.

On the first row of figure 3.1.10 we may find white noise images, on the second row we find their respective Fourier transforms, and in the third row we find the sum of the low frequencies $g(a)$. The first column represents the initial aberrated measurement, the second and third represents the measurement affected of positive and negative offset and the last column represents the measurement of the best position according to equation 3.1.15. The best focus position is directly estimated out of the originally aberrated image plus two images with offset aberration values.

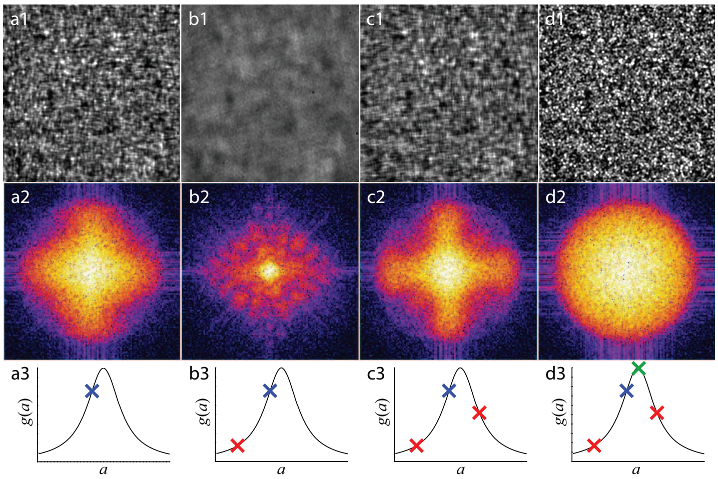


Figure 3.1.10: Example of optimization for a single mode.

3.1.4.4 Direct star measurement

It is also possible to measure some aberrations directly measuring the shape of a point source. Since the telescope is pointing at stars, we can easily find a bunch of point sources all over the field. Grisan & Naletto[25][26] demonstrated in 2007 that we can simply measure and minimize some aberrations on the image of a point source whenever this one is sampled enough onto the detector.
Grisan & Naletto used an iterative method to minimize and detect the aberrations one by one starting by the spherical and sending correction until no further improvement could be measured, then proceeded with coma, defocus and astigmatism. An example is shown in Fig 3.1.11. For the optimization of each mode, a box is drawn around the point source we want to analyse and assigned a merit function. The aberration is then measured and a correction is sent to the deformable mirror according to this measured aberration. Here we give some examples of these merit functions.

For spherical aberration, as a first step we can maximize the overall intensity of the box as in function M_1 and then maximize the mean intensity of the non zero pixels as in function M_2.

$$M_1 = \iint_{Area} I(x,y)\,dxdy \tag{3.1.16}$$

$$M_2 = \frac{1}{N_{I(x,y)>0}} \iint_{I(x,y)>0} I(x,y)\,dxdy \tag{3.1.17}$$

For coma, we can define the merit function M_3 as follows: Let's call P_1 the unweighed barycentre of the nonzero pixels in the box and P_2 the barycenter of the non zero pixels weighted by their respective intensities, then we have:

$$M_3 = \left(1 - \iint_{Area} I^2(x,y)\,dxdy\right) \left|\left|P_2 - P_1\right|\right|_2 \tag{3.1.18}$$

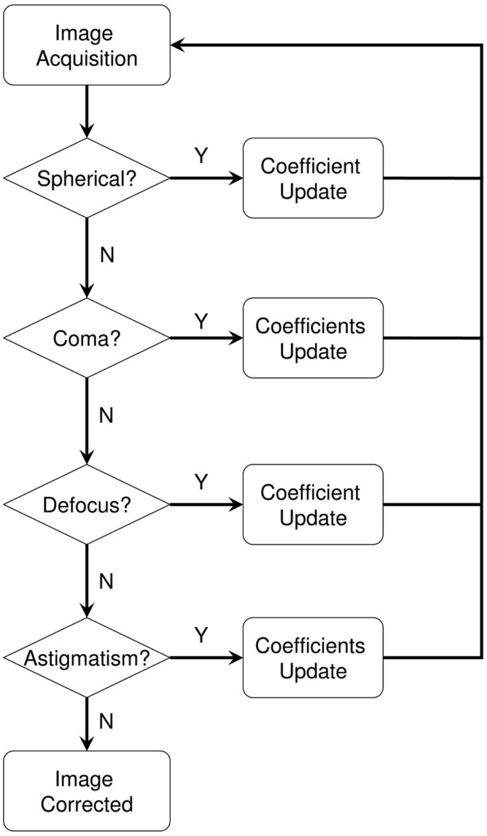


Figure 3.1.11: Grisan & Naletto sequential loop for optimizing aberration using point sources

3.1.4.5 The Roddier imaging test

Claude and Francois Roddier[27] published in 1993 a method for analyzing two out of focus images and retrieve the wavefront. This method is based on long exposure defocused images and can be used only for optimizing the deformations of the mirror so far. (it cannot be used for fast steering adaptive optics systems).
The method consists of taking two out of focus images centered on the same star, one having a positive defocus of a distance l and the other image with a negative defocus of the same amount.If we define $I_1(\rho,\theta)$ the intensity of the pixel of coordinates (ρ,θ) expressed in polar coordinates on the image with the positive defocus and $I_2(\rho,\theta)$ the intensity of the same pixel in the image with the negative defocus, the output of the sensor is defined by the equation 3.1.19:

$$S(\rho,\theta) = \frac{I_1(\rho,\theta)-I_2(\rho,\theta+\pi)}{I_1(\rho,\theta)+I_2(\rho,\theta+\pi)} \tag{3.1.19}$$

It is interesting to note that due to the inversion of the image through focus, the conjugate pixel of the pixel in the image before focus is rotated by an angle π around the optical axis in the image after the focus as we can see in figure 3.1.12 Then, we can find a relationship between the sensor output S and the actual wavefront. According to Roddier[28], the irradiance transport pupil equation yields to equation 3.1.20:

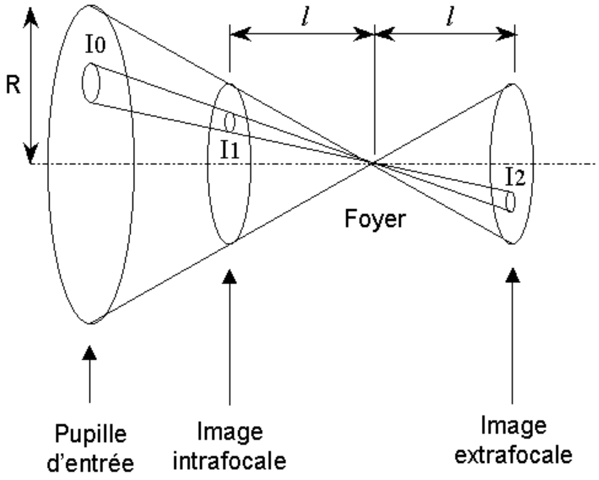


Figure 3.1.12: Relationship between the wavefront and defocused images

$$S(x,y) = \frac{\partial W}{\partial n}\delta_c - P\nabla^2 W\Delta z \tag{3.1.20}$$

where:

- x, y are the cartesian coordinates in the focal plane
- W is the wavefront surface
- n is a normal vector perpendicular to the edge of the pupil
- δ_c is a Dirac distribution equals to one on the edge of the pupil and zero everywhere else
- P is a function equals to 1 inside the pupil and zero outside
- ∇ is the operator $\frac{\partial}{\partial x}, \frac{\partial}{\partial y}$

According to Newton's law , we can add that:

$$\Delta z = \frac{f(f-l)}{l} \tag{3.1.21}$$

l being the distance bewteen the focal plane and the image of the cross section of the beam through the telescope. Inserting equation 3.1.21 into equation 3.1.20 yields to :

$$S(x,y) = \frac{f(f-l)}{l}\left(\frac{\partial W}{\partial n}\delta_c - P\nabla^2 W\right) \tag{3.1.22}$$

Roddier[27] implemented an iterative solution to this equation that would give a precise estimation of the wavefront. The convergence of the algorithm is fast so it can be used in closed-loop applications. The algorithm was tested on the ESO New Technology Telescope (NTT) at La Silla Observatory actuating on the primary mirror to correct the wavefront deformation. Fig.3.1.13 shows the effect of the algorithm for reducing coma on a star image. It can be used for collimation in our system.

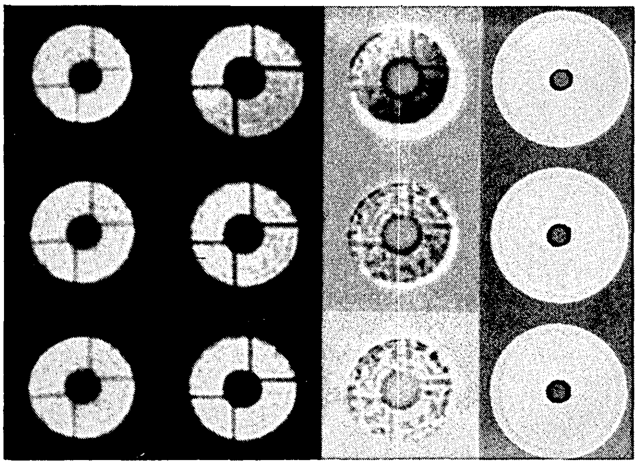


Figure 3.1.13: Reduction of the coma term with the Roddier algorithm on the ESO NTT. From left to right: inside focus, image, outside focus image, sensor signal and domain boundaries. From top to bottom, initial image, first iteration and second iteration.

3.1.4.6 Zernike annular polynomials

Zernike polynomials[29] have been used for a long time for defining optical systems. They consist of a set of orthogonal polynomials where every polynomial can be related to a specific optical aberration. They sometimes are separated in odd and even polynomials as we can see in equations 3.1.23–3.1.25.

$$\left.\begin{aligned} Z_{even_j} &= \sqrt{n+1}R_n^m(r)\sqrt{2}\cos m\theta \\ Z_{odd_j} &= \sqrt{n+1}R_n^m(r)\sqrt{2}\sin m\theta \end{aligned}\right\} \quad \text{for } m \neq 0 \tag{3.1.23}$$

$$Z_j = \sqrt{n+1}R_n^0(r) \quad \text{for } m = 0 \tag{3.1.24}$$

$$\text{where } R_n^m(r) = \sum_{s=0}^{\frac{n-m}{2}} \frac{(-1)^s(n-s)!}{s!\left(\frac{n+m}{2}-s\right)!\left(\frac{n-m}{2}-s\right)!} r^{n-2s} \tag{3.1.25}$$

The values of n and m are always integral and satisfy $m \leq n$, $n - |m| =$ even. The index j is a mode ordering number and is a function of n and m. Every randomly aberrated wavefront $\phi(r,\theta)$ of aperture radius R can be expressed as a ponderated combination of these polynomials as shown in equation 3.1.26.

$$\phi(\rho R, \theta) = \sum_j a_j Z_j(\rho,\theta) \tag{3.1.26}$$

With $\rho = r/R$. Zernike polynomials have been widely used in fields related to astronomical adaptive optics for several reasons developed by Robert J. Noll[30]:

- The first few terms of Zernikes functions express optical aberrations familiar to opticians as they may be related to classical third-order Seidel aberrations (tilt, defocus coma, sphericity, astigmatism, etc.)
- An analytical solution of the wavefront can be found provided we know the derivative of the wavefront, which is the resut we obtain in a Shack-Hartmann wavefront sensor
- An analytical solution of the residuals of any wavefront can be found
- When the wavefront is affected by Kolmogorov turbulence, Zernike polynomials permit to evaluate the solution integrals in a non-iterative process.

The main problem of Zernike polynomials when applied to astronomy is the fact that most of the telescopes are affected by a central obstruction due to the secondary mirror, which gives an annular pupil, different from the circular pupil domain where Zernike polynomials are defined. On the other hand, Zernike polynomials are orthogonal only on a circular pupil, so relevant mode crosstalk appears in annular pupils.
For that reason, Mahajan[31] developed a set of polynomials comparable to Zernike ones, with the property of taking into account the central obstruction and being orthogonal over an annular pupil. If we consider r as the pupil radius normalized to one and ϵ the obstruction factor, the Zernike annular polynomials can be written as :

$$U_n^m(r,\epsilon) = \left\{ \begin{array}{ll} \left.\begin{array}{ll} \sqrt{2(n+1)}R_n^m(r,\epsilon)\cos(m\theta) & m>0 \\ \sqrt{2(n+1)}R_n^m(r,\epsilon)\sin(-m\theta) & m<0 \end{array}\right\} & m \neq 0 \\ \sqrt{n+1}R_n^m(r,\epsilon) & m=0 \end{array}\right. \tag{3.1.27}$$

When $m=0$, the radial polynomials are given by:

$$R_{2j}^0(r,\epsilon) = R_{2j}^0\sqrt{\frac{r^2-\epsilon^2}{1-\epsilon^2}} \tag{3.1.28}$$

When $m>0$, the radial polynomials can be obtained recursively from the following equations 3.1.29–3.1.33

$$R_{2j+m}^m(r,\epsilon) = \sqrt{\frac{1-\epsilon^2}{2(2j+m+1)h_j^m}}\, r^m Q_j^m(r^2) \tag{3.1.29}$$

with

$$Q_j^m(r^2) = \frac{2(2j+2m-1)}{(j+m)(1-\epsilon^2)}\frac{h_j^{m-1}}{Q_j^{m-1}(0)}\sum_{i=0}^{j}\frac{Q_i^{m-1}(0)Q_i^{m-1}(r^2)}{h_i^{m-1}} \tag{3.1.30}$$

$$h_j^m = -\frac{2(2j+2m-1)}{(j+m)(1-\epsilon^2)}\frac{Q_{j+1}^{m-1}(0)}{Q_j^{m-1}(0)}h_j^{m-1} \tag{3.1.31}$$

$$Q_j^0(r^2) = R_{2j}^0(r,\epsilon) \tag{3.1.32}$$

$$h_j^0 = \frac{1-\epsilon^2}{2(2j+1)} \tag{3.1.33}$$

$$\text{when } m<0,\ R_{2j+m}^m(r,\epsilon) = R_{2j+m}^{|m|}(r,\epsilon) \tag{3.1.34}$$

$$\text{when } m=n,\ R_n^n(r,\epsilon) = \frac{r^n}{\sqrt{\sum\limits_{i=0}^{n}\epsilon^{2i}}} \tag{3.1.35}$$

In 2001, Restaino & al.[32] used a Roddier imaging method to test the surface quality of the Naval Observatory Flagstaff Station 1m telescope. Using the same sensor output, which is the difference of two out of focus images, they computed the wavefront aberration decomposing the signal into normal circular Zernikes as a fist approximation and then with annular Zernikes. The comparison of the results obtained from the same raw images show that if the obstruction ratio is large, the circular Zernike decomposition suffers a degeneracy compared to annular Zernikes. They concluded that using annular Zernikes would be more precise at the moment of analyzing the absolute aberration of the optical system. On the other hand,they affirm that in adaptive optics this degeneracy wouldn't be a particularly important effect as typically the wavefront analysis does not use the Zernike modes explicitly but rather sets the deformable element to null out the wavefront error. Dai & Mahajan[33] confirmed later that even if the error included using standard Zernikes instead of annular Zernikes would be dominated by the turbulence, it is interesting to use annular Zernikes to have a better estimation of the image degradation.

3.1.4.7 Other proposals

There are a number of other proposals of interest with the same purpose. M.J. Booth[34] uses an iterative method to optimize the PSF. Booth does not use a sequential optimization but the optimization strategy is based on a downhill simplex. For optimizing N zernike modes , we take N+1 measurements around a start point and a different offset on a separate mode in the others. The related merit function is based on the Strehl intensity defined as the ratio of the sum of the intensities of the measured image by the sum of the intensity of an image that would not be affected by any aberration:

$$M_4 = \frac{\iint_{Area} I(x,y)_{Measured}\, dxdy}{\iint_{Area} I(x,y)_{Ideal}\, dxdy} \tag{3.1.36}$$

Maeda[35] developed algorithms for optimizing a wavefront using a direct measurement of the PSF, without including offsets. Maeda[35] directly recovered the wavefront using a method defined by Gerchberg and Saxton[36][37]. The idea is to fit a model to find the best set of aberration for the measured PSF, then according to the measured phase aberration, we can give the corresponding correction on the actuator. Ideally no iteration is necessary and the correction is very fast but a high level of sampling is needed in order to get a proper estimation of the pupil.

3.2 Hardware implementations

Although software and algorithmics can drastically improve the positioning , tracking and image quality feedback as we saw in the previous section, it is important to prevent problems at the hardware level in order to avoid the need of correcting them.

3.2.1 Telescope mechanics

Telescopes generally use designs for the optical configuration which are independent of its mechanical configuration. Though the optical setups are well-known in general, mechanical designs and mounts are equally relevant regarding the final performance of the telescope, as the pointing accuracy of the telescope is directly dependent on them.

3.2.1.1 Active collimation

In order to obtain proper in focus image, the alignment of the mirrors and the detector of the telescope is crucial independantly from the telescope configuration. Effects of misalignments of the mirrors of a telescope can be analitically calculated [38] and we can see that even for small telescopes the flexure of the tube depending on where the telescope is pointing at can degrade the image quality.
Most large telescopes are equipped with active optics systems cabable of compensing the shape and positioning of the mirrors depening on the position of the telescope itself using a shack-hartmann sensor but the cost of such repositionable system with the required precision makes it out of budget of telescopes smaller than 1m in diameter.
In these telescopes the collimation is usually done once at a median height position on the sky in order to minimize the overall field aberrations. Krugler presented an optimum technique [39] for

aligning Ritchey-Chretien telescopes using a theodolite.
McLeod[40] presented a technique minimizing field coma and astigmatism for small wide field telescopes that allows to find a best position for a median telescope altitude in a few visual iterations of moving tilting screws behind the secondary mirror.
The main property of a cassegrain or Ritchey-Chretien telescope is that there is an infinity of configuration giving an astigmatism-free image on axis. Any position where the secondary mirror is tilted and shifted around a virtual point call the coma-free point situated on the optical axis of the telescope will give an image with no astigmatism on the axis of the telescope. On the other hand, only when both mirrors are perfectly collimated the field astigmatism will be minimized with a radial profile. In a cassegrain configuration, the Coma free point Z is located on the secondary axis at a distance L_0 behind the vertex expressed in equation(3.2.1). Where R_s is the secondary vertex radius of curvature, m is the system magnification and k_2 is the conic constant for the secondary.

$$L_0 = \frac{R_s(m+1)}{m+1-k_2(m-1)} \tag{3.2.1}$$

For this reason it is necessary to be able to move the secondary mirror with 5 degrees of freedom (two rotations , x tilt and y tilt and three tranlations) in order to be able to align the optics in an optimal way.

3.2.1.2 Parallel platforms

Historically, open loop serial chain manipulators were presented first because of their simplicity of use and manoevrability similar to a human arm. On the other hand for high load aplications paralel systems look way more suitable because of the higher stregth and the lower sensitivity to vibrations. The main particularity of parallel manipulators is that since it is necessary to move every parallel axis at the same time the whole system cannot be run in open loop mode and a close loop must be implemented.
Stewart [41] designed in 1965 a 6-DOF parallel mechanism to simulate flight conditions for helicopter pilots training by generating general motion in space. This design as seen in fig(3.2.1) consists of a triangular platforms linked to three controllable linear acutators with ball joints. Each actuator is linked to the ground by an indepentant two axes rotary actuator, one of these axes is controllable while the other is free. Among the numerous answer to the Stewart's paper,

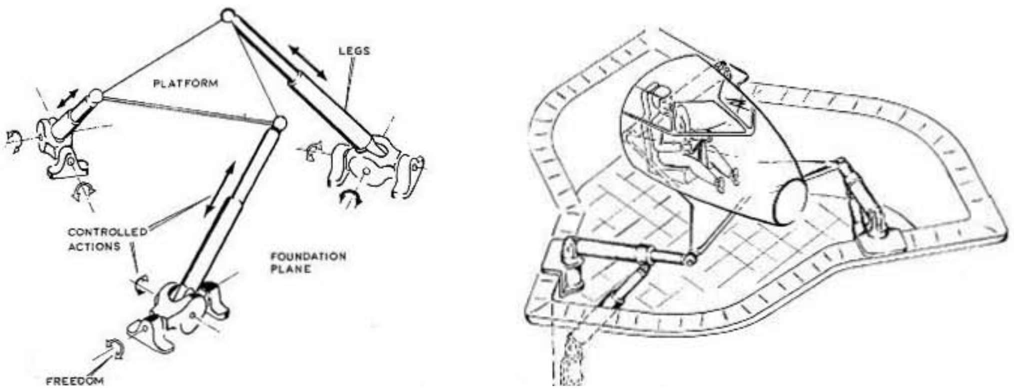


Figure 3.2.1: Original Stewart platform and application for a flight simulator

Gough proposed the use of 6 controllable linear actuators linket to both platforms by CV joints similar to the tyre testing machine he developped in 1958 with Withehall [42]. By removing the unnecessary internal degrees of freedom the Stewart's plateform becomes then a fully parallel system.
Later in 1978 Hunt [43] unified the previous ideas and proposed a first kinematic and dynamic

analisis for the application of the plateform into robotics. Although Gough's system is much closer than Stewart's, the name of Stewart platform will remain in further aplications. Some authors will refer to it as Stewart-Gough plateform later.

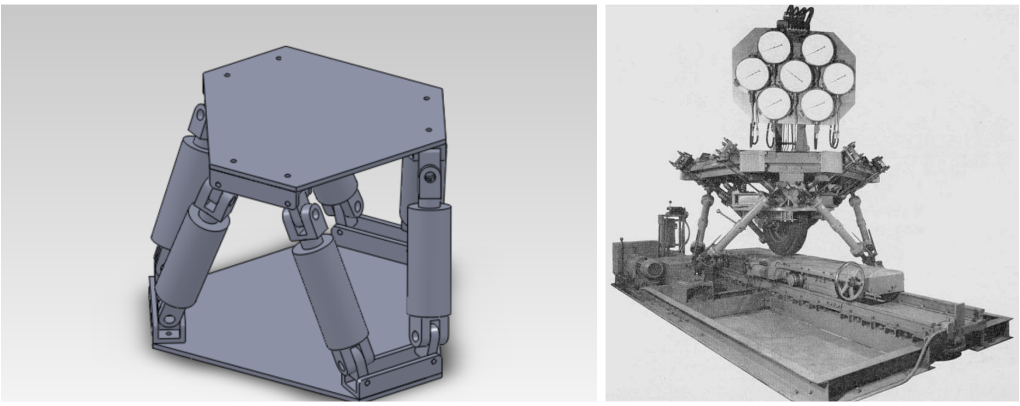


Figure 3.2.2: Implementation of the Gough platform in the tyre testing machine and modification of the Stewart's platform according to Gough comments by Hunt using CV joints

Generic Stewart platform kinematics is comonly defined as follows and as we can see in Fig(3.2.3):

- We consider n legs $[L_1...L_i...L_n]$ of lrespective lenghs $[l_1...l_i...l_n]$
- Attach point on the base of the leg L_i is defined by the vector $\vec{b_i} = \overrightarrow{O_1B_i}$ in the referential (O_1, R_{Base})
- Attach point on the platform of the leg L_i is defined by the vector $\vec{p_i} = \overrightarrow{O_2P_i}$ in the referential $(O_2, R_{Platform})$
- The translation vector $\vec{T}$ between the referential R_{Base} and the referential $R_{Platform}$ is defined by $\vec{T} = \overrightarrow{O_1O_2}$
- The rotation between the two referentials is represented by Euler's angles ϕ (Rotation around x axis), θ (Rotation around y axis) and ψ (Rotation around z axis).

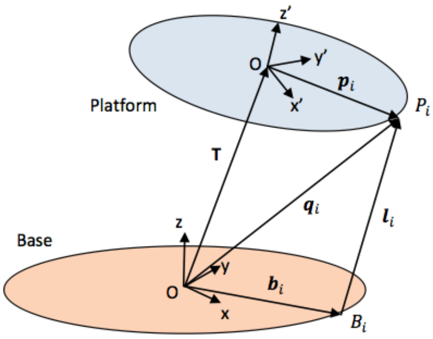


Figure 3.2.3: Hexapod kinematics

We can then compute the lengh of each leg L_i according to equation (3.2.2).

$$l_i = \left\| \vec{T} + \mathbf{R_B} \cdot \vec{p_i} - \vec{b_i} \right\|_2 \tag{3.2.2}$$

$$\tag{3.2.3}$$

where:

$$\mathbf{R_B} = \mathbf{R_z}(\psi) \cdot \mathbf{R_y}(\theta) \cdot \mathbf{R_x}(\phi) \tag{3.2.4}$$

$$\mathbf{R_B} = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos phi + \cos\psi\sin\theta\sin\phi & \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi \\ \sin\psi\cos\theta & \cos\psi\sin phi + \sin\psi\sin\theta\sin\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{3.2.5}$$

Even if control and dynamics of the Stewart platform was widely studied in 80's and 90's decade the first application such device for positioning of a secondary mirror appears with the upgrade from six primary mirrors to one monolithic primaray mirror of the MMT (Multiple Mirror Telescope) at Mt Hopkins and of the design of the LBT (Large Binocular Telescope) at Mt Graham [44]. A set of six secondary mirrors of different optical configuration focal ratio and aplication was designed for these two telescopes in order to fulfill different functions at each telescope. A similar support was designed at LBT and MMT as seen in Fig.(3.2.4) in order to be able to make coincident primary and secondary optical axese implemented a Stewart platfromed also called Hexapod. This technical solution allows precise and vibrationless positioning over 6 axes of a heavy weight at such a hanging position.

Starting from this period every large diameter telescopes of more than 4m diameter such as VLT[45], VST[46], VISTA[47], Magellan[48],GTC[49],GMT[50] and more were designed using a parallel manipulator for positioning and collimation of the secondary mirror.

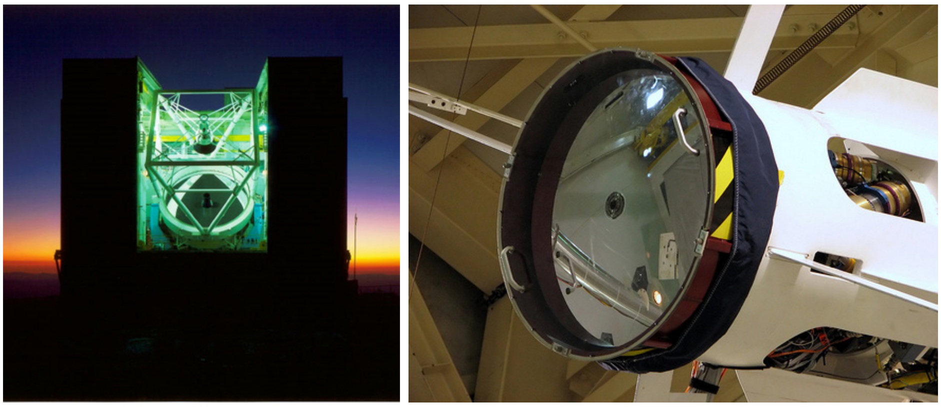


Figure 3.2.4: MMT 6.5m telescope and detail of adaptive secondary mirror and its Stewart platform support

However, the 1m telescopes class are usually equipped with a non controlled fixed secondary which can me only moved manually. The main reason for this is that the needed positioning precision of the hexapod is driven by the focal ratio rather than the mirror diameter and thus it remains independant of the size of the telescope, as a result the relative cost of the secondary mirror positionner compared to the overall costs of the project becomes bigger and bigger while the telescope size decreases. As a result for the 1m telescope class the cost of such hexapod is prohibitively expensive compared to the total cost of the instrument.

3.2.2 Hardware solution for high precision positioning

The main concern in telescope positioning is the wide dynamical range of speeds of the system. It must be able to track at very slow speed with high precision (a tenth of an arcsecond) while the movements between a target and the next one must be as fast as posible and therefore implies high slewing speeds up to 5 to 10 deg/s

3.2.2.1 High precision wormgears

Worm gears have traditionally been used in telescopes drives because of their important reduction ratio in a single mechanical link. Ratios from 1:100 to 1:500 are commonly used.
Unfortunately worm gears have the following drawbacks:

- Gearing efficiency is low
- They must be lubricated which implies regular routinary operation to clean and lubricate, in presence of sand, the grease becomes abrasive, and at low operating temperatures the grease trends to increase its viscosity and lowers the efficency even more.
- Manufacturing and alignment precision of both worm and gear must be extremely high in order to minimize the presence periodic errors and backlashes.
- A good gear precision implies the use of bigger wheels which trend to be an important cost of the relative budget of construction of the telescope.
- Even when manufacturing and alignment are of very high precision, a periodic error of a few arcseconds peak to peak always remains, which is not sufficient for tracking star blindly whithout observing drift at exposures longer thant one worm full rotation (typically a few mintes of time)
- With temperature variation, the material of the gear will shrink or expand and will make the pressure of the worm against the gear to change, affecting this way the drive efficiency or the presence of backlashes.
- This latest problem can be compensated in the design by spring loading the worm against the gear but this solution may introduce a lack of stiffness of the whole system.

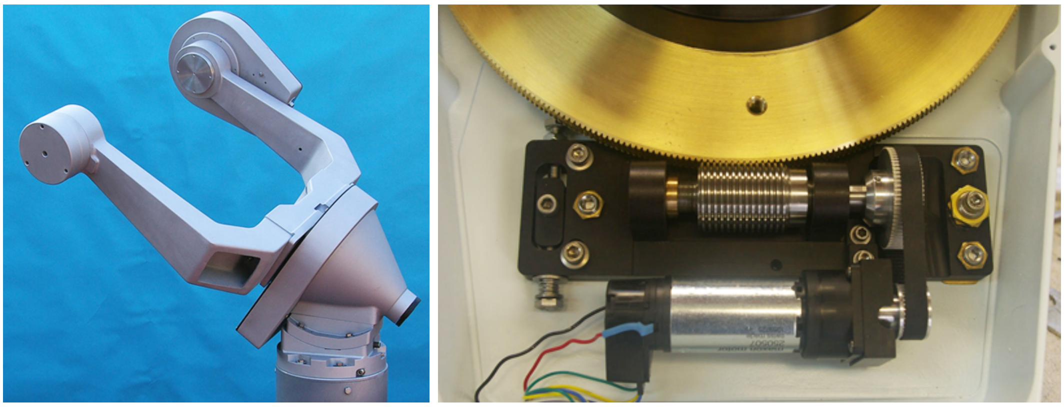


Figure 3.2.5: A comercial mount and its worm drive

Effects of periodical errors due to worm gearing have been known for the beginning of the 20th century [51], the first attempts to correct for the effects can be found in the 1960's. Kron first proposed to slightly decenter on end of the worm in its bearing to produce a counter periodic error[52] but only a first harmonic can be corrected using this technique.
Hardie and Ballard [53] proposed in 1962 the use of several gears between the motor and the worm. These gears would have an adjustable excentricity. By adjusting the number ratio phase and excentricity of each gear, they managed to mechanically correct for several harmonics of periodical error by introducing a known aberration that would compensate the exisiting worm error. The result of the effect of their technique is shown in Fig.3.2.6 which is a scan of a photographic plate obtained by setting a non null speed in declination while the telescope is tracking on a star. The effect of the drift due to speed variations can be appreciated with and without the correction.

A more direct solution to reduce the worm error is to obtain a better transmission from a begining. This way we can avoid to correct for the periodic errors. Groenveld [54] analyzed in 1969 the use of non-standard worm-gear shapes in order to optimze the contact surface which would minimize the variation errors at extremely low rotation speeds used in telescopes. On way to maximize the contact surface between the teeth and the worm pitch is to use smaller pressure angles than the standard ones. An illustration of the definition of the pressure angle is detailed in Fig.3.2.7. In the standard industry where the precision at very low speed is not that important

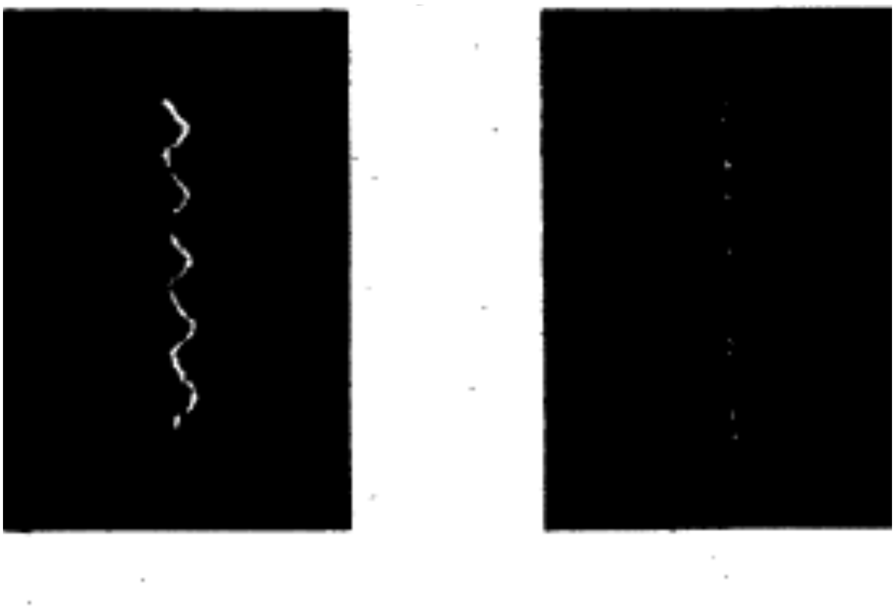


Figure 3.2.6: Enlargements of stellar images trailed in declination to reveail periodical drive error. Left. Original error, right: error reduced by compensation.

the commonly used angles are 14.5, 20 and 25 degrees. These relatively high values minimize the contact surface in order to increase the system efficiency. In our case the power is not a limitation so the contrary may be interesting in the sense that increasing the surface of contact make the micrometric variation to statistically compensate together. The overal movement in this case is then smoother. In order to obtain better transmission while keeping a good manufacturability, Groenveld proposed the use of 10 degrees pressure angles for the construction of a telescope at Mt Stromlo Observatory in Australia.

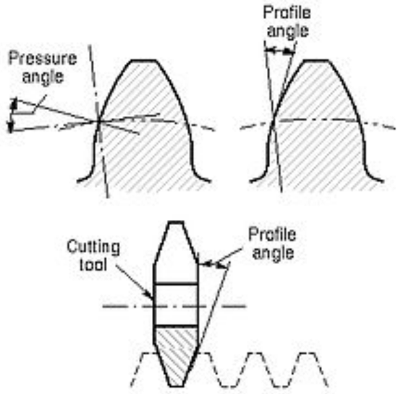


Figure 3.2.7: Definition of the pressure angle

3.2.2.2 Backlash-free drives

A common issue affecting the positioning precision of the drives is the presence of backlash in the gears. This results in a hysteresis effect observed when moving the axis back and forward, resulting in an important uncertaincy on the exact position of the telescope optics on the sky. This is why some solutions used to increase the precision of the telescope drives is the implementation of backlash-free gearing.

Several ways can be used to remove the backlash of a gear, in the case of an horizontal gear, the most common way is to split the gear horizontally as shown in Fig.3.2.8 and offset both parts by an angle removing this way any possible backlash. This technique as defined in [55] is efficient but increase the friction between both gears by a considerable amount.

Additionnaly the transmission ratio needed in a telescopes between the motor and the main axis it typically between 100X and 400X which cannot be achived with only two of these drives. As a result a minimum of 3 to 5 stages of such gearing would be necessary to complete the transmission ratio multiplicating also the friction level. This solution is commonly used for single reduction stages between the motor and the worm but is not suitable for the complete transmission chain.

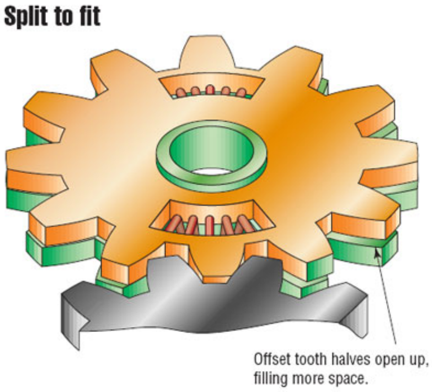


Figure 3.2.8: backlash compensated horizontal gear

In order to remove backlash in high reduction stages such as worm gears one solution is to split the worm into two parts as detailed in figure 3.2.9 and adjust the space between the two parts so that there is no gap remaining between the worm and the gear theeths. Doing this way it is possible not only to reduce the backlash but also to increase the drive precision by reducing the periodic errors.
On the other hand, this solution is quite costly since a more complex worm mechanics has to be designed with a high precision. Moreover it also increases the friction factor affecting negatively the efficienty of the gearing.

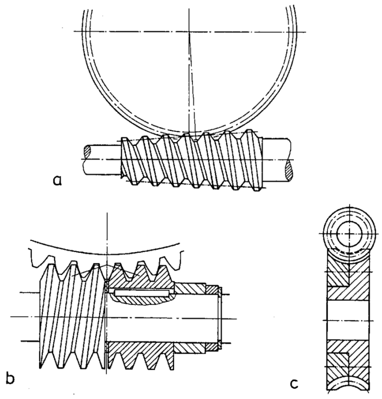


Figure 3.2.9: Duplex driven worm

The most commonly used solution so far is the use of a preload cable and weigh on each axis or, as defined in [56], where a single cable passes through both gears without any need to pass through the main axis of the telescope. Fig.3.2.10 shows how Hannel preloaded a 1m telescope at La Silla Observatory. The telescope has to be correctely balanced before installing the cable and the weight and the torque introduced by the weight unbalances sligthely each axis so that the gear teeth always pushes the worm on the same side, removing this way any possible backlash. The weight hung to the cable can be adjusted so that is is sufficiently heavy to unbalance the telescope but with the minimum amount so that the gear efficiency and the friction bewteen the worm and the gear would be affecteed by a neglectable amount.

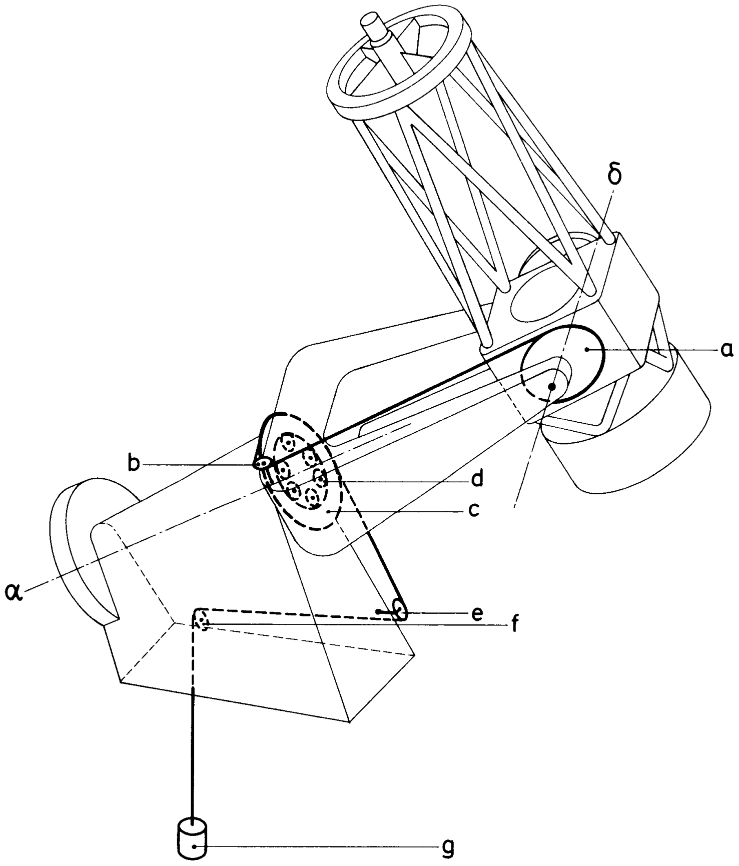


Figure 3.2.10: Single cable preloaded telescope

3.2.2.3 High resolution encoders

Positioning and tracking of the telescope can be obtained using a closed control position control loop. Therefore, a better way to reduce the positioning errors can be achieved by having a faster and better feedback of the actual position of the drive axis. If the actual position is properly known, it becomes easier to send corrections to the motors in order to avoid the telescope from deviating form its target. One way to achieve this is by installing industrial absolute belts and measureing heads such as the ones provided by Renishaw or Heidenhain where a magnetic belt is installed on a precisely machined ring fixed to the mobile part on the axis and one or multiple readheads collect information of the actual position as shown in Fig.3.2.11 . These systems which implementation is shown in Fig.3.2.12 . It consists in a scale printed on a tape glued on cicular rotary part. An LED illuminates the scale and the reflected light passes through a grating acting as a Vernier. The light pattern in then registerd on a photodetector and its analysis allows to

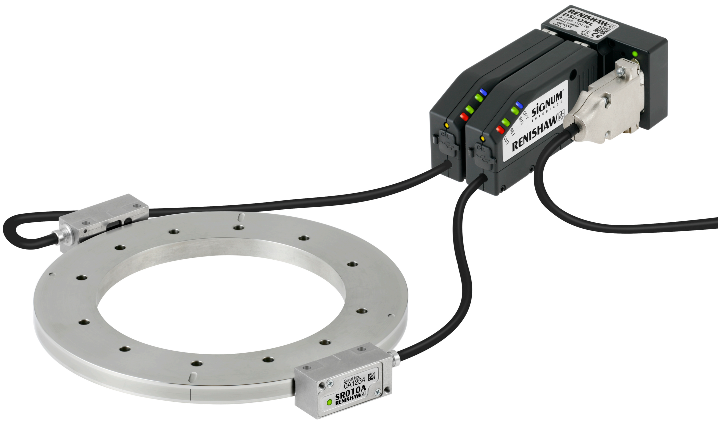


Figure 3.2.11: Renishaw tape encoder and readhead configuration

identify precisely the positioning of the tape relatively to the readhead . An absolute positionning is possible by using various contiguous scales printed on the tape.
This solution has shown to be extremely precise and reliable and has been used in many large profesional telescopes such as the William Hershel Telescope which is a 4.2m telescope installed at La Palama in 1985. The TCS upgrade was performed in 1993 [57] and the first use of such encoders compared to the epoch techniques demostrated the viability of such hardware.

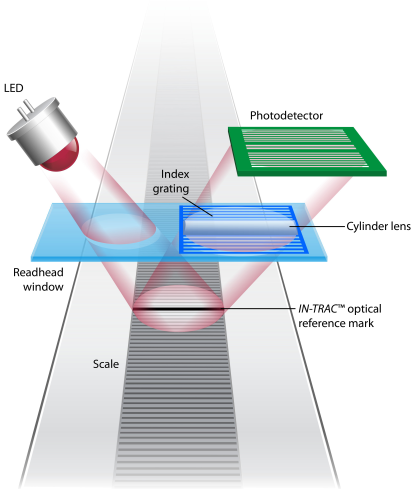


Figure 3.2.12: Renishaw tape encoder and readhead configuration

However, the use of a tape encoder and a read head may generate positioning errors, which may be periodical or not due to two main factors.Warner pointed out this problem in 2008 [58] and proposed a technique to take these variations into account.
Periodical errors are mostly due to a miscalibration of the sine and cosine waves offsets and gains and their frecuency corresponds to the grating spacing of the tape. They can not be corrected by a large lookup table since as it reaches the telescope encoder bandwidthg it depends on the telescope mount rates. As a result he proposes to model and correct for the amplitudes and offests of each encoder signal inside the control loop.
On the other hand the non periodic errors mostly induced by local variations of the distance bewteen the tape and the head, have to be modelized using a lookup table and kalmann filtering.

In the end, tape and read head absolute encoders require very high mechanization precision of the tape support because any variation of the distance between the readhead and the tape will induce variation of the grating period as seend form the readhead. Introducing this way, positioning errors with a important number of harmonics, with a 1-turn period which can be hardly corrected using classical pointing models equations.
Unfortunately their installation on small telescope still represents an important proportion compared to the overall cost of a 1m class telescope and less. As a results it is very unusual to find use of high precision encoding tapes and high resolution on-axis encoders on amateur's class telescopes of 50cm of diameter and less.
However it is possible to find some companies proposing affordables control systems making use of an absolute on axis encoder such as Gemini Telescope Design [59] become affordable to this class of telescopes. The Gothard Astrophysical OBservatory of Eotvos University, Szombatheley in Hungary Carried out a robotization of a 50cm telescope [60] using an integrated mount from Gemini Telescope Design. Even if in this commercial device, the absolute encoder allows to keep track of the mount's position even after failure or poweroff, its resolution is not high enough to have an active control of periodical errors or windburst. These still need to be corrected using a separate guiding telescope and a remote camera head.
In the commercial amateur oriented market, Sidereal Technology TCS[61] also offers the option to instal high resolution relative encoders placed on the main axes of the telescope. [62][63]

3.2.2.4 Direct drives

The Direct drive motor is a brushless motor with no reduction between the main axis of the telescope and the rotor of the engine. Permanents magnets are placed on the rotor while series of coils are installed on the stator part of the motor. Fig.3.2.13 represents a direct drive motor used in a telescope and the respective stator and rotor of the same system are shown in Fig.3.2.14.

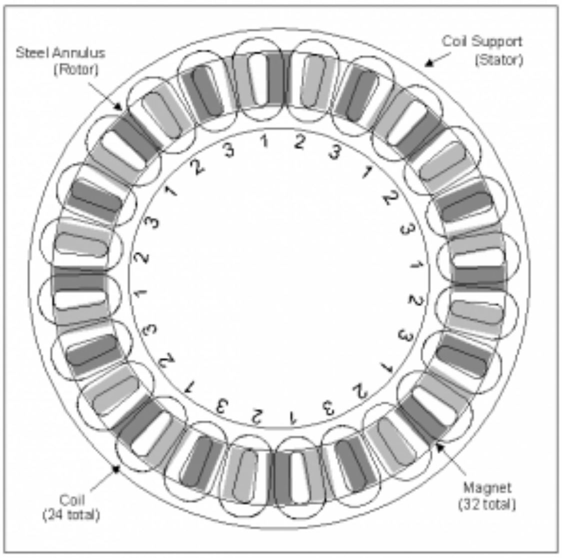


Figure 3.2.13: Coil and magnets organization of a direct drive brushless motor

Every newly designed large scale and very large scale telescopes trend to uses a drives technology based on direct drive motors. It can be seen by reviewing the system definitions of most of the lately constructed large telescopes such as the four 8.2m Very Large Telescope (VLT) [64] of the European Southern Observatory (ESO), or the Gemini 8m[65] telescopes in Hawaii an Chile uses this technology. Additionaly, the Preliminary Design Reviews (PDR) and Final Design Reviews (FDR) of the next telescopes generation such as the Thirty Meters Telescope (TMT)[66], or the next generation of chinese observatories [67] will also make use of this tecnology.

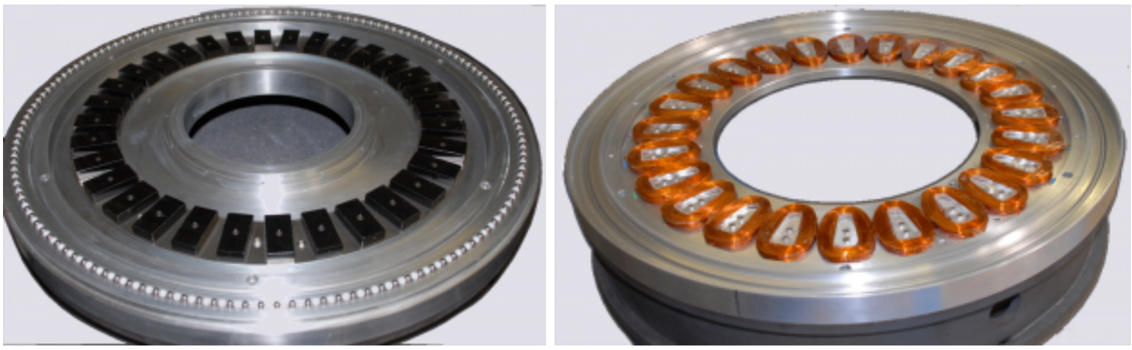


Figure 3.2.14: Stator and rotor of the defined direct drive brushless motor

Direct Drive systems aplications for telescopes were first proposed from the 90's [68] but spread progressively [69] and recently became a standard in the large telescopes aplications[70]. The success on large telescope is mostly due to the following reasons:

- Absence or periodical and non periodical error
- High efficiency due to the absence of reduction stage
- Absence of backlashes
- High range of available speeds from very slow to extremely fast movements (tens of degrees per second) compared to geared systems

However, in order to fulfill the precision specifications inherent to a telescope drive the position control loop of a direct drive motor has to be closed using an encoder with a precision and resolution of at least $1/20^{th}$ of an arcsecond. With this restriction, it implies an important minimum cost regarding the encoder and limits the possibility of application of this technique to small telescopes (50cm-1m range). As a result this technical solution has been used in very few small systems such as in [71] since the cost of the drive system would remain very high compared to the overall cost of such a telescope.

3.2.2.5 Auto-guiding

This last method we mention to compensate and correct for mechanical aberration of the telescopes mechanics consists of adding a second detector either on the focal plane of the telescope or behind a second smaller telescope in parallel of the main instrument.

The goal is to obtain an image as close as possible from the target field where a bright star can be selected and its centroid computed at a periodicity between 1/10s and a few seconds. At every centroid computation its value is compared to the one taken at the begining of the observation and a tracking error is computed. An offset in arcseconds is computed and sent to the telescope mount to correct for the positioning. This method has been extensively detailed in [72] . It has been shown in [73] that the precision can be increased by computing better the centroid of the target star.

Performing this way we can measure and correct for the aberration related to bad mechanics. Even if this method has been widely used in every range of telescope sizes it presents some drawbacks. The main complexity is to find a proper guiding star which is bright enough to have a proper centroid computation at small exposures without saturating the detector which must also be close from the target and can be found into the guiding field. Since the guiding field is usually a few arcminutes wide and the guide telescope is less powerfull thant the main one, a proper guiding star may not be found easily.

Another problem exposed by this method is that since the guiding telescope is installed in parallel of the main ones, the support and proper flexures of the devices are not necesarily the same as the ones of the main telescopes. As a result, during long exposures, the drift recorded in the guider will not be the same as the one in the main instrument and even if the star was properly guided we can observe a differential drift in the science detector.

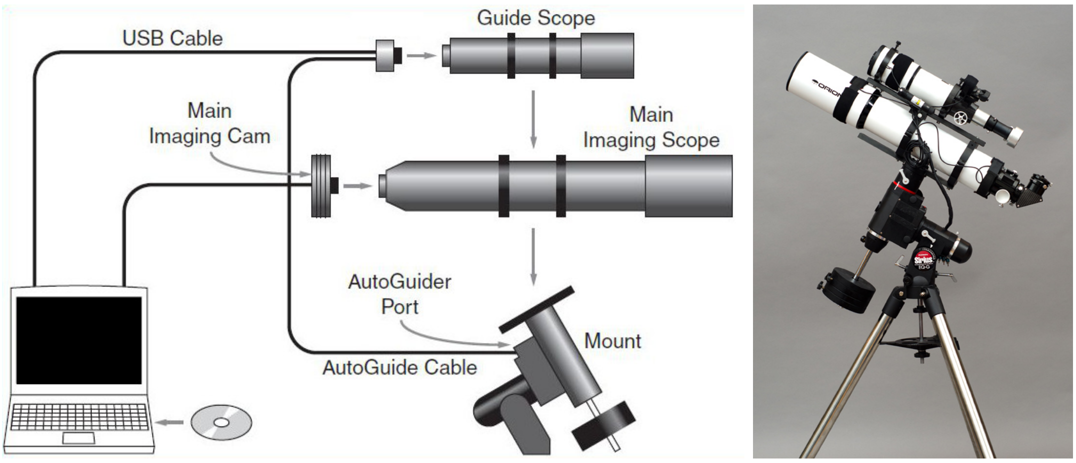


Figure 3.2.15: Autoguiding principle and its possible implementation on a amateur's setup

3.2.3 Generic control systems solutions

Additionally to the hardware, the important cost of the profesional grade Telescope Control Systems is also related to the fact that they have to be most of the time designed from scratch for each aplication. As a result there is very few intempts such as [74] or [75] of standardization of such profesional telescope control systems. The most efficient way to reduce this part of the cost would be to have a generic control system which could be installed either on new devices or upgrade old telescopes.

3.2.3.1 Software Bisque

Commercial solutions of telescope mounts for high-end amateur astronomers and small profesional projects do exist on the market, as an example, the US company Software Bisque have commecialized a Telescope Control System for the mounts produced by the same company [76]. These algorithmics are able to perform observations as a stand alone system. The system responds TCP/IP commands and can be scripted in order to automatize an observing run . It has been widely used in scientific aplications such as exoplanets surveys such as M-EARTH [77, 78] as seen in 3.2.16 or KELT [79]. These projects consist of a set a telescopes pointing series of fields of candidates all over the night during several months in order to detect peridic light variations of somes stars. These periodic variation patterns would correspond to the light decay due to to a partial eclipse of a planet around its host star seen from the Earth every time it passes in front of it.

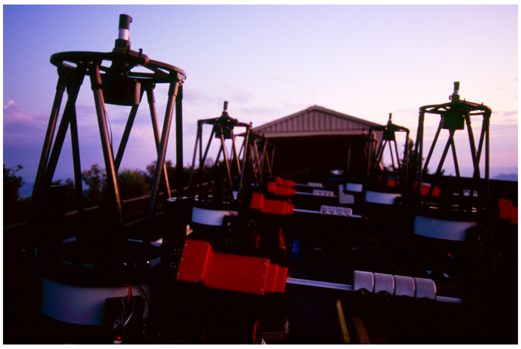


Figure 3.2.16: MEARTH Telescopes on Software Bisque mounts

The same Telescope Control System was also used in the defense area in projects such as the Raven Automated Telescope System used at NASA [80] in order to track and discover space debris

with small telescopes.
Even if this varierty of aplications is a proof of the versatility of the telescope control system it was mostly used in an autosuficient way in aplications where only short exposure times are needed. The main reason why it is not easily adapted to long exposures aplications is because the axes drives are moved by worm gears without high resolution encoders. This configuration produces as defined previusly periodic errors and backlashes when pointing an tracking. Even if these effects can be handled and compensated by using an autoguiding camera, finding and locking a system on guide in robotic operation blindly is not always easy and its aplication ,calibration and configuration is time consuming at both installation and operation time.

3.2.3.2 Telescope upgrades

The size of profesional telescopes have been increasing lineraly in the past century but the cost of a manned operated 1m class telescope is not considerably smaller than 4m class one. As a result, most of the 1m class and less telescope saw their funding cut in the past two decades in order to dedicate budget to bigger and scientifically more challenging projects. However these old telescopes could still perform observing mission for science cases where a bigger collecting surface is not absolutely necessary. This could be achieved for a small operating cost if they could be controlled remotely or robotically. In Chile, most of the telescopes built in the 1960's at Tololo's or La Silla Observatories have been given away in the past decade to universities and cientific project dedicated to upgrades their control electronics and operate them in a unmanned way. As an example we can cite the refurbushing of the ESO50cm telescope from La silla and Tololo 40cm telescope refurbished by Universidad Católica in Santiago, the Bochum 60cm telescope given to Universidad de Valparaiso, the ES01m telescope now operated by Universidad Catolica del Norte. As a consequence, the increase of shut down telescopes with the possibility to be refurbished with low budget brought the need to build a high quality control system for a low cost. In 1984 Melsheimer[81] proposed a control system to control a 0.4m telescope in an automatic way for photometry purpose that could be adapted to other systems. Among the similar following projects, Mossop proposed in 2013[82] to use the new upcoming commercial low cost Arduino boards and Raspberry pi as control electronics. In this project the Telescope Control System was achieved using a commercial solution proposed by Sidereal Technology TCS. Even if this system can handle high resolution encoders, the cost of these remains important compared to the rest of the electronics and the cost of these remains out of budget of most refurbushing projects.

3.2.3.3 Telescope Drive Master

On the amateur astronomer side, a commercial system called Telescope Drive Master was proposed in 2001 on the market [83]. This hardware was initially developped for the HAT-South telescopes and consists of a stand alone electronical devices consising of an 250000 steps per turn incremental encoder class placed on the main Right Ascention axis of the telescope. The microchip inside the electronics counts the pulse per second given by the encoder while the telescope is tracking and is able to measure this way the speed variations of the mechanical system such as perodic errors of the gearing and send corrections via a standard TTL interface.
The TDM commercial system and its encoder are presented Fig.3.2.17. In Fig.3.2.18 we show an experiment of measurement of the tracking error of a star using an amateur mount and a 15cm telescope. The plot represents the tracking error against time. The mount is equipped with a TDM system which is turned on after 650s of measurement. It is obvious on the graph that the tracking error goes down from ±20arcsec peak to peak to 1arcsec approximately. This system demostrated to be low cost and easily could compensate for periodic error corrections during tracking and backlashes during pointing efficiently. However it will always try to compensate in order to keep the telescope main axis tracking at the theoretical sidereal speed (15.04 arcsec/sec). As a consequence it does not allow to track at different speeds which is not only necesary for specific observing runs such as asteroid or comets measurements which do not keep a constant position on the sky during the observation but also when we want to apply a high precision control system taking into acount the pointing model effects as defined in section 3.1.2 in a closed loop control. In this ultimate case the TCS will compensate in realtime for the effects of mechanical assembly of the telescope and athmmospheric refraction effects and thus none of the axes will be tracking

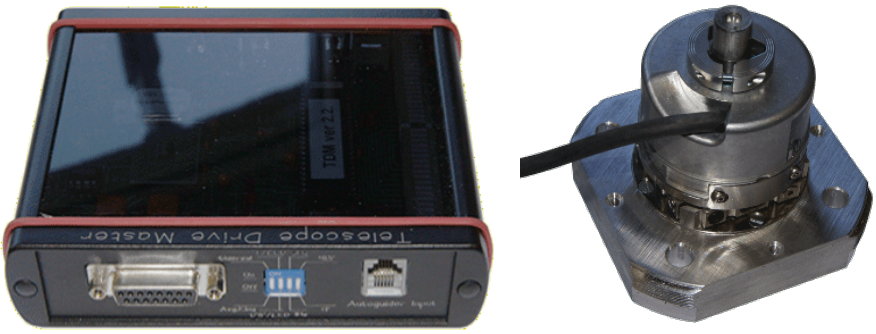


Figure 3.2.17: TDM Electronics and encoder

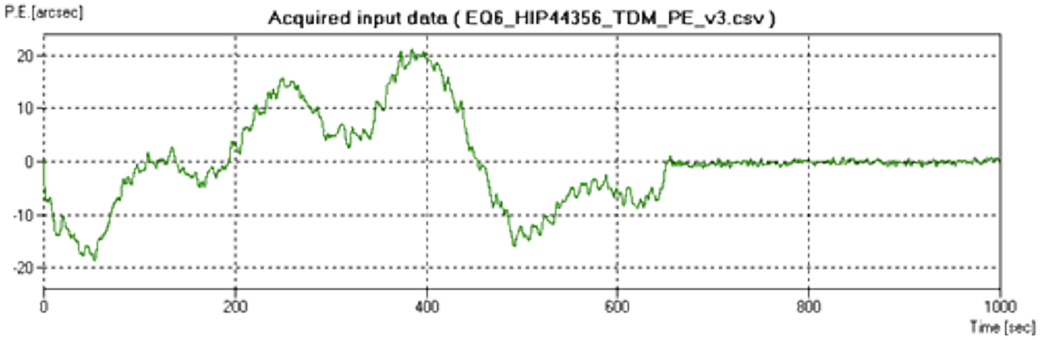


Figure 3.2.18: Tracking error before and activating TDM correction

constantly at the same speed. This need makes this subsystem unusable in this configuration for our application when we want the system to be tracking at very high precision for several hours.

3.2.4 Existing software for remote and robotic observing

With the previously described impulse for using robotic telescopes, several projects using sets of telescopes such as Hungarian Automated Telescope Network (HATNet)[84], Hungarian Automated Telescope South (HATS) [85], Panchromatic Robotic Optical Monitoring and Polarimetry Telescopes (PROMPT)[86], or Las Cumbre Observatory Global Telescopes LCOGT [87] were constructed. The existence of collaborative projects such as Microlensing Follow Up Network (μFUN) [88] increased the need of common software used on every telescope of the newtork from different manufacturing and responding to common command sets. Several intempts have been done of making a homogeneous control protocol for robotic telescopes.

3.2.4.1 Astronomy Common Object Model

The Astronomy Common Object Model (ASCOM) common astronomical platform consists of a digital hub software installed on a computer connecting to a wide number of possible peripheral that can be found in an observatory. This peripherals are grouped by type such as camera, mount , focuser, rotator, weather station etc and the ascom software will include specific driver for most of the periheral vendors and manufacturers. Then Ascom provides a common open source protocol for each kind of periphral that external software can connect to blindly without the need of beeing concern of which specific hardware is connected behind.
Fig.3.2.19 shows a comparation of traditional astronomical software control, plugin based software related to ASCOM arquitecture. In traditional (or Monolithic) arquitecture, the astronomical software has to include hardware specific drivers for every possible harware that could be controlled with it. This solution has became the less optimal since every software susceptible to use astronomical hardware needs to develop its own set of driver for each possible hardware. Additionally, any new hardware would need a new version of the software to be delivered. On the

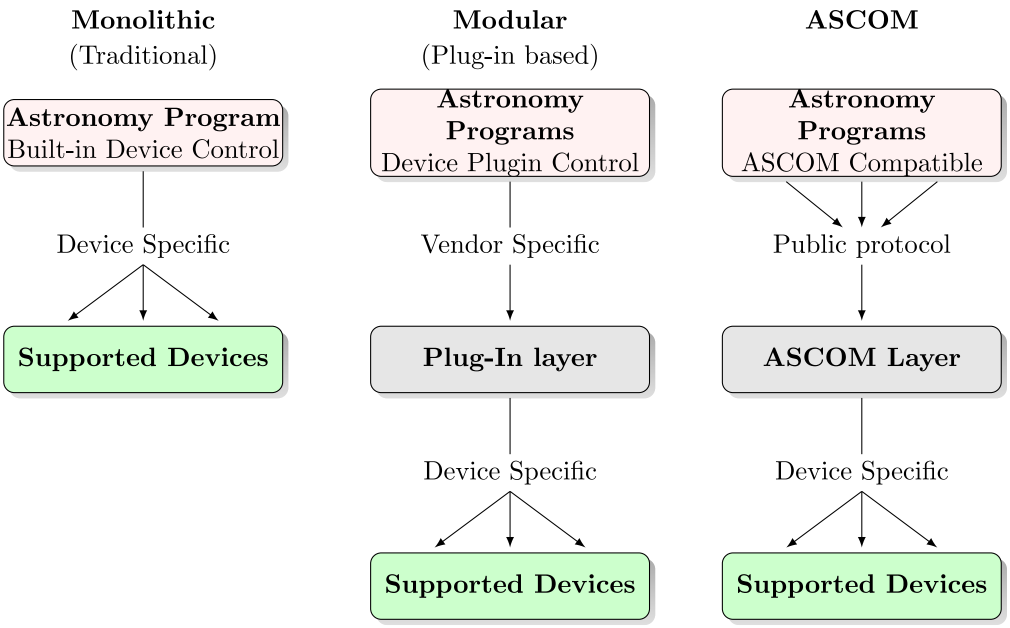


Figure 3.2.19: Comparison of monolithic, modular and ascom arquitecture

other side Plug-In based and Ascom compatible software use a similar structure way. Ascom or Plug-In drivers can be developped by any user as soon as they respect the defined common protocol independantly from the astronomical software using these peripherals. However, the Plug-in arquitecture forces the control software to use a specific programming language while with Ascom protocol, the astronomical software can connect to the hub undependantly of the programming language it used which makes it a more versatile solution. In both cases, the big advantage compared to the monolithic UI arquitecture is that new features integration or new hardware drivers would only need to respectively release a new plugin or new ascom driver independantly of the client program and its programming langage.

The ASCOM protocol is widely used in amateur astronomy, even in remote operation and was also used in some scientific implementations such as the preivously cited Mossop'upgrades [82]. The open source protocol allowed many independant software with an important variety of specialities to connect to it and operate a station remotely. Howerver it presents to major restrictions if we want to operate an observatory in a autonomous configuration in a robust and optimized way:

- ASCOM software only run on Windows, thus every drivers and client software must be run on the same platform
- Every peripheral drivers has to be coded using C# programming langage. But not all periphral manufacturer would give a C# API. Additionnally it does restrict the collaborative programming since not all collaborators would fell comfortable with this
- Every peripheral and client has to be connected to the same computer which can become heavy sowtare consuming in come cases of observatories with an important number of peripherals. Additionaly Windows operating system trends to give erratic and none constant response when several peripherals are connected to the same computer. (Ports number can change after reboot and similar issues that can be handled but do need an important time consuming effort at the moment of putting a new telescope in operation)

3.2.4.2 Instrument Neutral Distributed Interface

The Instrument Neutral Distributed Inteface (INDI) is a software arquitecture similar to the previously defined ASCOM. The general arquitecture presented in Fig.3.2.20 and is composed of three

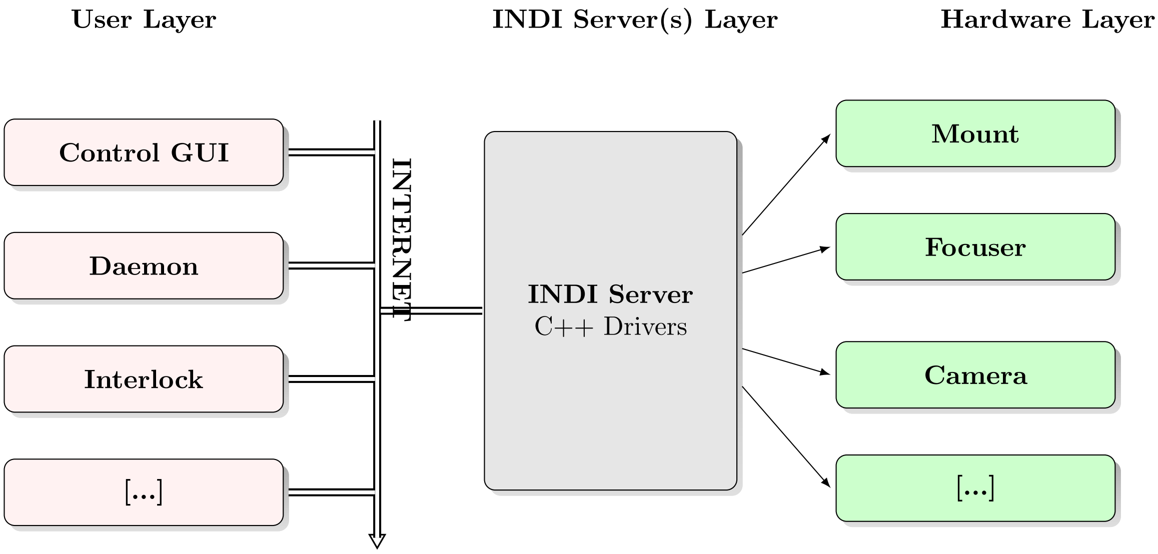


Figure 3.2.20: INDI arquitecture

layers: The lower level is the hardware layer, the drivers programmed in C++ language are included into the servers while the users can interact with the servers from the user layer over Internet. The public protocol of interaction over the TCP/IP network is managed by XML commands.

As we can see INDI offsers three main advantages over ASCOM:

- Command are sent over network, so that the peripheral load can be distributed over several computers which makes the system more reliable
- The server run on Unix/Linux systems which is more stable than windows but since the control is done through XML commands over TCP/IP it remains independant of the user's Operating System (OS)
- New hardware drivers can be programmed by the user using C/C++ standard APIs

Most of the mutliplateform astronomical software with the possiblitiy of connecting to hardware as seen in [89] have linked their interfacing through the INDI protocol arquitecture. And the fact that a hole observatory peripherals can be installed in separate computers or even single board computers such as raspberry pi's, because of its versatility and robustness compared to ASCOM , have allowed many scientific robotic projects such as the 60cm robotic telescope of Montsec Observatory [90][91] to prefer this technical solution.

However, even if INDI platform is multiplatform compatible, which means clients can be run on any platform, the servers needs to run on a Unix/Linux environment. As a consequence, every driver have to be develop in C/C++ only and will run under Linux environment only which also limits the compatible hardware since some peripheral vendors will only provide Windows APIs. Another interesting point is that the fact that drivers can be developped only in C/C++ language also limits the amplitude of collaborative community even if the software is open source. Most of the potential collaborators in terms of drivers programming will prefer other programming languages, as a result , fixing the inteface to a single language limits the quantity of people able to collaborate on the project. Finally we can point out that the fact that INDI server will only run under Linux and ASCOM is Windows only, both system will remain totally exclusive from each other and it is not possible to develop any kind of bridge system that would allow peripherals with an ASCOM driver to be controlled from an INDI client and vice-versa.

3.2.4.3 Remote Telescope System

The Remote Telescope System (RTS) was initially developped in 2000-2001 by Jelinek & Kubanek [92, 93] for the automatization of the BART Telescope, a small aperture telescope designed for quick Gamma Ray Burst (GRB) follow-ups. The system is Linux/Unix based and is widely configurable and extensible. It is fully compatible with any INDI drivers and offers the same flexibility and system restriction as INDI has. RTS was then extended to its second version (RTS2) in 2006[94] improving the versatility and stability to be used as a generic observatory handling platform. While INDI only provides an interface between the hardware and an XML protocol, RTS2 provides a high level autonomous interface which is able to observe by itself without any kind of human intervention. The system can be fed with lists of targets to be observed from web interfaces, it will then analyze the weather conditions by itself and observer autonomously while it finds favorable condition. One interesting feature is that it is even able to detect failing subsystems and decide wether it is necessary to stop observing for safety reasons. An example of an RTS2 driven observatory is presented in Fig.3.2.21. presented in [95] .

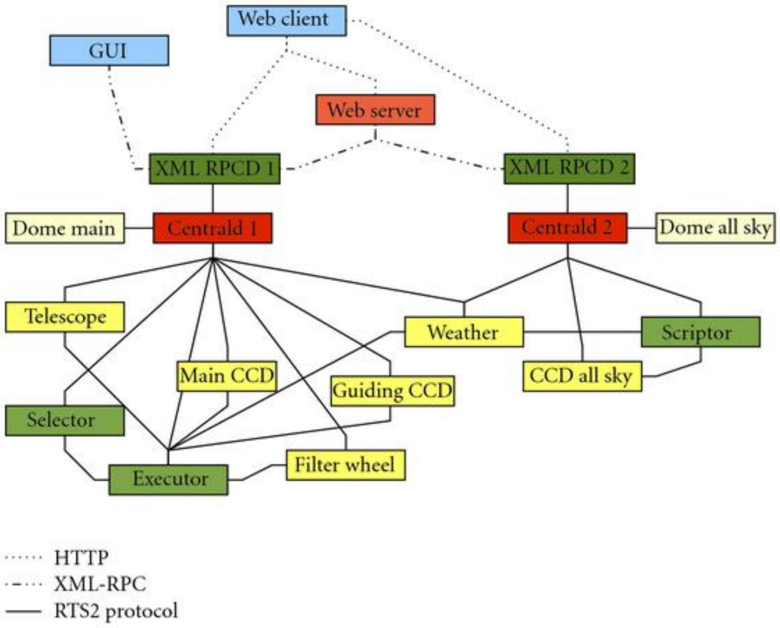


Figure 3.2.21: Example RTS2 environment. Two basic setups are present on the site—a telescope with a guiding CCD and a main CCD equipped with a filter wheel, and an all sky camera. There are two domes (one for telescope, second for all sky camera), three CCD detectors, executor and selector services controlling telescope setup and scriptor, a simplified executor service, controlling the all sky camera. Both observatories offers external access through XML-RPC protocol, provided by XMLRPCD. This is used by the Graphical User Interface and the Web server. XMLRPCD also provides a Web browser with direct access to some functions. Access to XML-RPC and Web functions can be protected with a password.

The RTS2 plateform constantly evolved [96][97] since its creation and its latest features implementation such as scheduling based on either classical dispached[98, 99] or queued scheduling implemented in big observtories such as the Very Large Telescope (VLT)[100] but also provides scheduling facility based on Genetic Algorithms (GA Scheduling) [101, 102, 103]. The versatility modularity and stability of the system makes it the most used integrated software in robotic astronomy projects.

4 Algorithmics

In this chapter we will present the five major contributions of this Thesis related to control of robotic observations. Obtaining data through a telescope can be affected by errors introduced in several levels of the acquisition chain and this has been tackled along the Thesis through different methods.

As for the hardware level subsystems, in the first section of this Chapter we will present the hardware level improvements applied to the telescope drive control. This first contribution consists of a novel approach based on Kalman's filter data fusion coming from multiple rotative encoders placed at strategic positions in the mechanical axis chain. The data from thes rotative encoders feeds a closed loop algorithm which allows to obtain a very high resolution measurement of the telescope speed and an accurate position using affordable hardware for astronomical projects with small budgets.

The software level subsystems will be covered in sections 2 and 3. In Section 2 we will present a software for integrated control which offers the possibility of integration of multiple pointing models in the same telescope, in order to improve the guiding quality. Such an approach is proposed and implemented for the first time, to our knowledge. Currently, most of the commercial telescope control system for small telescope do propose a pointing model software but in most of them it only corrects for the telescope position at the moment of aiming the target at the beginning of the observation. Very few of them have the capability of track the pointing model evolution in a closed control loop as the telescope tracks the target. The technique we propose would be the first to allow small telescopes to have a dedicated pointing and tracking model for a guiding system. Such a simple proposal is able to improve the guiding precision of the overall system, improving the quality of imaging, specially when long exposures are involved.

The third contribution of this Thesis involves a new distributed and portable software architecture for sharing sets of peripherals in an observatory, which will be presented in detail in Section 3. This system is specifically designed to facilitate the collaboration of a developer community in order to easily integrate new compatible hardware, a situation usual in many small-sized amateur observatories. As defined in the State of the Art, the two mostly used software platforms for controlling telescope peripherals are currently ASCOM and INDI. Both solutions implement a modular structure of peripherals allowing collaborative development, and a hardware abstraction layer for the end-user's observatory control softwares. In addition to this, the solution we propose offers two drastic improvements which are the possibility of distributed servers and peripherals over the network, which is not possible with ASCOM, and the multi-plateform compatibility for both servers and clients. The proposed system is designed for being compatible with ASCOM or INDI, which currently are not compatible with each other, so the software may also be used as a bridge between them.

Sections 4 and 5 will present respectively two improvements allowing a better positioning of the optics in order to consistently obtaining a high image quality. While Section 4 describes in details a system for active repositioning of the telescope optics using a simplified, yet effective, mechanical approach, the proposal presented in Section 5 describes an approach based on software analysis of the defocused images near the focal plane. This analysis shows how it is possible to generate a model of the focal plane behavior of the optics in order to be able to position precisely the detector using a system such as the one described in Section 4 from single images shots, using the entropy of the image as the main parameter in a merit function.

4.1 High Precision Drive Control

After defining the drive configuration of our prototype telescope, we will analyze the behavior of its motors and present the Kalmann-based encoder fusion algorithm we propose in order to obtain a higher tracking precision while keeping a cost-effective approach.

The first goal of our design is to obtain a cost-effective drive mechanism, suitable for advanced amateur astronomers, able to track sidereal objects on the sky smoothly, with no visible jumps or periodic errors larger than typical sky turbulence jitter of the stars. As we previously saw in the State of the Art Section, closing the control loop with an absolute high resolution band encoder placed on the main axis of the telescope would easily permit to fulfill this requirement, but its cost is way above the overall budget for mechanics of a typical amateur astronomer telescope. This is why the second goal of our design is to be able to fit the cost of electronics, motors and positioning sensors within the same order of magnitude of a typical amateur astronomer hardware.

4.1.1 System Description

Since we cannot afford a high resolution belt or disk on-axis absolute encoder, we will next discuss how it is possible to obtain similar precision and resolution using a tipical worm/gear driven approach to which we will add a medium resolution encoder of large production scale which remains within the limits of a typical budget. This first Section is intended to describe in detail the system proposed.

Along the following subsection we will first define the hardware components we chose to drive each of the two axes of the telescopes. We then expose the specifications of the encoders we used to measure speed and position of each axis and their positioning into the chain. Then, we will present the electronics used to give power to the motors and close the loop with the previously defined encoders, and then all the components will be embedded together to handle all the previouly detailed hardware and sensors. Finally, we will expose which are the requirements this system must fulfill in order to be able to track smootly a star on sky so that the hardware would now affect the image quality in any sense.

4.1.1.1 Reduction and motors

In the proposed test setup, the telescope is built with two axes and each axis is driven in an equivalent way, as defined in Fig.4.1.1:

- A horizontal axis is used for azimuth pointing and tracking using a simgle motor
- A vertical axis is used for elevation pointing and tracking using a single motor

Both motors are DC Motors model Pittman 16 AMP 14204 24V. their specifications are shown in Tab.4.1.1. Electrical performance of the motor is expressed in Fig.4.1.2. Each axis is driven by a worm/gear assembly with 359 teeth which induce a reduction factor of 359X between the worm axis and the main axis. In addiiton ot this reduction the worm is driven by a 7X reductor from APEX Dynamics. The specification of the reductor can be seen in Table 4.1.2. The reductor is driven itself by a belt drive with an additional 3X reduction factor.

As a result, the total reduction factor between the main axis and the motor shaft may be calculated as:

$$\omega_{\text{motor}} = \frac{\omega_{\text{axis}}}{(\text{Number of teeth}).(\text{Reductor factor}).(\text{Pulley factor})} \quad (4.1.1)$$

$$\omega_{\text{motor}} = \frac{\omega_{\text{axis}}}{7539} \quad (4.1.2)$$

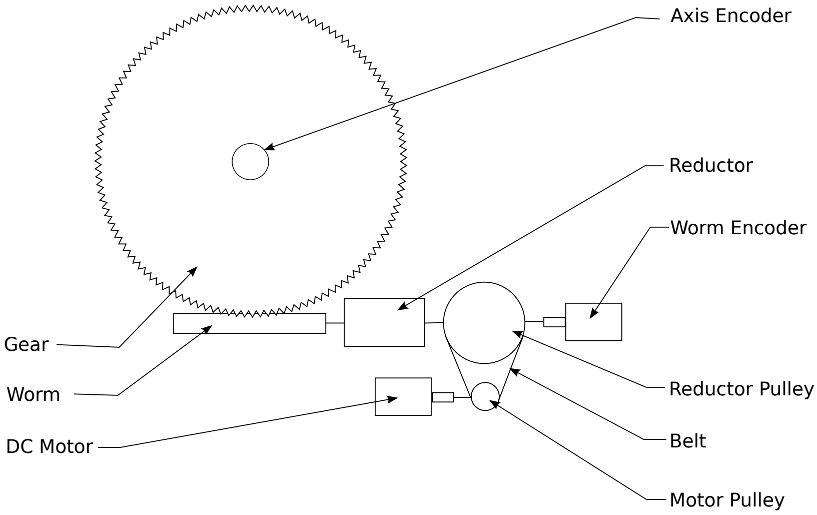


Figure 4.1.1: Mechanical setup of each axis

Model Number	Pittman 1420424V	No-Load Speed	3630 RPM
Supply Voltage	24 V	Peak Current	23.8 A
Continuous Torque	0.1836 N.m	Peak Torque	1.4402 N.m
Speed at cont. Torque	3200 RPM	Coulomb Friction Torque	0.0113 N.m
Current at cont. Torque	3.67 A	Viscous Damper Factor	$1.21x10^{-5}$ N.m s/rad
Continuous Output Power	63 W	Electric Time Constant	1.6 ms
Motor Constaant	0.061 N/$\sqrt{W}$	Mechanical Time Constant	7.0 ms
Torque Constant	0.0612 N.m/A	Thermal Time Constant	29 min
Voltage Constant	6.41 V / kRPM	Thermal Resistance	7.7°C/W
Terminal Resistance	1.01 Ohms	Max Winding Temperature	155°C
Inductance	1.60 mH	Rotor Inertia	$2.61x10^{-5}$ kg-m2
No-Load Current	0.26 A	Weight	997 g

Table 4.1.1: Specifications of the Pittmann Direct Current motors used

Model Reference	PE050	Max. Radial Load	820 N
Number of stages	1	Max. Axial Load	410 N
Ratio	7	Service Life	20,000 hr
Nominal Output Torque	12 Nm	Efficiency	$\geq$ 97%
Emergency Stop Torque	36Nm	Weight	0.8 Kg
Nominal Input Speed	4500 RPM	Operating Temp	$-10°C \rightarrow +90°C$
Max. Input Speed	8,000 RPM	Lubrication	CASTROL LMX Grease
Backlash	$\leq$ 8 arcmin	Gearbox Protection	IP64
Torsional Rigidity	2 Nm/arcmin	GearBox Inertia	0.15N.cm2

Table 4.1.2: Specifications of the APEX dynamics reductor used

4.1.1.2 Encoders

Each axis is coded in position and speed by two independent encoders of different resolutions. While the first encoder, of higher resolution, is positioned on the main axis, the second one mea-

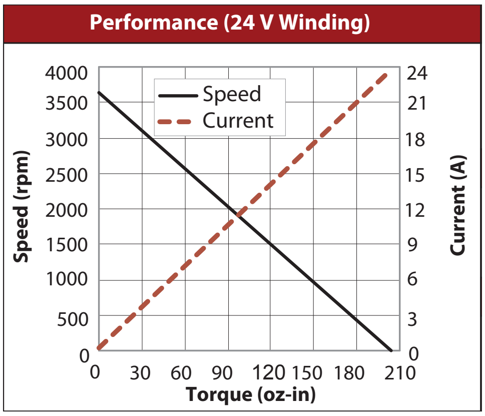


Figure 4.1.2: Motor Performance

sures the angle and speed of the worm. Their respective configuration can be detailed as follows:

- The axis encoder (with higher resolution) is a quadrature incremental Gurley RS158S encoder. Its specifications are detailed in Table 4.1.3. When considering the inner interpolation of the sinusoidal signals, this encoder delivers quadrature signals of 250000 steps per revolution. With the proper reading interpolation, the final resolution of this encoder can reach 1.000.000 steps per revolution. The encoder is installed directly on the main axis as shown on fig4.1.1. The total resolution on the main axis in arcseconds per step may be calculated as:

$$\mathrm{ResAxis}_{\mathrm{Arcseconds}} = \frac{360 * 3600}{(\mathrm{ResAxis}_{\mathrm{StepsPerTurn}})} \tag{4.1.3}$$

$$\mathrm{ResAxis}_{\mathrm{Arcseconds}} = 1.296 \text{ Arcsec/step} \tag{4.1.4}$$

- The worm encoder is a quadrature incremental ELCIS 58-10000, whose specifications are detailed in table 4.1.4. This encoder gives a direct output of 10.000 periods per revolution. With the proper reading interpolation the final resolution can reach 40.000 counts per revolution. This encoder is installed on the input axis of the 7x reductor. The mean total resolution in arcseconds per step on the sky will then be:

$$\mathrm{ResWorm}_{\mathrm{Arcseconds}} = \frac{360 * 3600}{359 * 7 * 40000} \tag{4.1.5}$$

$$\mathrm{ResWorm}_{\mathrm{Arcseconds}} = 0.013 \text{ Arcsec/step} \tag{4.1.6}$$

However, although the resolution of the second encoder is effectively smaller as the absolute accuracy of the system will be limited by the errors of the reductor and of the worm/gear assembly. As a result, the axis encoder will give a low resolution with high accuracy while the worm encoder will give a high resolution with low accuracy.

Maximum line count on disc	5000
Maximum cycles/rev (quad sq waves)	250,000
Max counts/rev (after quad decode)	1,000,000
Internal square wave interpolation	50X
Encoder error	±15 arcsec
Maximum output frequency	2500kHz
Starting torque	< 0.01 Nm
Moment of inertia	1.710^{-6} kg.m^2
Maximum acceleration	310^6rad/s^2
Operating temperature	0°C → 70°C
Humidity	non-condensing 98%
Shock	< 300 m/s^2, 10 ms
Vibration	< 100 m/s^2, 55-2000 Hz
Maximum weight	0.25 kg
Sealing	IP64
Bearings	Grease-lubricated and sealed
Maximum radial shaft load	20N
Maximum axial shaft load	10N

Table 4.1.3: Specifications of the axis encoder

Base	Anodized Aluminium
Cover	Valox420
Maximum cycles/rev (quad sq waves)	10,000
Max counts/rev (after quad decode)	40,000
Max RPM	8000
Max Consumption	120mA
Operating temperature	0°C → 70°C
Storage temperature	-30°C → 85°C
Maximum output frequency	100kHz
Starting torque	18g.cm
Shock	50G, 11 ms
Maximum weight	0.33 kg
Sealing	IP64
Bearings life	1.510^9 revolutions
Maximum radial shaft load	30 N
Maximum axial shaft load	30N

Table 4.1.4: Specifications of the worm encoder

4.1.1.3 Drivers

The connexion layout of the drivers in order to control speed in position is detailed in Fig.4.1.3. For each axis we use a driver controller IPECMOT 48/10. This driver handles the connexion of one DC (or brushless, if brushless motors were preferred for the setup) motor, its corresponding limit switches and the axis encoder. The connexion to the IPECMOT is done using a TCP socket from the versalogic single board computer . The IPECMOT can handle position reading and TCP connection every 0.5ms and encoder frequencies of up to 400kHz. The local position counter on the board is incremented every 0.5ms and saved to the **Electrically Erasable Programmable Read-Only Memory (EEPROM)** memory as soon as the supply voltage goes down to 7V. As a result the absolute position is always recorded, even when the system is shut down. This is useful whenever the system is power cycled or is started after being off for a while, it will remember its actual position whenever it was not moved after being parked. Since in most of the cases, the system is powered off every day and is not moved while it is not used, storing the last known position allows the system to imediately know its position even if it does not use absolute encoders. This avoids the need of finding a home position for each axis at the begining of operation which can be a tedious operation.
The reading of the worm encoder is done using a dedicated PIC 18F26K22 microcontroller connected to the single board computer serial port. Reading can be done every 5ms and the device can handle encoder signal frequencies up to 1MHz.

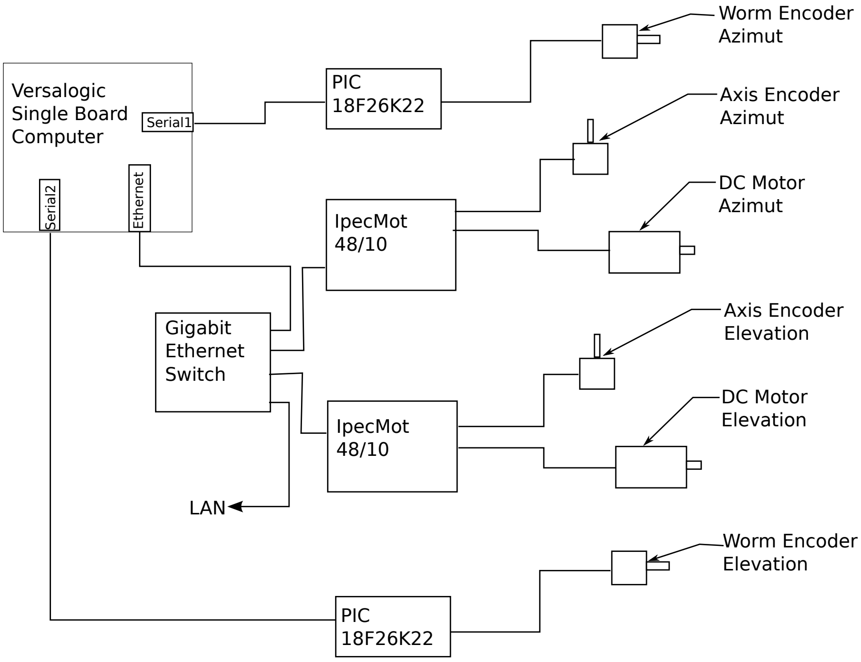


Figure 4.1.3: General layout of the drivers for motors and encoders

4.1.1.4 Specifications

The precision in motor positioning and tracking is dependent essentially on one main parameter, which is the sampling on the CCD. Sampling was designed to oversample stars by a factor oftwo under normal seeing conditions, in order to satisfy the Shannon-Nyquist sampling theory. As a result we chose a pixel sampling of 0.5 arcsec/pixel on sky.
However, for photometric applications, which are one of the key applications enabled by small telescopes, the telescope must be able to track a star so that the mean position over an exposure stays within a tenth of the pixel. This permits to avoid photometrical perturbations due to different pixel sensitivity within the same pixel.

Regarding its speed limit, telescopes depend on the position in the sky and the slewing speed. For the first item we express the tracking speed in Az and El as a function of Az and El in the drawing figure 4.1.4. Speeds are computed for a site at 44° North latitude. The drawing expresses speeds on elevation and azimut axes as a function of azimuth and zenithal distance. We can see that in terms of tracking the tracking az speed should be infinite at the zenith, but if we limit the observation to $+85^o$ from the horizon tracking can go from 0 arcsec/sec to 200 arcsec/sec, which become reasonable values. This is obviously at the prize of a blind tracking cone of only 5 degrees in the zenit.

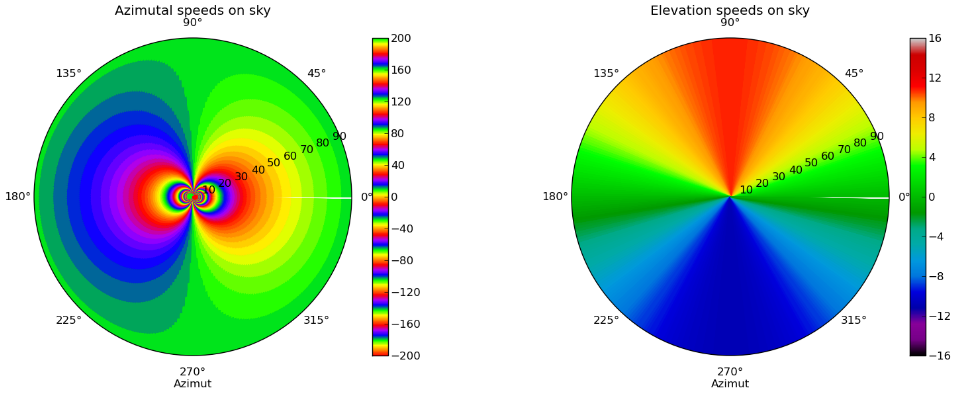


Figure 4.1.4: Elevation and azimuthal speeds on sky at 44° North latitude

Then, in order to take advantage of the best seeing conditions, the tracked star must stay within 1 arcsec RMS of its nominal position. If such condition is attained, the error in the tracking speed over a measurement sampling period does not depend on the setpoint of the tracking speed. In order to keep the actual speed of the motor as close to the target desired speed, the motor is controlled with a **Proportional–Integral–Derivative controller (PID)**. This algoritm consists of measuring the Target and actual speed at a fixed timing period, and at each of this measurement samples, the target instantaneous input voltage of the motor will be a combination of three values:

- The actual speed error multiplied by a proportional fixed gain K_p
- The cumulated sum of the error and previous error measurements mutiplied by a integral fixed gain K_i
- The cumulated difference of the error and previous error measurements mutiplied by a derivative fixed gain K_d

The acceptable error then only depends on the **EEPROM** sampling time and can be expressed in equation 4.1.7. Two typical examples of caculated acceptable error at two **EEPROM** frequencies are presented in equations 4.1.8 and 4.1.9.

$$\text{AcceptableError} = \frac{\text{SpeedError}}{\text{PIDFreq}} \tag{4.1.7}$$

$$\text{AcceptableError}_{|\text{PIDFreq}=10Hz} = 5\text{Arcsec/sec/sample} \tag{4.1.8}$$

$$\text{AcceptableError}_{|\text{PIDFreq}=20Hz} = 10\text{Arcsec/sec/sample} \tag{4.1.9}$$

4.1.2 Performance in standard configuration

4.1.2.1 Ipec device under open loop control

For the control of the full system an industrial DC motor servo controller IpecMot 48/10 device was selected. Its characteristics are listed in 4.1.1.3. It provides an embedded **Pulse-width modulation (PWM)** control running either at 48kHz or 76Khz. The pulse width ratio can be

set using either a TCP Socket or the internal web interface of the device. For our application we will use the TCP socket since it can be directly controlled by our python scripts with a console. Additionnaly, the TCP socket is compatible with the ICE network distributed architecture of the complete control system to be presented in the proper moment.
Despite of the important reduction ratio between the main axis and the motor, the minimum speed requirements is so low that the motor needs be arranged to turn at sub-rpm speeds, which cannot be obtained in standard designs and thus will be developed on purpose.
On the other hand, the maximum speed requirements for slewing do not permit to have a larger reduction ratio, as otherwise the motor would work out of its maximum speed limits. Table 4.1.5 shows the necessary RPM on the motor axis depending on the rotation speed in the main axis.
In a first test, we determined experientally which was the minimum acceptable **PWM** ratio before the motor stalled at low speed. In the graph 4.1.5 we express the measured speed on the main axis as a function of the **PWM** value. The **PWM** value is expressed as normalized on a 16bits signed scale, so 32767 means 100% clockwise , -32767 means 100% counter clockwise, while e.g. 16536 means 50% .
For each **PWM** value measured, we recorded the speed on the telescope main axis during 20s. The graph 4.1.5 represents the mean speed measured on these 20s as a function of the PWM value. Error bars represent the standard deviation of the speed during the related measurement.

Axis Speed	Motor Speed
0.5 Arcsec/s	1°/s = 0.165RPM
2.55°/s	3200RPM

Table 4.1.5: Motor speed required at minimum and maximum tracking speeds

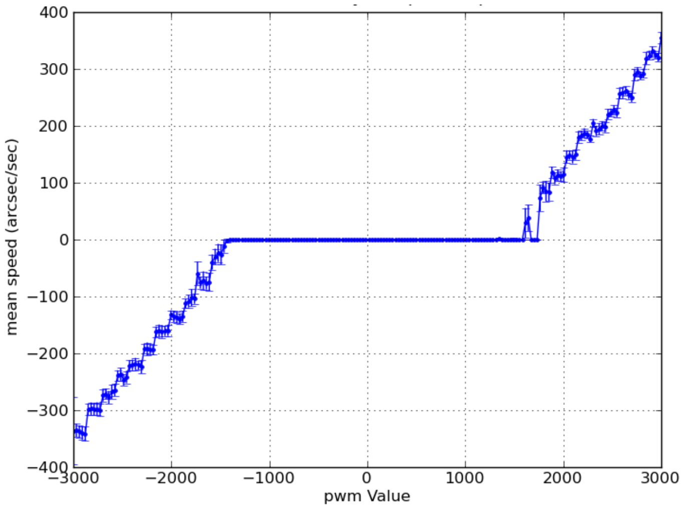


Figure 4.1.5: Motor linearity in open loop mode

In the graph we can observe that the motor stalls at a **PWM** value of 1800 units, which is about 5% of its full speed, both in clockwise and counterclockwise directions. The minimum speed before the motor stalls is around 100 arcsec/s on the main axis, which is not enough to fit our requirements. This stalling value is essentially due to the 2ms electrical time constant of the motor, which filters the **PWM** pulses when they become shorter than 2ms.
We can also observe that the speeds are increasing by steps of 50arcsec/s. This effect is due to the digitalization inside the **PWM** hardware, and as a consequence introducing a fast closed loop control at the lowest possible level is absolutely necessary if we want to be able to smoothly move the motor in the complete range of speeds from 0.1 arcsec/sec to 10000arcsec/sec.

4.1.2.2 Ipec device under closed loop control

Beyond the **PWM** values, the Ipec device also provides a 32bit PID embedded control loop running at 2KHz. At every sample of the device it measures speed by counting the number of encoder pulses occurred in each direction along the last sampling interval, and compares it to the target speed. As with a standard PID controller, the **PWM** output of the device during the next sampling interval will be the a weighted combination of the speed error, the accumulated speed error and the derivative speed error.

In Fig. 4.1.6 we plot the mean speed measured over 20s against the target speed sent to the Ipec device . Every 20s measurement is obtained by computing the mean of 20 measurements over 1s. Error bars represent the standard deviation of the speed for individual measurement according to the 20s measurement.
We can see that the obtained mean for each measurement speed is not only accurate over the complete desired range, but also how the standard deviation is also stable along the complete measurement. Thus, the implementation of the closed-loop working mode at the Ipec is mandatory, and it has been shown to enable our specified range of speed.

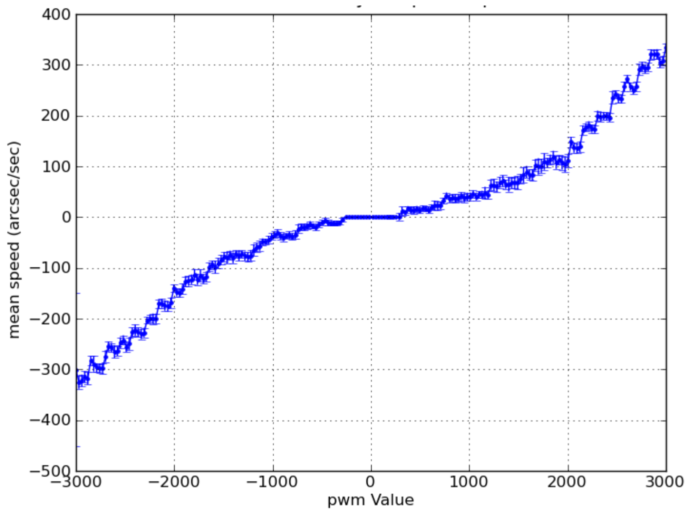


Figure 4.1.6: Motor Linearity in Closed loop mode

4.1.3 Data fusion from the two encoders

As we presented in Fig. 4.1.1 of section 4.1.1, we are actually using two encoders per axis, with the axis encoder will be giving a high accuracy (≤ 1.3Arcsec) with a low resolution. The worm encoder will be giving a better resolution which can be useful for the control(≥ 0.05 Acrsec/Pulse), but with a much lower accuracy ($\geq \pm 15$Arcsec) because it is driven by the worm/gear and reduction errors.
Before feeding the control loop algorithm, however, it is necessary to fuse the data of both encoders in order to compute a position which would have both high accuracy and high precision.
Along this section we will first consider the simulated data for both encoders. The simulated data will be generated according to the experimentally observed signal from our test telescope. This model will be used in order to tune aproximately fusion algorithm parameters and have a better initial set of parameters when it will be tested in real conditions. The same filters will then be tested on real data to validate their performance. In order to obtain simulated datasets as close as possible to the real data we first take into account the following facts and assumptions:

- The Axis encoder data does not suffer propagation or mechanical error since it is directly installed on axis and its step changes every 1.296 Arcsecond (we consider 5.000 sinusoidal

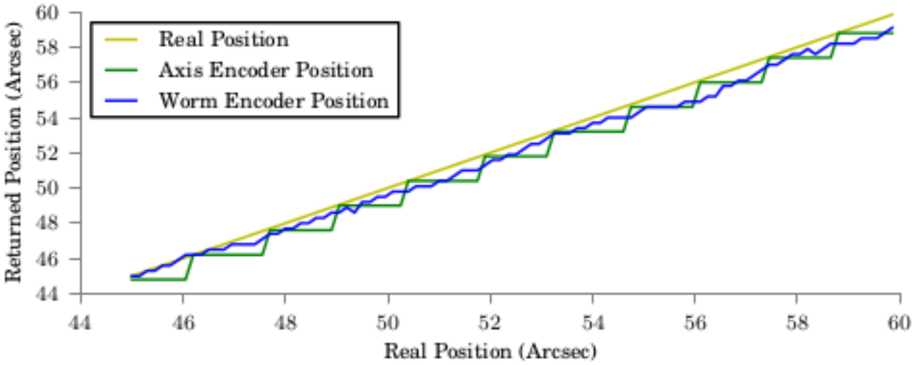


Figure 4.1.7: Encoder response as a function of the real position

periods per revolution in the encoder, with a 50X internal interpolation and 4X quadrature interpolation and no reduction stage)

- The worm encoder has a resolution of 0.18 Arcsecond per step (we consider a 5.000 sinusoidal periods per revolution in the encoder, with no internal interpolation and 4X quadrature interpolation and one 360X worm/gear reduction stage)

- The worm gear affects the position of the worm encoder with a period of 1/360 turn of the main axis corresponding to one turn of the worm. The amplitude of this sine wave periodic aberration is set to ±19.1Arcseconds

- Local aberration of the worm gear and general transmission are modeled as brownian noise. Each step is affected by a random normal error between -0.2 and $+0.2$ Arcseconds which gets propagated to the next step.

- We also include two mechanical eigen frequencies affecting the worm encoder position as experimentally measured from the same encoder. These two eigen frequencies have respective periods of 3.0 and 8.2 Arcseconds and amplitudes of 0.2 Arcseconds.

- Robustness of the algorithm can be tested against a step of 4.0 Arcsecond affecting only the axis encoder, meaning the maximum step jump acceptable to the algorithms will be that value

- Robustness of the algorithm can also be tested against a simulation of a wind burst, consisting in a exponentially amortiguated oscilation with an amplitude of 10.0 Arcsecond and a 500 Arcseconds pseudo-period

Fig.4.1.7 shows a close-up view of the response of both encoders as a function of the real position of the mount on the sky over a movement of 30 arcseconds. By comparing the relative positions given by the worm encoder and the axis encoder with the real position, we can clearly see the effect of the resolution of both encoders and the high frequency aberration introduced by the worm encoder.

In order to see the effect of the main worm periodic aberration, we have to refer to Fig.4.1.8. This plot shows the positioning error of the worm encoder and of the axis encoder over a movement of 9000 arcseconds. In this plot we can clearly see the large periodic error due to the worm effect. The relatively lower resolution of the axis encoder is expressed as a thicker zero mean uncertainty around the exact position.

4.1.3.1 Data fusion by direct interpolation

The easiest way to combine the data of both encoders in a single output according to the previously defined behavior of the worm and axis encoder is to proceed as follows:

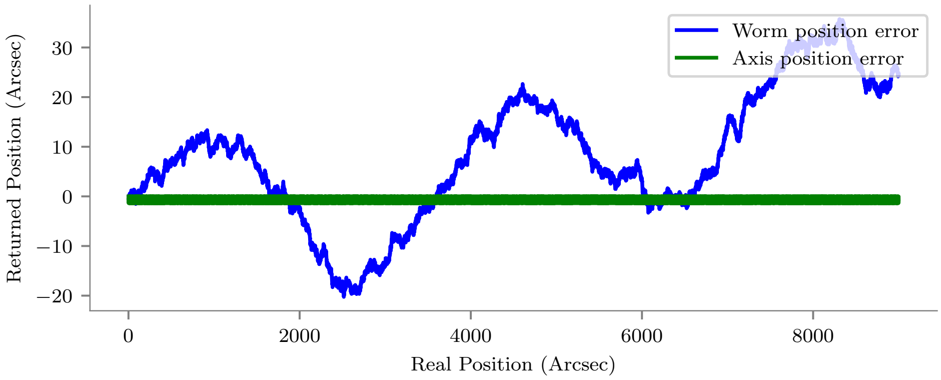


Figure 4.1.8: Encoder response as a function of the real position

- The fused position and the position of the encoders is initialized with the latest known fused position
- At each sampling time we measure the position in arcseconds given by both encoders
- If the position of the axis encoder has changed since the last measurement, the fused position is set to the axis position
- Othrewise, in case the axis encoder position has not change since the last measurement, the fused position is set to the last known axis position plus a correction corresponding to the movement done by the worm encoder since the last change in position of the axis encoder.

This simple algorithm has shown to be extremely robust, since it completely filters out most of the long period aberration of the worm gear, and significantly increases the final resolution of the system compared to the value given by the data from the axis encoder. However, this approach will not reduce the high frequency noises included in the worm encoder due to its resolution, or aberrations included by the gearing system with periods in the order of magnitude of the resolution of the axis encoder. The effect of the direct interpolation method can be seen in Fig.4.1.9 where the result can be compared to the axis encoder and worm encoder values.

4.1.3.2 IIR filtering of the axis encoder data

In order to reduce the resulting noise previously mentioned, due to an interpolation error of the worm encoder, resulting in a periodic variabe offset in the position measurement, we propose to filter the position data from the worm encoder before its fusion, using an Infinite Impulse Response filter (IIR) designed in the frequency domain. For simulation purposes, the filter is defined as a combination of a low-pass filter and two second order notch filters.

The low pass frequency is chosen to equal half of the worm encoder resolution, while the notch filters are tuned to be synchronized with the frequencies observed on the power spectrum of the positioning error of the worm encoder.

The effect of the filtered interpolation position method can be seen in Fig.4.1.9, where the result can in addition be compared in the time domain to the axis encoder values, worm encoder values and standard direct interpolation values. Since the effect of the removal of the aberration frequencies is not clearly apreciable on Fig.4.1.9, we plot the same data in the in the frequency space in Fig.4.1.10 where the filtered data can be compared to the non filtered values. It may be seen how the two frequencies visible on the original worm data are completely removed. However even if this filtering method turns to solve the initial issue in simulation when applied on a system where the motor moves at a constant stable speed, a test in real conditions show that it may take tens of seconds to stabilize itself when the motors arrives to this speed or when a mechanical perturbation like a windgust affects the theroretical movement of the telesocpe.

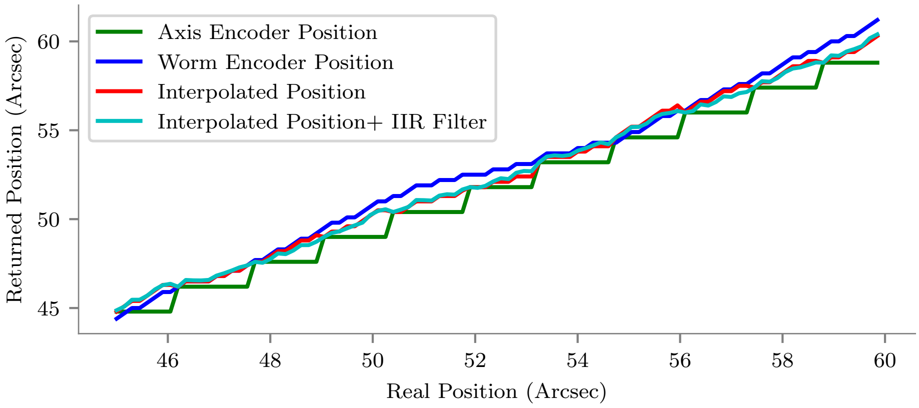


Figure 4.1.9: Response of direct fusion interpolation and effect of IIR Filtering on the worm data

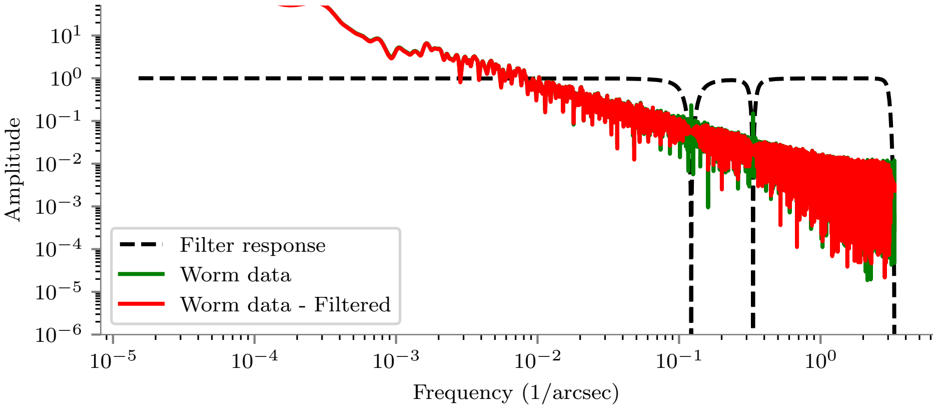


Figure 4.1.10: Direct fusion response and effect of IIR Filtering on the worm data

4.1.3.3 Sensor fusion based on an extended Kalman Filter

When the previous data fusion algoithmics were tested on sky, the analysis of the precision of the traking enabled to observe an oscillation of the tracked star around its theoretical position with a period of 280 arcseconds, corresponding exactly to 200 steps of the axis encoder, which is also corresponding to one complete sinusoidal phase of this encoder before interpolation. As a conclusion, the encoder model which had been used showed a periodical error of 3.5 arcseconds of amplitude every 200 interpolated steps, which makes it insufficient for its use as a true reference position.

One possible solution to take this effect into account is to add an aditional IIR filter on the data from the axis encoder. However, its low frequency introduces a delay in the measurement of the position which cannot be calibrated, and which will deteriorate the true position measurement obtained. Another potential solution to filter out this effect would be to use a lookup table of offsets for each of the 200 steps of the sinusoid, but this becomes technically impossible since the phase of the sinusoidal signal of this encoder may be lost when the system is shut down.

Such an effect has been known since a few years on tape encoders, as we can see in [104], and also in the case of analog encoders [105]. In both cases, the measurement errors induced were compensated using an Extended Kalman filter as defined in [106][107]. We propose to implement a similar technique in order to correct for the periodic interpolation error of the quadrature encoder

we are using. As a result, the best solution is to evaluate dynamically the phase and amplitude of the periodic encoder error according to the data from both encoders.
The goal of a Kalman Filter is to estimate the real state of a system by statically evaluating which part of the measured signal is noise an which part is the real state. In practice we use a priori-information of the system state in order to predict a measured vector according to the previous state and evaluate the noise of the obtained measurement according to the deviation of the measurement compared the predicted one.
The output estimated state is in fact a ponderation of the estimated measurement and the mesured one. The ponderation is expressed in a matrix called the Kalman Gain. In other words, more the predicted measurement is close to the measurement itself , more the estimated state will take into account the measurement and vice-versa.
In our situation, we measure the position from two encoders which we know the following information:

- The worm encoder has a high resoultion but is affected by the periodical error and noise of the worm/gear system which is hardly predictable.
- The axis encoder has a lower resolution and is not affected byt the worm/gear noise and periodical errors. Howerver it is affected by an interpolation error of 200 steps of period and roughly ±3arcseconds of amplitude which phase is unknown.

Our goal is to have a proper estimation of the real position of the axes by melting and removing the respective noises of these two measurements by using a-priori knowledge of the system such as the speed and amplitude of the axis encoder interpolation error. The phase of this error is then constantly estimated and reevalueated as a state parameter of the system.
This said, we propose to fuse the data of both encoders using a non linear, or Extended Kalman Filter (EKF) defined as follows. Let's assume the state vector $\mathbf{x}_k$ at the sampling time k of the system is defined by the following four states, as depicted in equation 4.1.15:

- p_a is the position of the axis given by the axis encoder
- $\dot{p}_w$ is the speed of the axis given by the first derivative of the worm encoder
- a is the current amplitude of the periodic error of the axis encoder
- ϕ is the current phase of the priodic error of the axis encoder

In this case, the periodic error e of the axis encoder may be expressed as a single harmonic sinusoidal function of period ω equal to 200 interpolated steps of the axis encoder, as we can see in equation 4.1.10. For improving the linearity of the system we prefer to use the linearized version of the function as expressed in equation 4.1.11 to 4.1.14

$$
\begin{aligned}
e &= a\cos(\omega p_a + \phi) && (4.1.10)\\
e &= a(\cos(\omega p_a)\cos(\phi) - \sin(\omega p_a)\sin(\phi)) && (4.1.11)\\
a1 &= a\cos(\phi) && (4.1.12)\\
a2 &= -a\sin(\phi) && (4.1.13)\\
e &= a_1\cos(\omega p_a) + a_2\sin(\omega p_a) && (4.1.14)
\end{aligned}
$$

$$
\mathbf{x}_k = \begin{bmatrix} p_a \\ \dot{p}_w \\ a_1 \\ a_2 \end{bmatrix} \tag{4.1.15}
$$

The state transition equation model that allows to pass from the state at sampling time k to the state at sampling time $k+1$ can then be expressed as in equation 4.1.16.

$$
\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \tag{4.1.16}
$$

Where $\mathbf{u}_k$ is the input form the actuator input and $\mathbf{w}_k$ is the estimated noise introduced in the transition. In our case, $\mathbf{f}$ is linear and can be expressed as in equation 4.1.17.

$$\mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) \quad = \quad \begin{bmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k \tag{4.1.17}$$

While the standard Kalman Filter uses a linear state transition matrix, the Extended version allows to use non linear function for the state transition. The matrix used for the state transition is then the Jacobian matrix, or matrix of partial derivatives constructed from the non linear transition functions.
Since the state transition funcion $\mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k)$ is time invariant, its Jacobian is equal to the state transition matrix as seen in equation 4.1.18.

$$\mathbf{J_F}_k \quad = \quad \begin{bmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1.18}$$

The state measurement function $\mathbf{h}_k(\mathbf{x}_k)$gives the relationship between the obtained measurement vector $\mathbf{z}_k$ and the actual state of the system $\mathbf{x}_k$ as expressed in equation 4.1.19. In our case, the measurement vector is composed of two values, given by our two sensors, which are the position m_1 of the axis encoder and the speed given by first derivative v of the worm encoder. Then we can compute the state measurement function, in which we take into account the nonlinearity of the axis encoder according to equation 4.1.20.

$$\mathbf{z}_k \quad = \quad \mathbf{h}_k(\mathbf{x}_k) \tag{4.1.19}$$

$$\mathbf{h}_k(\mathbf{x}_k) \quad = \quad \begin{bmatrix} m_1 \\ v \end{bmatrix}^T = \begin{bmatrix} p_{a\,k} + a_{1\,k} * \cos(\omega * p_{a\,k}) + a_{2\,k} * \sin(\omega p_{a\,k}) \\ \dot{p}w_k \end{bmatrix}^T \tag{4.1.20}$$

The Jacobian $\mathbf{J_H}_k$ of the measurement function $\mathbf{h}_k(\mathbf{x}_k)$ can then be expressed as shown in equations 4.1.21 and 4.1.22.

$$\mathbf{J_H}_k \quad = \quad \begin{bmatrix} \frac{\partial m_1}{\partial p_a} & \frac{\partial m_1}{\partial \dot{p}_w} & \frac{\partial m_1}{\partial a_1} & \frac{\partial m_1}{\partial a_2} \\ \frac{\partial v}{\partial p_a} & \frac{\partial v}{\partial \dot{p}_w} & \frac{\partial v}{\partial a_1} & \frac{\partial v}{\partial a_1} \end{bmatrix} \tag{4.1.21}$$

$$\mathbf{J_H}_k \quad = \quad \begin{bmatrix} 1 - \omega a_{1\,k} \sin(\omega p_{a\,k}) + \omega a_{2\,k} \cos(\omega p_{a\,k}) & 0 & \cos(\omega p_{a\,k}) & \sin(\omega p_{a\,k}) \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.1.22}$$

Once the measurement function is known, the Kalman filter needs an apriori knowledge of three covariances matrixes.

The process covariance matrix This matrix express how the target state can evolve. Since each state parameter is independant from the others, it is initialized empirically as a diagonal matrix. The trace of this matrix is obtained by setting initial guess of the squared standard deviation of movement of each state parameter.

The process noise covariance matrix This matrix express how the state can deviate from its target affected by movement noise. Since each state parameter is independant from the others, it is initialized empirically as a diagonal matrix. The trace of this matrix is obtained by setting initial guess of the squared standard deviation of the white noise of each state parameter.

The measurement noise covariance matrix This matrix express how the measurement is affected by noise evolve. Since each measured parameter is independant from the others, it is initialized empirically as a diagonal matrix. The trace of this matrix is obtained by setting initial guess of the squard tandard deviation of the noise of each sensor.

let's consider as initialization parameters the initial process covariance matrix $\boldsymbol{P}_{0|0}$, the process noise covariance matrix $\boldsymbol{Q}$ and the measurement noise covariance matrix $\boldsymbol{R}$ as diagonal matrices. The amplitude of the diagonal values is initialized as the squared value of the estimated process standard deviation and noise standard deviation. This leads to the matrices express in equations 4.1.23, 4.1.24 and 4.1.25.

$$\boldsymbol{P}_{0|0} = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \tag{4.1.23}$$

$$\boldsymbol{Q} = \begin{bmatrix} 1e^{-5} & 0 & 0 & 0 \\ 0 & 1e^{-5} & 0 & 0 \\ 0 & 0 & 1e^{-5} & 0 \\ 0 & 0 & 0 & 1e^{-5} \end{bmatrix} \tag{4.1.24}$$

$$\boldsymbol{R} = \begin{bmatrix} 0.01 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix} \tag{4.1.25}$$

Once the model is properly defined we can now apply it at each step of measurement in order to obtain the estimated prediction of the next state of the system $\hat{\boldsymbol{x}}_{k|k-1}$ together with the estimated prediction of the covariance matrix $\boldsymbol{P}_{k|k-1}$. The prediction equations for the next state are expressed in equations 4.1.26 and 4.1.27.

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_k) \tag{4.1.26}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{J}_{\boldsymbol{F}k-1} \boldsymbol{P}_{k-1|k-1} \boldsymbol{J}_{\boldsymbol{F}k-1}^{\top} + \boldsymbol{Q}_k \tag{4.1.27}$$

The filtered estimated state is then computed at each step as a sum of the predicted $\hat{\boldsymbol{x}}_{k|k-1}$ state and a influence of the latest measurement innovation $\tilde{\boldsymbol{y}}_k$ expressed in equation 4.1.28. As we can see in equation 4.1.29, the latest measurement innovation is ponderated by a gain matrix $\boldsymbol{K}_k$ recomputed at each step according to the last measurement, and updated according to the new measurements. The estimation of the position is then obtained by solving the Riccati equation.

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_{k|k-1}) \tag{4.1.28}$$

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \tilde{\boldsymbol{y}}_k \tag{4.1.29}$$

The gain matrix is in fact an expression of the reliability of the latest measurement itself. In simple words, the closest is the measurement from its predicted value, the bigger will be its relative influence in the estimation. The Kalman Gain $\boldsymbol{K}_k$ joint with the estimated covariance matrix are obtained by solving the Riccati equation which solution is expressed in equations 4.1.30 to 4.1.32.

$$\boldsymbol{S}_k = \boldsymbol{J}_{\boldsymbol{H}k} \boldsymbol{P}_{k|k-1} \boldsymbol{J}_{\boldsymbol{H}k}^{\top} + \boldsymbol{R}_k \tag{4.1.30}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{J}_{\boldsymbol{H}k}^{\top} \boldsymbol{S}_k^{-1} \tag{4.1.31}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{J}_{\boldsymbol{H}k}) \boldsymbol{P}_{k|k-1} \tag{4.1.32}$$

The previously defined filter is then applied to simulated data representing the output of the axis encoder when the worm is moved at a constant speed. The simulations take into account the periodic aberration of the worm gear coupling with a period of 7200 arcsecond, which corresponds to a system with 180 teeth similar to our available setup, and with an amplitude of 15 arcseconds which has been experimentally measured on the worm/gear couple.
The aberration of the axis encoder is also taken into account with a period of 259.2 arcseconds and an amplitude of 3.5 arcseconds. Finally we also added to the system a brownian noise of 0.2 arcsecond RMS amplitude, which simulates the local friction aberrration of the gearing and mechanics of the setup. The simulated result is shown in Fig.4.1.11 where we plot the difference between the position in the axis encoder and the position of the worm encoder when the worm moves at constant speed. The telescope encoder position is displayed following the value given by

the axis encoder in green, and the value obtained after Kalman filtering is displayed in black. We can clearly see the effect of the filter which removes sucessfully the aberration due to the encoder periodical error without affecting other kind of fast variation which are clearly identificable in the filterered data.

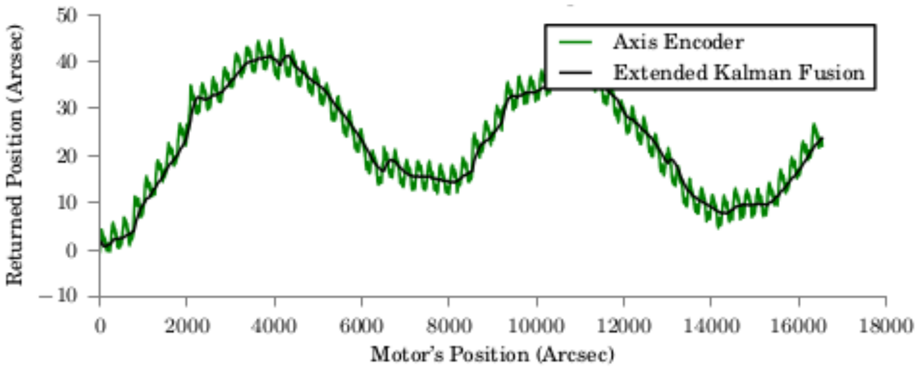


Figure 4.1.11: Response of the dat fusion using an extended Kalman filter

Testing of the proposed data fusion algorithm based on Kalman filtering under real conditions was carried out using a comercial amateur's mount Orion AzEq-G and its standard tracking controller. A TTL quadrature encoder Gurley R158S with 50x interpolation was adapted and installed on axis of the mount using a 3D printed part designed ad-hoc to replace the polar scope as can be seein in Fig.4.1.12.

The encoder output consits of two lines giving a square signal of 250000 period per turn with a $\pi/4$ phase shift. After quadrature interpolation the described setup gave a resolution of 1.296 arcseconds per step. The signal is affected with a periodic interpolation error of ±3.5arcseconds in amplitude and 200 steps period certainly due to an interpolation done by the encoder internally. The Kalman filtering will then help us in removing this interpolation error in order to obtain a precise measurement of theal position of the telecope axis.

The previously defined Kalman filtering algorithm was coded and embedded in an Arduino board in a script reading the encoder position with a time sampling of 1ms. A Python script running on a RaspberryPi computer performed a readout of the position at 10Hz, while the mount was tracking at a theoretical sidereal constant speed of 15.04 arcsec/s for 180s which is the correponding time of one complete turn of the worm for this setup. The results obtained are presented in 4.1.13.

It may be appreciated that even if the encoder aberration is not exactly sinusoidal, a second order harmonic remains present in the processed signal. However, most of the effects have been removed by the Kalman filter whilst keeping a fast response to high speed positioning variations. The approach is thus considered adequate. A similar measurement over a longer period corresponding to a 6.5 turns of the worm gear is shown in Fig.4.1.14, which also shows a measurement of the mechanical periodical error of the mount.

As a result, we have shown Kalman filtering provides a precise method to estimate the exact position of the mount, using the best advantage of the high precision on axis encoder which is not affected by mechanical aberration, and of the higher resolution of the worm encoder. Once we can measure properly the position of the axis, the next step would be to close the control loop in order to keep the system moving whithin the specifications previously exposed.

4.1.4 Closed loop control

A closed loopcontrol is mandatory in this type of instrumentation, as any system only using open loop speed will necesarily end up in a cumulated drift of the target in the field of vue as the time passes. Additionally the closed loop allows to constantly check on the distance between the target and the real position and remove any kind of static positioning errors which often appears in comercial telescope control systems.

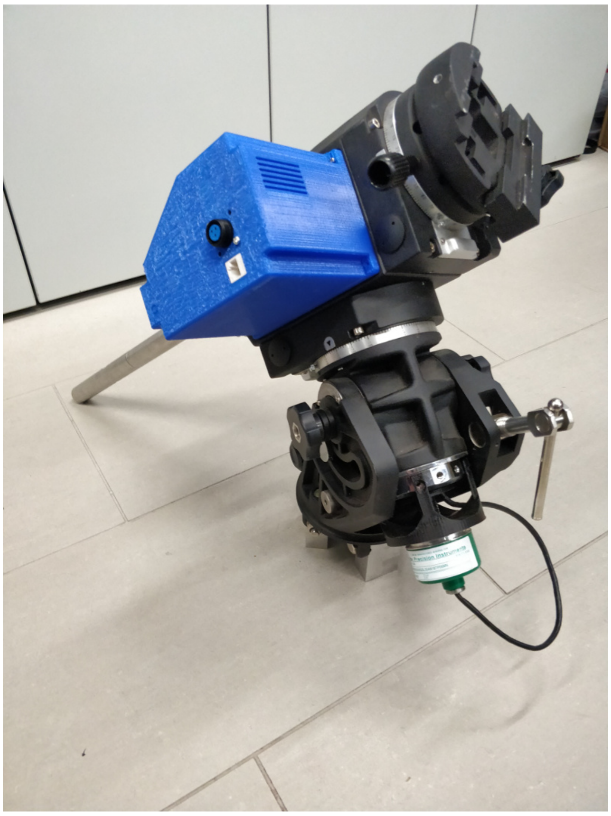


Figure 4.1.12: Picture of the setup and the encoder installation on the main axis of the commercial mount

Our first attempt to implement a closed loop control in the tracking system was by adding to the open loop measurement setup desbribed in the last Section a relay output controlling the guide port intput of the commercial mount as depicted in Fig.4.1.15. In this setup, the position of the axis of the mount is read by a Gurley encoder placed on the right-ascention axis of the mount using an Arduino board in an infinite loop. The corresponding positions are then retrieved using a Raspberry Pi computer equipped with a Linux Debian operating system and Python scripts reading the data from the Arduino over the USB.

The guiding port strategy implemented in commercial mounts is standardized to the following widely used protocol usually referenced as ST-4 port:

- When conductivity is established between pin 2 and 3 of the mount guide port, the integrated mount controller increases the tracking speed on the right ascention axis by 1.5 arcsec/sec.
- When conductivity is established between pin 2 and 6 of the mount guide port, the integrated mount controller decreases the tracking speed on the right ascention axis by 1.5 arcsec/sec.

A standard ST-4 port and its pinout is shown if Fig.4.1.16. Conductivity between the pins can be done by using relays for instance.

As a consequence, the implemented control loop on this test setup is a simple proportional position control loop in the Arduino C++ code with a time sampling $T_s = 10$ms whose flow chart is represented in Fig.4.1.17. This is a very simple control strategy since it only consists of relay activation/deactivation when we detect that the mount is delayed or in advance of the theoretical position. The speed control being handled by the integrated commercial controller of the mount does not need to be taken into account at the Arduino level in this case.

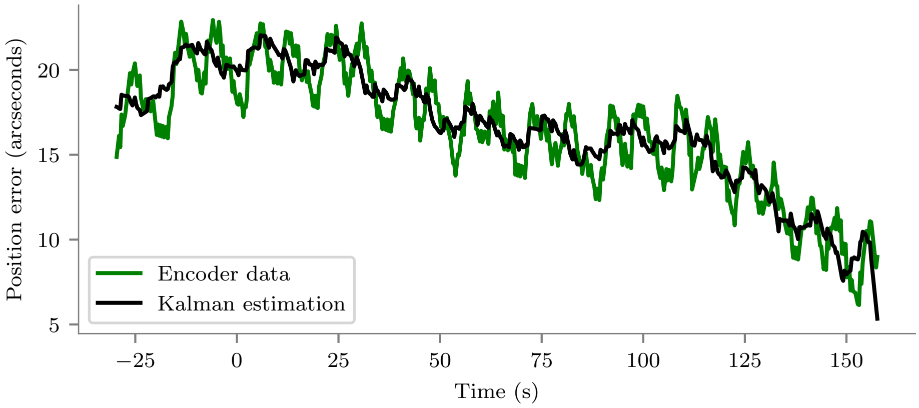


Figure 4.1.13: Response of the data fusion on real data using an extended Kalman Filter along a complete turn of the worm encoder

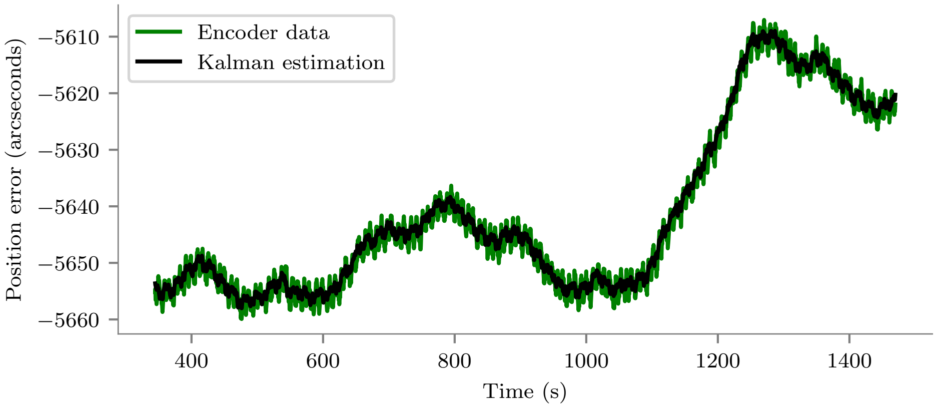


Figure 4.1.14: Response of the data fusion on real data using an extended Kalman Filter along 6.5 turns of the worm encoder.

After implementing the described setup on the commercial prototype presented previously, we recorded the behavior of the telescope position using the Raspberry Pi computer while the telescope was tracking. The closed loop position control was activated after 1150s of recording. We can see in the results in Fig.4.1.18 that at this precise moment, the variable drift completeley disappears and the mount starts tracking its target within ±0.3 arcseconds. This value can be compared as being two orders of magnitude below the previous ±30arcseconds positioning error due to mechanical aberrations in the gearing of the mount.

In order to test the robustness of the system we left it activated and generated a perturbation by mechanically displacing abruptly the telescope tube by 5mm, and releasing it after 1s. This action is stronger than any external actual perturbation expected on the mount, such as e.g. a windburst which may happen under real operation conditions (we are omitting strong earthquakes, which won't be treated). The obtained result is shown in Fig4.1.19. In this figure, the mechanical impulse on the mount is realized at the instant $t = 270s$ of the record is clearly visible. We also observe that even under such a level of perturbation (a pulse inducing an error of almost 20 arcsec) the system does not loose stability after the perturbation and recovers the tracking accuracy within less than 5s, wich is acceptable considering the strength of the artificially induced perturbation applied.

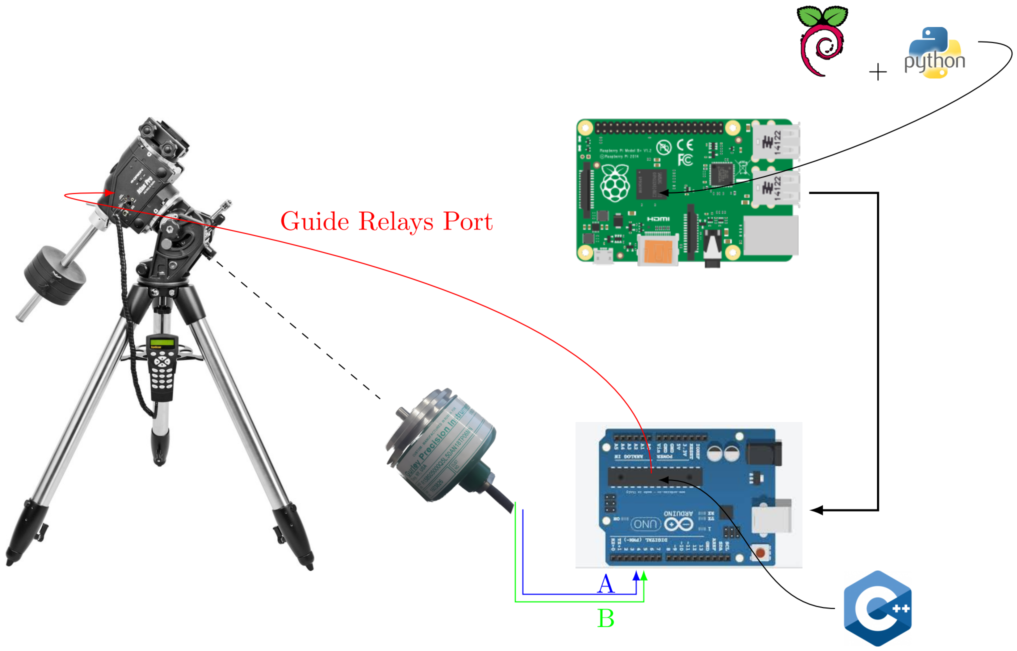


Figure 4.1.15: Schema of closed loop test setup

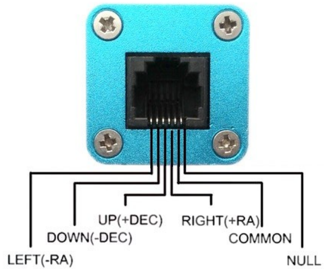


Figure 4.1.16: ST-4 port standard connexion and its description.

In conclusion, the simple and very economic setup proposed completely fulfills the requirements required for astronomical object tracking when adapted to a commercial setup. Further, it could be packaged as a standalone precision tracking system. On the other hand, others conventional commercial telescope control systems have a lot of aditional precision restrictions. These restrictions are not clearly evident but are usually consequences of the comunication protocol being used and its software implementation:

- The commonly used standard LX200 returns or accepts truncated values with a precision of only 1 arcsecond in declination and 1s of right ascention, corresponding to as much as 15 arcseconds.
- These systems do not accept non-sidereal speed inputs and so cannot track using real time pointing data models in order to reduce the tracking drift due to non perpendicularities or flexure effects.
- Internally, these systems cannot generate pointing models with more than 3 to 6 stars, which is indeed another strong limitation since it is not enough to take into account extended parameters. Typically the first 6 parameters represent the polar alignment of the mount and

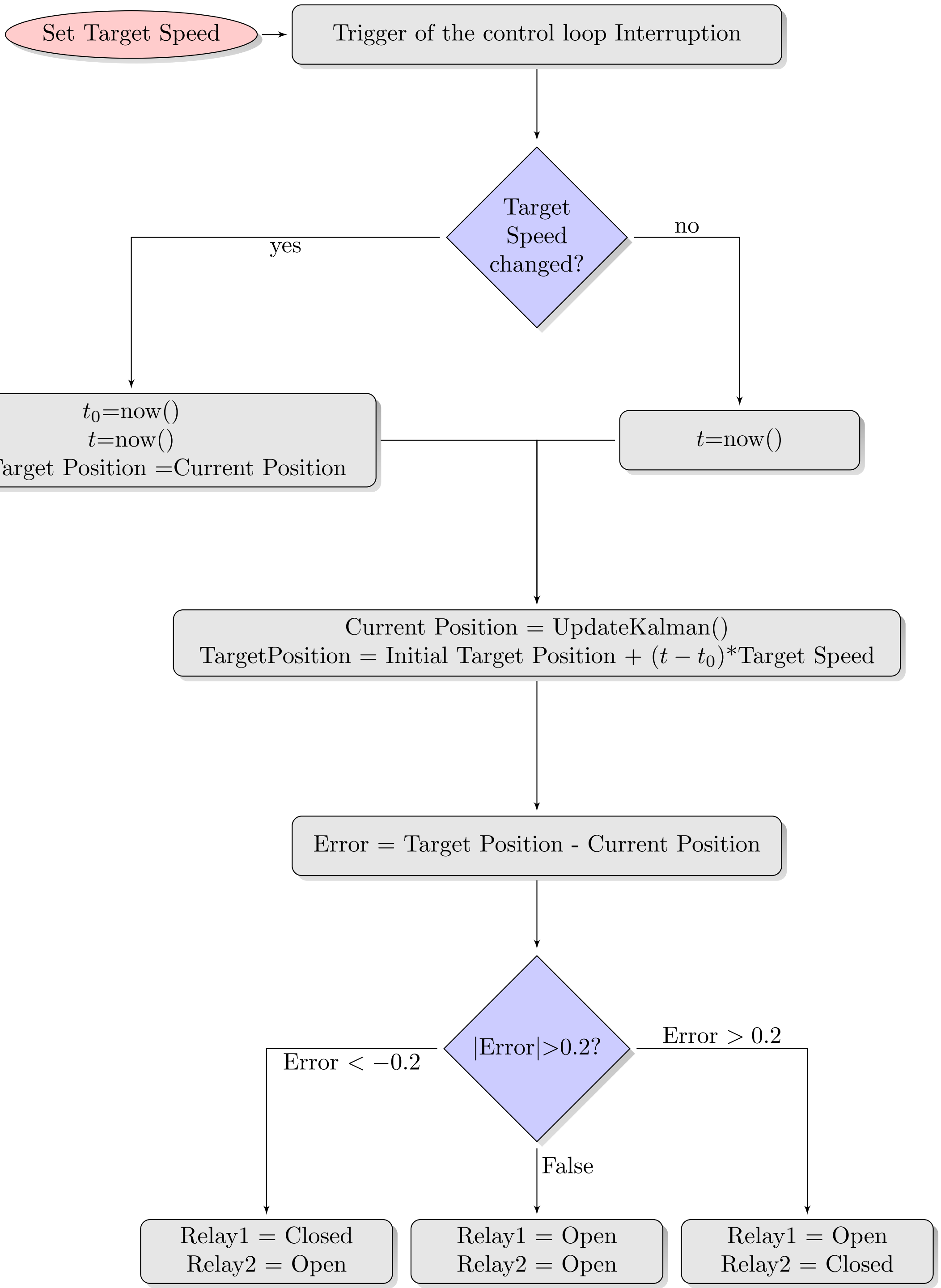

Set Target Speed
Trigger of the control loop Interruption
Target Speed changed?
yes
no
t_0=now()
t=now()
Initial Target Position =Current Position
t=now()
Current Position = UpdateKalman()
TargetPosition = Initial Target Position + $(t - t_0)$*Target Speed
Error = Target Position - Current Position
|Error|>0.2?
Error < -0.2
Error > 0.2
False
Relay1 = Closed
Relay2 = Open
Relay1 = Open
Relay2 = Open
Relay1 = Open
Relay2 = Closed

Figure 4.1.17: Flow chart of the proportional position control loop proposed.

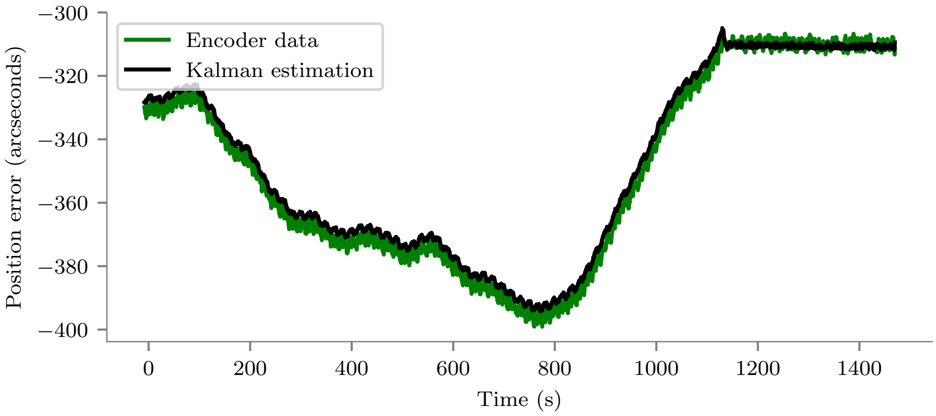


Figure 4.1.18: Data fusion response on real data using an extended Kalman Filter and a closed loop position control. The loop is closed after 1150s of data recording, and its effects easily visible on the error residuals.

the non perpendicularities but extended parameters such as tube or fork flexures do have an important effect on the pointing residuals.

As a consequence, the setup finally implemented in our work slightly differs from the one we applied on a commrecial mount. The motor control is handled as described previously by a TCP industrial controller taking care of an internal PID control loop of the DC motor and its encoder. The position of the motor encoder and speed of the motor can be read by the TCP protocol from a Raspberry Pi while the position of the axis encoder is still obtained via an Arduino board. The output sent to the TCP motor controller corresponds directly to the computed error of the control loop multiplied by a proportional control K_p value.

The main difference between this setup and a commerical one is that on a commercial mount the tracking speed is applied blindly in an open loop and raises or lowers by 1.5arcsec/sec while one of the guide input is activated. In our case the system directly measures the positioning error and applies the correction speed inside the control loop which increases precision and reactivity.

In this configuration, the Kalman filtering is implemented inside the Raspberry Pi, but since the parameters and speeds of the control loop are the same, the results and performances are equivalent. The main difference is driven by the fact that this last configuration, presented in Fig.4.1.20 does not have any major restriction of precision in terms of protocol and speed when it gets integrated in the complete Telescope Control Software.

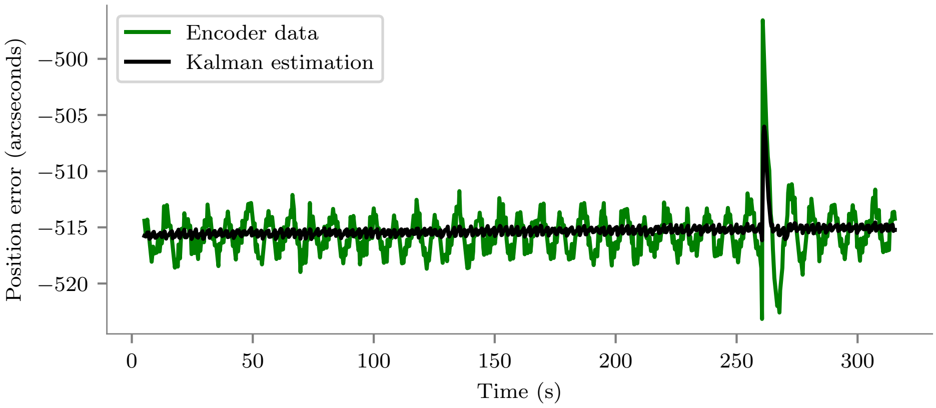


Figure 4.1.19: Data fusion response on real data using an extended Kalman Filter. This plot shows the effect of a manual effort on the telescope tube while tracking, to simulate the action of external perturbations such as windburst. The effort is generated at the instant 270s along the recording and we can see that the system recovers itself in less than 5s.

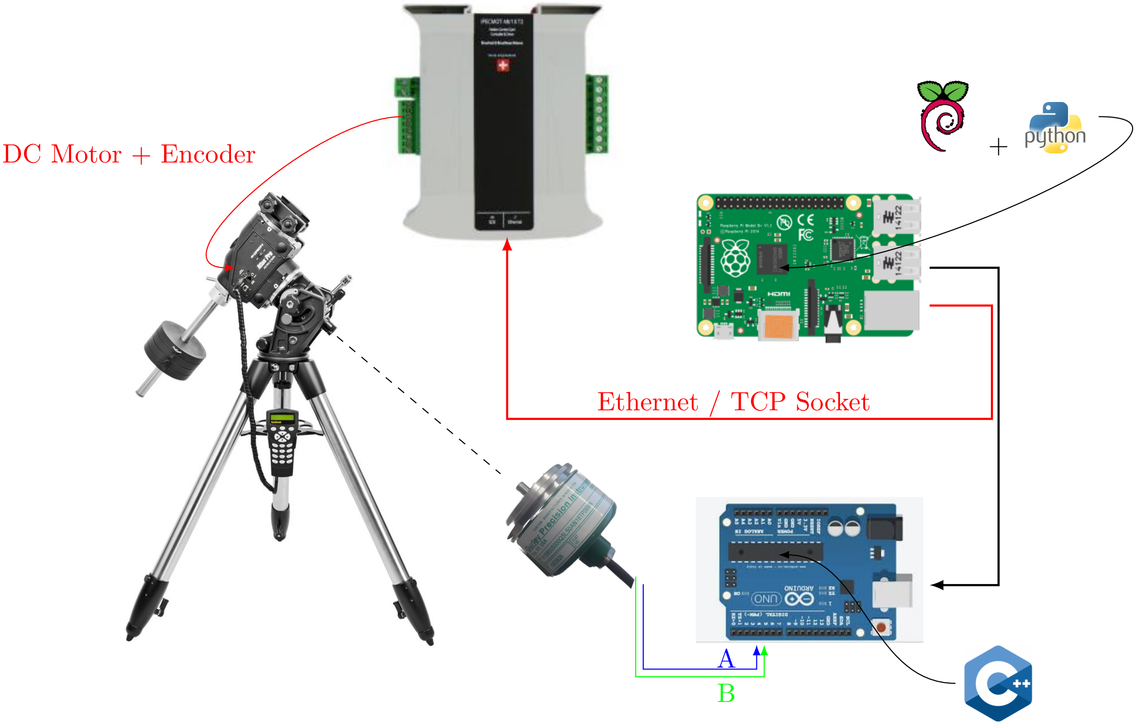


Figure 4.1.20: Schema of closed loop final setup

4.2 Advanced Telescope Control System

Accurate pointing and tracking of telescopes is an important aspect of their operations. When using an external guider telescope, there will be in general differences in the flexures experimented by the different optical assemblies that will cause unwanted positional drifts in the images produced by the main instrument. In this work we propose the use of a pair of pointing models in order to explicitly take into account the different flexures and imperfections of each of the telescopes in the system. We describe a scheme to incorporate two simultaneous coupled pointing models in practice, and provide algorithms to use the scheme for accurate blind pointing and tracking. We provide an on-sky test of our procedures, showing that the use of a double pointing model can provide significant improvements in pointing and tracking for systems with external guiders, and thus can be a valuable tool in systems with significant relative flexures between the guider and the main instrument.

Accurate pointing and guiding of a telescope is a task of utmost importance for any observatory. Achieving such a goal entails a transformation from the mean coordinates and velocities of an astronomical target at a given epoch and equinox to a coordinate system attached to the mechanical axes of a given telescope. Most of these transformations can be accurately carried out based on astronomical and geographical data, and geometric considerations.

If a telescope could be made that was a perfectly aligned rigid body, ultimately the transformation required for pointing the telescope to a given astronomical source would entail transforming the right ascension, declination and proper motion of a given source into the corresponding angles of the telescope mechanical axes, either an altitude-azimuth or hour angle-declination pair, at the time of the observations. Alas, some level of misalignment and flexure is unavoidable. Therefore, an additional transformation, done with a so-called *pointing model* [12], is made to go from the topocentric hour, angle and altitude predicted from the astronomical coordinates to the actual positions at which the telescope axes angles have to be set. The pointing model corrections will depend on where the telescope points to. Such corrections have been implemented either as lookup tables that are interpolated, or as analytic models depending on a number of adjustable parameters, which are determined by pointing at a number of stars evenly distributed in the sky and minimizing the difference between the catalog stellar coordinates and those predicted by the model.

Once an accurate pointing is achieved, the telescope needs to track objects accurately across the night sky. In some sense, this is not a different problem than pointing, as tracking can be thought as just as series of movements determined by the evolution of pointing with time. That said, any remaining errors in the pointing model will as a rule accumulate as we track, resulting on a drift of the astronomical sources in the detector. Lacking a perfect mechanical systems, the solution to this drift is to provide a *guiding* system, with which the centroid of a bright source is monitored at a relatively high frequency (typically in the range 1–10 Hz) and any drifts on this source are used to correct the pointing of the telescope via a closed loop control.

The positioning control loop is a Proportional Integral Derivative (PID) controller we present in Fig.4.2.1. We can see that the controller is fed with a model and features a feed forward predictive capability as explained in [108]. In practice the input of the system is the target position of the axis. From this position the model computes a target position achievable by the system depending on a set of parameters, representing the behaviour of the axis. This set of parameters is composed of the previous positions and speeds, the maximum speed, and the maximum acceleration and deceleration of the device.

Additionally, the feedforward facility uses this same set of parameters and, using the previous target position, computes an achievable target speed of the system, which is added to the output of the PID controller. This sum is then fed as an input speed to the axis. The presence of the model and of the feedforward loop reduces to the maximum the input error r_m of the PID. This minimizes the integrated error, which avoids oscillations and overshoots, but remains very responsive in order to minimize the slew time between a target and the next one. In practice such an approach is much more efficient than a anti-windups strategy which only consists in saturating the integrated error.

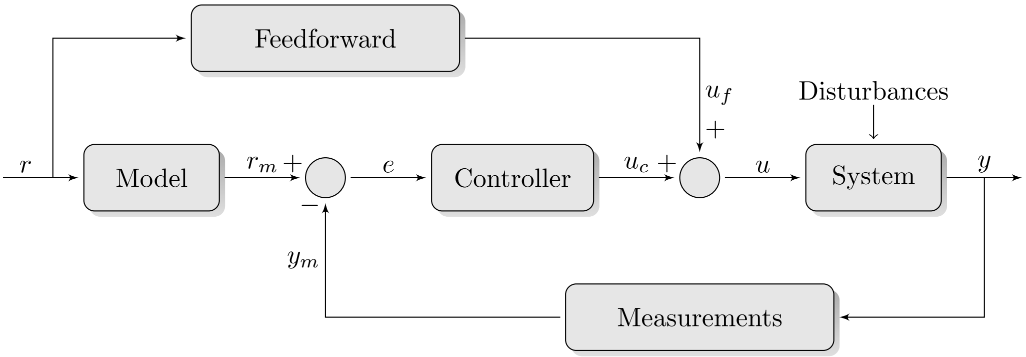


Figure 4.2.1: Model-based PID with Feedforward Axis Control structure

When considering instruments mounted on large professional telescopes the guiding is usually done using stars very close to the instrument's field of view, via imagers that gather light in the outskirts of the center of the focal plane. Even in this most favorable case there will be differences between the drifts of the guide stars and the center of the focal plane, and more than one guide star is required to avoid, for example, field rotation [12]. A minor complication with this setup is to find a proper guiding star which is bright enough to have a proper centroid computation at small exposures without saturating the detector, which must also be close to the target. Since the guiding field is usually a few arcminutes across, a proper guiding star may not be found easily.

Thus, the most usual guiding setup for small telescopes is to have a second smaller telescope attached to the main one. By having a small telescope with a wider field finding appropriate guide stars is not an issue in this setup, but the axis alignment and flexures of the small telescope need not be identical to those of the bigger telescope. Therefore, discrepancies between the pointing corrections needed for the guiding star and science targets can be significantly larger than when having a guider in the same focal plane of the instrument, resulting usually in noticeable drifts during long exposures. We note that guiding can also be done using the science frames, in the sense that we can keep a series of exposures aligned by computing the needed corrections to bring back the image to a given reference image of the field, a scheme successfully implemented in the DONUTS software package [109]. While this algorithm works well in the short exposure regime or a very robust mechanical setup, it fundamentally cannot correct for drift within the exposure, so it is in general not suited for long exposures. Thus, the fundamental problem when using a second guiding telescope is that the pointing model of the main telescope will not be accurate enough for the second guider telescope, which is the one determining the correction.

In this work we describe and test on sky the use of a double pointing model that allows accurate tracking when using a second guiding telescope by accounting for the differences in the alignment and flexures of the axes between the main and guider telescopes. The work is structured as follows. In § 4.2.1 we describe the double pointing model as states in a telescope pointing machine; in § 4.2.2 we describe the implementation of the presented double pointing model to two real-world applications, namely blind acquisition and autoguiding; in § 4.2.3 we present the on-sky performance of the autoguiding application, and finally in § 4.2.4 we conclude.

4.2.1 Double Pointing Model as New States in a Telescope Pointing Machine

The transformation from mean celestial coordinates to topocentric hour angle (h) and declination δ (or azimuth β and altitude θ for an alt-az telescope) can be done using several routines. Among these, the *Telescope Pointing Machine* [hereafter TPM,[1]] stands out for its speed, compactness and portability. It is designed as a software *state machine*, i.e., it defines a table of possible states corresponding to state vectors, which are 6-element vectors specifying position and velocity of a given target. All the knowledge of the TPM is encoded in a table indexed by the initial and final desired states. Each entry in the table, labeled by one of the states defined in the machine, gives an operation connecting to the nearest state, and the label of the resulting new state, and by following iteratively this procedure until the desired state is reached any operation can be efficiently encoded. This scheme has many advantages, but the most important for our purposes is that it is very easy to add new states: you just need to define a new state, and provide the transformation (and its inverse) to the nearest state *only*, not to all existing states.

In what follows we will concentrate in the case of an equatorial mount, but all that follows is equally applicable to an alt-az system (we will indicate the required modifications below). The first step when pointing is a series of transformations required to go from heliocentric mean right ascension α and declination δ to topocentric observed h and δ at a given observatory. The relevant states (denoted by the letter S followed by a number) and transitions (denoted by T followed by a number) in the TPM library for the equatorial mount are illustrated in Figure 5.4.4. The arrows in the transitions indicate the direction of the direct transformation as defined in the TPM, while going in the opposite direction requires the inverse transformation. Once we have the topocentric observed coordinates (h, δ), as explained in § **??** we have to convert to the hour angle and declination of each of the mechanical axes, $(h_{\text{axis}}, \delta_{\text{axis}})$, using a pointing model. In the context of TPM, this can be implemented by just adding an additional state. Given that we have both a guiding and a main telescope, we propose to add *two* new states, each associated with its own pointing model and corresponding to each of the telescopes, as illustrated in Figure 4.2.3.

The basic idea we propose in this work is to use the new states, which for an equatorial mount are connected to a single state S20 in TPM (topocentric observed h and δ), in order to properly propagate guiding information to the main instrument. We note that there is one possible final state for each pointing model. Depending on which instrument in the system we want to use for tracking and which for pointing, we only have to select which is the desired final state. The telescope control system will then use the target hour angle and declination $(h_{\text{axis,a}}, \delta_{\text{axis,a}})$ of state S23a if we want to track and point for the main instrument, or those of state S23b if we want the telescope to aim at a particular position for the guider telescope. The target hour angle and declination of the desired state will be recomputed at each iteration of the control loop, recalculating in the process the new target speed for each of these axes.

For the case of an alt-az mount, the difference is that the new states are incorporated differently into the TPM scheme. We show in Figure 4.2.4 how the pointing model states would be incorporated in this case. Other than this difference, all that follows applies just by considering alt-az pairs of angles for the mounts instead of the equatorial ones.

4.2.2 Applications of a Double Pointing Model: Blind Acquisition and Autoguiding

In this section we will present two applications of the multiple pointing model approach proposed in Figure 4.2.3 to two real-world cases: blind high precision pointing and autoguiding.

4.2.2.1 Calculation of the exact position by astrometrical reduction

When using small telescopes not equipped with high resolution on-axis encoders, residuals of the pointing model can be rather high due to the effects of different backlashes and periodical errors from the mount and the motors which cannot be taken into account with a pointing model. As a result, when the main instrument is an imager with a rather small field of view, or a spectrograph where a source has to be placed into a fiber or slit, it can be hard in practice to directly point towards the target even in presence of a good pointing model. In some cases it can be even impossible to automatically compute a pointing offset using a conventional astrometry software as `astrometry.net` [110], due to the limited size of the field or the lack of stars in the image. Equipped with a guider telescope and our proposed double pointing model, we put forward the

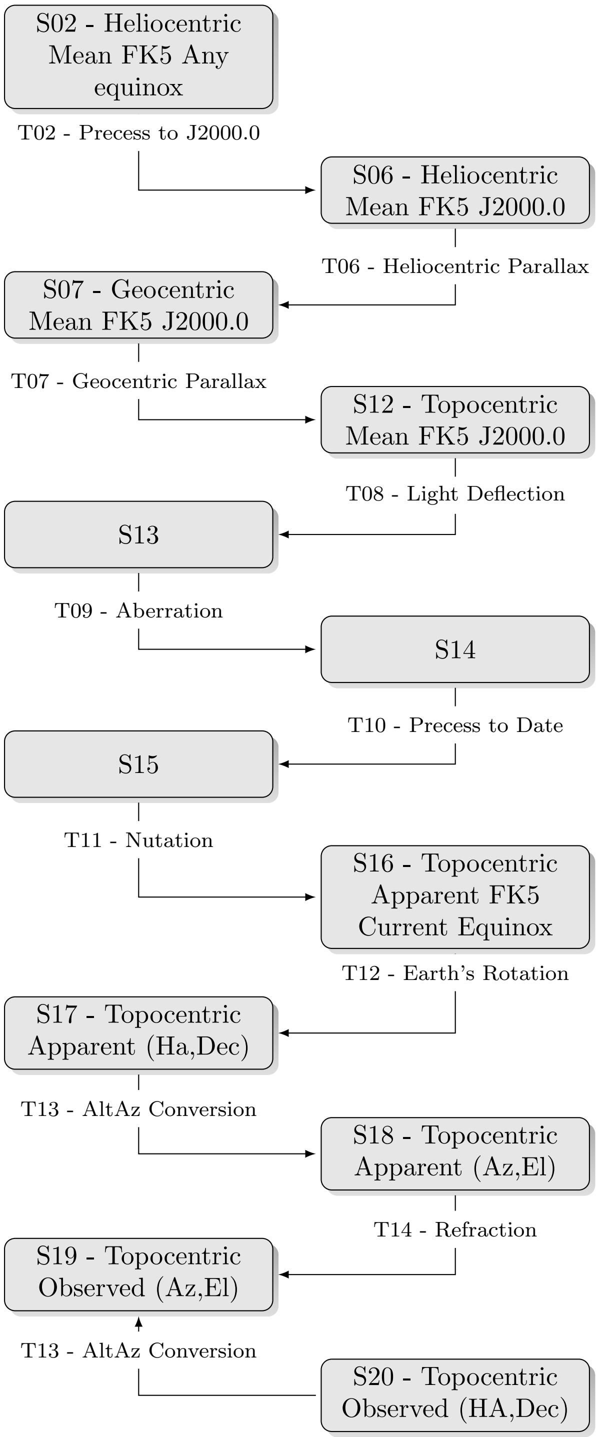

S02 - Heliocentric Mean FK5 Any equinox
T02 - Precess to J2000.0
S06 - Heliocentric Mean FK5 J2000.0
T06 - Heliocentric Parallax
S07 - Geocentric Mean FK5 J2000.0
T07 - Geocentric Parallax
S12 - Topocentric Mean FK5 J2000.0
T08 - Light Deflection
S13
T09 - Aberration
S14
T10 - Precess to Date
S15
T11 - Nutation
S16 - Topocentric Apparent FK5 Current Equinox
T12 - Earth's Rotation
S17 - Topocentric Apparent (Ha,Dec)
T13 - AltAz Conversion
S18 - Topocentric Apparent (Az,El)
T14 - Refraction
S19 - Topocentric Observed (Az,El)
T13 - AltAz Conversion
S20 - Topocentric Observed (HA,Dec)

Figure 4.2.2: Most useful pointing machine states and transitions according to the TPM definitions [1].

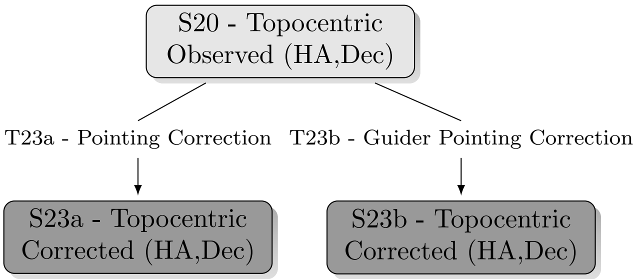

S20 - Topocentric Observed (HA,Dec)
T23a - Pointing Correction
T23b - Guider Pointing Correction
S23a - Topocentric Corrected (HA,Dec)
S23b - Topocentric Corrected (HA,Dec)

Figure 4.2.3: New states S23a and S23b added to the TPM (in dark gray) and their associated transitions for the case of an equatorial mount.

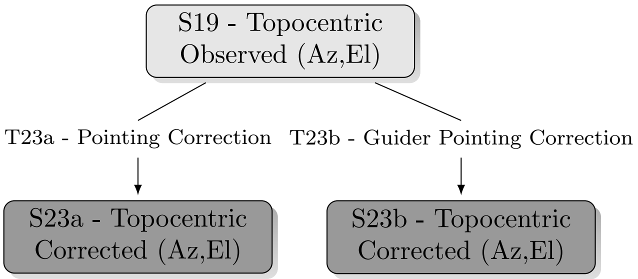

S19 - Topocentric Observed (Az,El)
T23a - Pointing Correction
T23b - Guider Pointing Correction
S23a - Topocentric Corrected (Az,El)
S23b - Topocentric Corrected (Az,El)

Figure 4.2.4: New states S23a and S23b added to the telescope pointing machine (in dark gray) and their associated transitions for the case of an alt-az mount.

following sequence of 12 steps using the secondary pointing model and its wide field imager in order to blindly send the target very close to the field center or the spectrograph slit of the main instrument:

1. Set the state S06 of a TPM state machine, we denote as SM_1 as the *target* state of the telescope control loop. At every iteration of the loop, the control algorithm will transform the target state S06 to state S23a. The output position of S23a is then used as the current floating target position of the telescope's drives. This means in practice that we will point and track the telescope according to the main imager.

2. Take an image I_1 with the guiding (wide field) telescope.

3. Store the current celestial coordinates of the state S23a of the the machine SM_1 in a variable P_1, as expressed in equation (4.2.1), below this list of actions.

4. Initialize a new state machine SM_2 with the coordinates of S23a, i.e. P_1.

5. Transform the state of SM_2 from the state S23a to the state s23b representing the guider telescope and store the position of the state S23b in a variable P_2, as expressed in equation(4.2.2).

6. Transform from the state S23b to the heliocentric mean FK5 J2000 coordinates represented by state S06, in order to obtain the expected celestial position P_3 of the guider, as expressed in equation (4.2.3).

7. Build the expected world coordinate system (WCS) of the guider imager according to the coordinates of P_3.

8. Run an astrometry software such as `astrometry.net` [110] or similar on the guider image in order to obtain the *actual* (real) coordinates of the guider imager, which we denote as P_4, as expressed in equation (4.2.4)

9. Set the position vector of the machine SM_2, which is in state S06, to P_4.

10. Transform the state of SM_2 back to S23b in order to obtain the axis coordinates P_5 of the mount that correspond to position P_4 in the guider image, according to equation (4.2.5).

11. Compute the pointing offset dP, the difference between the expected and real axes coordinates of the guider, according to equation (4.2.6).

12. Apply the offset to the target coordinates P_1 in order to get the final, corrected coordinates P_6 of the state machine SM_1, according to equation (4.2.7).

$$P_{1,\mathrm{S23a}} = \begin{bmatrix} h_{\mathrm{current}} \\ \delta_{\mathrm{current}} \end{bmatrix}_{\mathrm{Imager}} \tag{4.2.1}$$

$$P_{2,\mathrm{S23b}} = \begin{bmatrix} h_{\mathrm{current}} \\ \delta_{\mathrm{current}} \end{bmatrix}_{\mathrm{Guider}} \tag{4.2.2}$$

$$P_{3,\mathrm{S06}} = \begin{bmatrix} \alpha_{\mathrm{expected}} \\ \delta_{\mathrm{expected}} \end{bmatrix}_{\mathrm{Guider}} \tag{4.2.3}$$

$$P_{4,\mathrm{S06}} = \begin{bmatrix} \alpha_{\mathrm{real}} \\ \delta_{\mathrm{real}} \end{bmatrix}_{\mathrm{Guider}} \tag{4.2.4}$$

$$P_{5,\mathrm{S23b}} = \begin{bmatrix} h_{\mathrm{real}} \\ \delta_{\mathrm{real}} \end{bmatrix}_{\mathrm{Guider}} \tag{4.2.5}$$

$$dP_{\mathrm{S23b}} = P_{5,\mathrm{S23b}} - P_{2,\mathrm{S23b}} \tag{4.2.6}$$

$$P_{6,\mathrm{S23a}} = P_{1,\mathrm{S23a}} + dP_{\mathrm{S23b}} \tag{4.2.7}$$

As a result, in a case like this, where for example we cannot have an astrometric reduction on the main instrument, the pointing error can be improved by accounting explicitly via the two pointing models the differences between the guider and main telescopes. As long as the calibration data for both pointing models is taken simultaneously, the measuring points are equally distributed, and there is no erratic differential movements between both telescopes, the residuals of the pointing quality should be determined by the standard deviation of the difference of both pointing models.

4.2.2.2 Autoguiding

A similar approach may be used for autoguiding, which as stated in §1 can be seen as differential pointing. In what follows, we assume that the main instrument was properly recentered before the start of autoguiding. At the end of the procedure described in the last section, the imager field will be centered to a precision characterized by the typical difference in accuracy between the two pointing models. To center exactly the imager or a target in a spectrograph slit one can use several methods, for example centroiding on the target, which is by now very close to the center given the former procedures have been properly performed. We denote the difference in position between the fine-tuned center and the center obtained by using the method of the previous section as dP_{init}. Note that if we run the centering procedure of the previous section after fine-tuning the center we will be told by the procedure that we need to correct by this value to center (assuming an insignificant amount of time has ellapsed since we run the pointing procedure, otherwise the value will start evolving as the sky moves and we start sampling different regions in the pointing model). This just reflects that we are working at the precision limit of the pointing models and the value will be of a magnitude typical of the pointing model errors.

Once we have determined dP_{init}, autoguiding is run on a continuous loop in which we repeat successively steps 1 to 11 of the list we used in the previous section for exact astrometrical reduction. Step 12 is replaced by a similar operation, where the offset applied to the machine SM_1 is defined by dP_{guiding} as expressed in equations (4.2.8) and (4.2.9). These two equations replace equations 4.2.6 and 4.2.7 above, the value of dP_{S23b} is given by equation 4.2.6.

$$dP_{\mathrm{guiding}} = dP_{\mathrm{S23b}} - dP_{\mathrm{init}} \tag{4.2.8}$$

$$P_{6,S23_a} = P1_{1,S23_a} + dP_{\mathrm{guiding}} \tag{4.2.9}$$

The process is thus essentially the same we used for the blind pointing procedure, but taking into consideration the initial offset dP_{init}. Performing the process in this way, the autoguider will guide on a *drifting star* in the guider telescope, and this drift will correspond to the evolving differences between the pointing models given by the instances of states $SM23a$ and $SM23b$. This is, the commanded drift of the star in the guider is such that the star stays centered on the main imager.

4.2.3 On-Sky Performance

We tested the schemes proposed above with a 40cm diameter telescope of the Pontificia Universidad Catolica de Chile equipped with a SBig STL1001 CCD Camera and an external guider camera. The main imager is installed on the Cassegrain focus of the 40cm f/19.5 telescope, giving a field of view of ≈ 8'×8', while the guider imager is at the focus of a 80mm f/5 refractor installed in parallel to the main telescope, with a field of view of 30'×40'. Pointing models using ≈ 100 stars distributed across the whole sky (but at least 40 degrees above the horizon) were constructed for both instruments. These models have residuals of ≈ 30 arcsec each.

In order to test the effectiveness of the blind pointing approach presented, we pointed at 10 different fields and found a 5 arcsec rms difference between the commanded and actual coordinates. As we noted in the previous section, this rms corresponds to the difference between the pointing models of the main imager and the guider. Even though each pointing model has separate errors of the order of 30 arcsec, the difference of the two pointing models has typical deviations of only ≈ 5 arcsec. The reason is that the models are very correlated due to sharing the same mount, making the difference between them significantly more precise than their individual absolute pointing accuracy. We conclude from this exercise that using a wide-field guiding imager and a formalism incorporating two pointing models leads to a significant advantage in absolute pointing accuracy.

To test the effectiveness of the scheme for guiding over large amounts of time, we set the system to a position of hour angle -1h30 at the beginning of the observation, and a position of declination close to the latitude of the observing site (≈-30 degrees). Autoguiding offsets were commanded every 10 seconds using the proposed scheme for a total of 30 minutes. In the meantime, the imager on the main telescope was taking images every minute. In Figure 4.2.5 we plot the accumulated sum of the offsets sent to the mount by the telescope control system (TCS) (blue curve),corresponding to the corrected drift, along with the drift measured on the guider (red) and on the main imager (green) according to the coordinates obtained from astrometry after applying the corrections obtained from our proposal. We can observe from Figure 4.2.5 that the drift on the main imager never went further than 2.5 arcseconds from its nominal position, while the commanded offset wandered up to ≈ 15 arcsec. This means that the star being imaged would have moved up to about 15 arcsec if we were observing without any autoguiding. Additionally, the measured drift on the guider moved steadily accumulating ≈ 10 arcsec. This means that the guiding star would have drifted about 10 arcseconds over this same period if guiding was done by keeping a star fixed on the guider telescope. Using our approach, the guided drift was an order of magnitude smaller than the unguided drift and the drift obtained with the typical guiding method. In other words, if the system had been guided with the external guider using the classical method (which consists of keeping the guide star in the same pixel), the main imager would have observed a drift of 10 arcseconds during the observation due to differential flexures of the mounts and other model imperfections, which has not been the case due to the use of a double pointing model.

4.2.4 Conclusions

We have proposed and implemented a new method using multiple pointing models which is able to perform high precision recentering and autoguiding of a small telescope using an external imaging train installed in parallel to the main optical tube assembly. Precision autoguiding usually is achieved having a guide sensor in the same focal plane of the main imager, either using a multichip imager with a dedicated chip for this use, or using a pick-off mirror for guiding purposes. Both of these solutions need a reduced field of view for the guider, which cannot be used for recentering purposes, and can be limited in finding guide stars on low density fields or when used behind a narrowband filter. With the solution we have proposed, both high accuracy recentering and tracking can be done using the guiding imager, which is not affected at all by the filter we use on

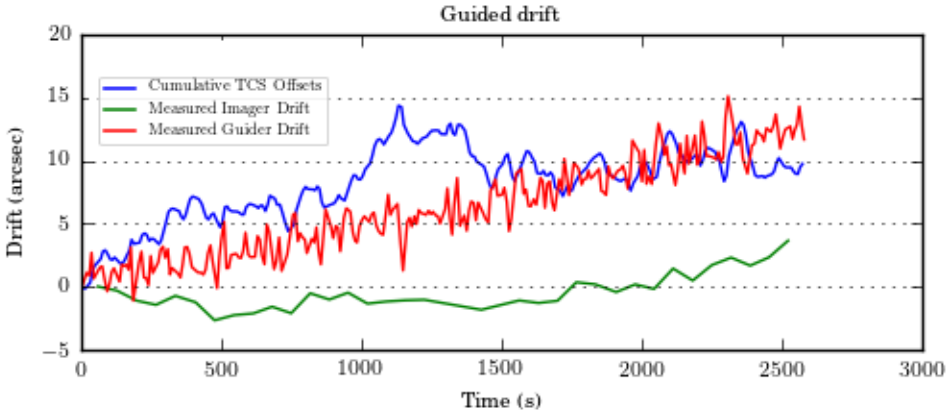


Figure 4.2.5: Right ascension drift in the main imager (green) (corrected by our proposal), measured drift in the guider (red), and accumulated corrections sent to the mount (blue).

the main optical train or the position of the star field under analysis. Since we can use a wide field telescope, we can choose the external guider sensor size and related optics so that a much wider set of guiding stars can be considered, and astrometry can be performed regardless of the target field or moonlight pollution. An improvement of a whole order of magnitude in the quality of the tracking in conventional autoguiding application was experimentally demonstrated.

Our scheme is specially useful for telescopes where total cost is a major design consideration. In our test setup the flexures of the telescope and the differential flexures between the main telescope and the guider are significant, and our solution solves this problem in an effective and low-cost fashion, via software. One can always improve the mechanics and achieve improved overall performance without using multiple pointing models, but our scheme provides an additional tool to increase the performance of small telescope observations, which have played an increasingly prominent role in Astronomy in the late years due in part to the increase in their use fueled by research in exoplanets.

4.3 Portable Observatory Software Arquitecture

In the State of the Art Section of this Thesis, we discussed the efforts currently ongoing to make the peripheral layer of small telescopes more device independent. The idea of a peripheral hub abstraction layer was introduced for the first time with the **AStronomy Common Object Model (ASCOM)** [111] platform available under single computer Windows platforms. Linux-based systems were handled later in a comparable manner by the **Instrument Neutral Distributed Interface (INDI)** [112, 113] software.

Both solutions propose a list of drivers for the most common peripherals used in Astronomy-related projects, grouped by types of interfaces (camera, mount, weather station, focuser, rotators, and so on.). Each kind of interface proposes then a public common **Application Programming Interface (API)**, which is expected to be independent of the hardware used. The interface to these **API**s is done by a software hub which asynchronously translates requests from the common **API** to the device of the specific peripheral. As a result, client's software can connect and send orders to the **API** regardless of the brand or driver of the hardware connected to the computer.

Under Windows systems, **ASCOM** allows device hardware drivers and clients to be programmed in any Microsoft .NET language since the communication is done using Windows COM objects. On the other hand, it does not provide a network distribution layer of these objects. As a consequence, when using an astronomical setup handled using **ASCOM**, every peripheral and client have to be installed in the same computer unless a specific driver or a specific client including a dedicated workaround to communicate with other devices on the network is present, which is not always the case. This bottleneck represents the major limitation of the **ASCOM** platform in the moment of integrating it in a remotely controlled environment, which may involve an observatory local network, susceptible to have some common peripherals in the network.

As of Linux systems, INDI provides a hub interface similar to the one proposed by **ASCOM**. This interface is developed in C++ and drivers for new peripherals can be installed on the folder where the **INDI** hub is running. These drivers have to be compiled in C++ and need to be run on a Linux system. Regarding the communication with the clients, the hub proposes an XML based communication language exported through TCP/IP sockets. These communication protocols allow running the client on any platform regardless of the system or machine the server is running on. This feature represents the breakthrough of **INDI** versus what is proposed by **ASCOM**. However, even if **INDI** allows to distribute peripherals and their respective drivers and clients over multiple machines on a local network, it is still necessary to develop the drivers in a common language running on Linux machines, which is not evident.

In this Section, we will first present a novel software architecture based on the **Internet Communications Engine (ICE)** Library and its implementation in the management of observatory peripherals. The main advantage over previously defined existing architectures is that the system is genuinely distributed, as far as both the software and the clients are entirely system independent. In the second part of this Section, we will present the list of peripherals and their inheritances currently handled by the proposed software, and how new types of peripherals may be added. Then, we will we present how all the configuration of a complete observatory can be grouped in a single XML file joint with its architecture and management. Finally, we will present how we can additionally include a fast web-socket based communication technique for each driver allowing real-time communication with clients over the web interface, which allows the client to communicate with any kind of device without the need to install any additional software on the client computer.

4.3.1 Objects Distribution Over the Network

As exposed, there are two primary standards of centralizing communications in the field of small telescopes and observatories automation which are known do be **ASCOM** and **INDI**. The centralization of communication being performed by a specific software called hub which will receive orders from various clients and jeopardize them to their respective drivers. **ASCOM** runs on Windows systems and proposes to export COM objects between applications. This COM objects can be retrieved from clients programmed in any .NET language. **ASCOM** is widely used among the amateur astronomers' community because most of the hardware vendors do propose compatible drivers. However, clients and hardware using **ASCOM** must imperatively run on the same Windows computer. The reason for this is that the as the **ASCOM** hub is a particular driver developed for each client which need to run on the same computer.

In order to communicate between peripherals, drivers and clients, we propose to use the **ZeroC-ICE** framework. When compared to **ASCOM**, **ZeroC-ICE** allows to get all the benefits **INDI** has proven over **ASCOM**, so:

- Clients and servers can run on different computers
- Clients can run on different platforms and operating systems
- The computing load of the system can be distributed over several computers across the network, and each of them may even be dedicated to a single task to increase the robustness and reliability of the system

On the other hand, it is better to use ICE rather than **INDI** for the following reasons:

- In **ZeroC-ICE** servers can run on different platforms and operating systems, which is not the case with **INDI** where the servers need to be run on Unix-based systems only. Additionally, all the **INDI** drivers must be programmed in c++ which may add unnecessary complexity to the coding of some peripherals. On the other side, the modularity of **ZeroC-ICE** allows as an example, that the graphical interfaces of all the cameras to be a single universal Python client exporting the interface as a web server. As a result, it can be accessed by any computer from any system without any need to installing anything on the user's computer. On the **INDI** driven system, the drivers necessarily run on Linux and have to be programmed in C. The Users have then to install a client on its computer, for all the peripherals, and every interface has to be programmed within the same software, and thus, within the same programming language. This makes the **ZeroC-ICE** interface much more versatile and much easier to use for both the programmer and the user.
- Servers and clients can also be programmed using any language, not being limited to just C++
- Servers can run in a standalone mode and do not need to be started through a node. As an example, a server can also uses client objects, and become a client to other servers, to which it can connect directly without going through third entities. This greatly simplifies the system architecture and the overall understanding of the relationships of a typically complex collection of peripherals
- The **ZeroC-ICE** interface also provides the use of callback functions of the network, which are not available in **INDI** and allow programming completely asynchronous objects.
- Finally, **ZeroC-ICE** provides the tool ICEGrid which allows to automatically check and keep needed servers running, or to start them as soon as they are needed, which also simplifies the programming of the complete architecture.

The implementation of the architecture is performed by defining every server object using the **Specification Langage for ICE (SLICE)**. **SLICE** is a simple definition object-oriented language in which we define member functions of every possible server object present in our setup. For the case of the test setup developed in this Thesis we will need the following objects to be specified:

Axis Motor controller: The motor server will be in charge of interacting with the Ipec motor Controller and will receive orders from the mount (target speeds and position requests)

Dome: A dome server will interact with the dome electronics to move it according to orders received from the mount, or from other peripherals, to open or close it. As a typical example, the weather station will be able to send directly an order to close the dome in case of possible rain.

Field Rotator This server will move the motor rotating the camera to position the field correctly to dynamically compensate for the field rotation which appears in the case of an alt-azimuthal telescope. As the dome, it needs to be synchronized with the mount server.

Focuser The focuser server will move the focus motor of the camera, under direct orders from the user after visual inspection of the images, or from the camera manager after automated focus analysis.

Filter Wheel The filter wheel will interact with the motor positioning the proper filter in front of the detector. It will receive orders from the Acquisition Manager

Mount The mount server contains the implementation of the Advanced Telescope Control System as defined in Chapter 4.2. As a consequence, it will continuously monitor the target object and transform the coordinates to the pointing machine state 23.

Camera The Camera Object includes all possible functions needed to handle a standard CCD. It will receive direct orders from the Acquisition Manager. Since most camera **APIs** are defined in C or C++ language, a significant part of these drivers will be programmed in this language. This will be an exception, as we chose to program all other servers in Python for convenience. Images are sent back to the Acquisition Manager using Callback functions, which allow utterly asynchronous handling of this critical peripheral. As a result, the camera and Acquisition manager objects will not remain blocked by any process while the camera is exposing or retrieving images from the device.

Acquisition Manager This server is the top level of the acquisition chain and should be able to connect to all the previously defined systems (directly or indirectly) to perform the observation of an object. It should be able to communicate with the camera to take an image and retrieve it, to process it, to analyze the data in to refocus or correct for the mount position if necessary and to image again as soon as the mount is on target. We can consider the acquisition manager as the chief director of all peripherals who makes them work together smoothly.

WeatherStation Above the other peripherals, the weather station can monitor for environmental condition and decide to send orders to the proper ones to protect the system if a dangerous weather condition is detected.

Another significant advantage of being able to distribute object servers on different computers is that it becomes simple to choose a way of placing the computers to avoid the cables to pass through moving mechanical parts. In our case, we identify three primary locations for placing computers allowing a minimal number of moving cables:

Concrete pier We chose to use a computer placed on the ground which will control peripherals which are not moving such as the azimuth axis controllers, the dome rotation or the weather station.

Mount fork A single board computer is placed on the mount's fork to control the elevation axis.

Telescope tube Another single board computer is placed on the telescope tube and will handle the camera, focuser, rotator and filter wheel.

Since we use this specific arrangement, only power cables for computers and peripherals need to be dragged to the mount and the tube of the telescopes, reducing the complexity of the setup and preventing from complications due to cable wrapping. All the communication can be performed wirelessly, using WiFi in our case. We represent in Fig.4.3.1 a schematic lateral view of an alt-azimuthal telescope and components putting in evidence this optimal positioning of the three computers to limit cabling passing through axes.

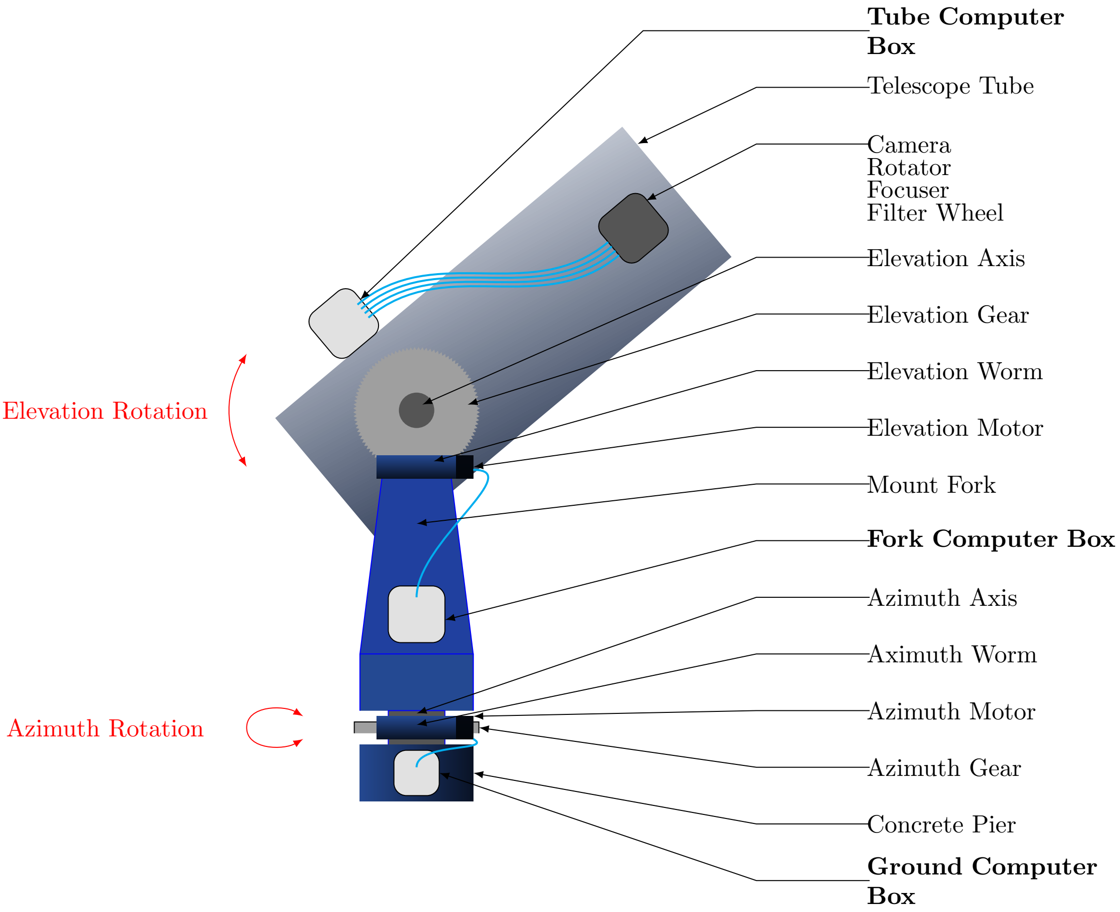


Figure 4.3.1: Alt-Azimuthal telescope and components with optimal placement of computers to limit the possibility of cable wraps. The software architecture allows maximizing intercommunication between peripherals using WiF, so we only need power cables in the mount and tube.

4.3.2 Object Architecture

Once we defined the standard hardware and software components of the telescope, we will now present a more detailed view of the communication between the different components and its deployment around the three computers we previously described. Hereafter we will use the Unified Modeling Language (UML) as defined in [114]. Fig.4.3.2 is a UML Class Diagram representing the eight previously defined types of objects for a standard configuration. There is a one to one correspondence with the objects defined in the Slice file. We represent each type as a separate class composed of three parts as follows:

- The class name
- A list of the most relevant class attributes
- A list of the most relevant class member functions with their respective parameters

Simply shaped arrows point towards the corresponding class it derivates from, while rhombohedric-shaped arrows point towards the classes containing the type of object connected to the departure of the the arrow. Rhombohedric arrows show on their side the number and names of objects instantiated in the class indicated by the arrow. It is essential to keep in mind that for simplicity of understanding and visualization only the most relevant components have been displayed. We can find an extended definition of each class and **SLICE** object in Annex A.
It is possible to see from this graph that the different objects which make the observatory work together communicate precisely in the same way as if they were a single Object Oriented program. However, the **ZeroC-ICE** Architecture we propose for the system, allows each of these components to be included in a separate softwares, and can even be programmed in different languages or run on different computers. For the case of our example, we present in Fig.4.3.3 the UML Deployment Diagram. This diagram shows the same objects presented previously in Fig.4.3.2,placed in the hardware environment with their associated servers and connections as follows:

- Red entities represent the hardware peripherals with their respective names
- Each grey entity corresponds to a control computer
- We represent the servers which are running on each computer inside their associated computer in green boxes with the name of the device server
- We represent physical hardware connections as continuous black lines between the peripheral and their respective driver. The Communication type is expressed on the server side next to the line
- We represent Inter-object communications using the **ZeroC-ICE** framework as blue lines. Each communication is composed of a server and a client. The same object can be a server and a client of other objects at the same time. As we can see, we express each client pointing to a specific server as a local class attribute in the UML Class Diagram. The TCP port used by Ice for each connection is represented next to the line on the server's side

4.3.3 Common Configuration File

When distributing multiple peripherals over a network, it soon appears the problem of obtaining a configuration of the system which could be straight-forwardly spread among the computers and client, but also which remains consistent when a change of configuration happens to each single peripheral. The simplest way to achieve centralization of the configuration of the complete system is to group the entire definitions in a single file. In a first approach, we implemented a system where we can express the definitions of every peripheral in an object-oriented filed expressed in Extensible Markup Language (XML) format. We would then copy this XML file with a similar name on each computer in the same folder. As a result, when a server would start on this computer, it would look and read this file to look for the proper server configuration and the execute it accordingly. The main advantage of this format is how quickly we can converted it into a Python or C++ object using a standard library, while it remains easily readable to a human eye, more than if we used the JavaScript Object Notation (JSON) format.

However, even if this is an efficient approach to the architecture of the system when it does not evolve, it may bring several issues when the configurations trend to change. Consider the case when we chagne a parameter one peripheral. The configuration will have to be changed consequently in the file on the local computer where it is running on. However, we will also have to synchronize it with the respective configuration files of every computer in the network to keep all the files consistent together. This step becomes a more and more tedious operation as the complexity of the overall system increases. We successfully managed to get around this problem by centralizing the complete configuration inside a single MySQL Database running on one single computer. Every peripheral driver at startup would connect to this database to retrieve its configuration, and act accordingly. Since the location and parameters of the database will not change frequently (or, in practice, will not change), it is possible in this case to store the configuration of the database itself

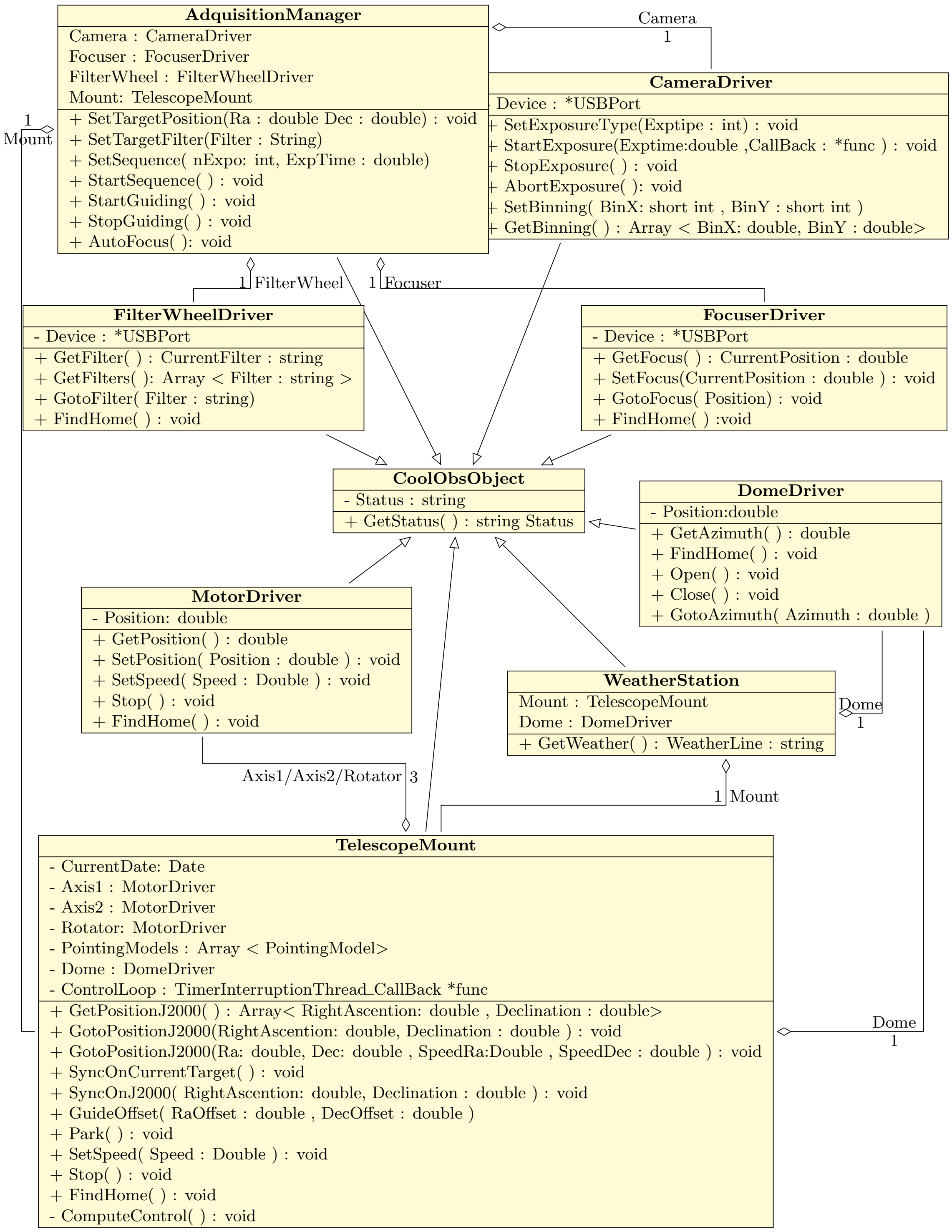

AdquisitionManager
Camera : CameraDriver
Focuser : FocuserDriver
FilterWheel : FilterWheelDriver
Mount: TelescopeMount
+ SetTargetPosition(Ra : double Dec : double) : void
+ SetTargetFilter(Filter : String)
+ SetSequence(nExpo: int, ExpTime : double)
+ StartSequence( ) : void
+ StartGuiding( ) : void
+ StopGuiding( ) : void
+ AutoFocus( ): void
Camera
1
CameraDriver
- Device : *USBPort
+ SetExposureType(Exptipe : int) : void
+ StartExposure(Exptime:double ,CallBack : *func) : void
+ StopExposure( ) : void
+ AbortExposure( ): void
+ SetBinning(BinX: short int , BinY : short int)
+ GetBinning( ) : Array < BinX: double, BinY : double>
1
Mount
1 FilterWheel
1 Focuser
FilterWheelDriver
- Device : *USBPort
+ GetFilter( ) : CurrentFilter : string
+ GetFilters( ): Array < Filter : string >
+ GotoFilter(Filter : string)
+ FindHome( ) : void
FocuserDriver
- Device : *USBPort
+ GetFocus( ) : CurrentPosition : double
+ SetFocus(CurrentPosition : double) : void
+ GotoFocus(Position) : void
+ FindHome( ) :void
CoolObsObject
- Status : string
+ GetStatus( ) : string Status
DomeDriver
- Position:double
+ GetAzimuth( ) : double
+ FindHome( ) : void
+ Open( ) : void
+ Close( ) : void
+ GotoAzimuth(Azimuth : double)
MotorDriver
- Position: double
+ GetPosition( ) : double
+ SetPosition(Position : double) : void
+ SetSpeed(Speed : Double) : void
+ Stop( ) : void
+ FindHome( ) : void
WeatherStation
Mount : TelescopeMount
Dome : DomeDriver
+ GetWeather( ) : WeatherLine : string
Dome
1
Axis1/Axis2/Rotator 3
1 Mount
TelescopeMount
- CurrentDate: Date
- Axis1 : MotorDriver
- Axis2 : MotorDriver
- Rotator: MotorDriver
- PointingModels : Array < PointingModel>
- Dome : DomeDriver
- ControlLoop : TimerInterruptionThread_CallBack *func
+ GetPositionJ2000( ) : Array< RightAscention: double , Declination : double>
+ GotoPositionJ2000(RightAscention: double, Declination : double) : void
+ GotoPositionJ2000(Ra: double, Dec: double , SpeedRa:Double , SpeedDec : double) : void
+ SyncOnCurrentTarget( ) : void
+ SyncOnJ2000(RightAscention: double, Declination : double) : void
+ GuideOffset(RaOffset : double , DecOffset : double)
+ Park( ) : void
+ SetSpeed(Speed : Double) : void
+ Stop( ) : void
+ FindHome( ) : void
- ComputeControl( ) : void
Dome
1

Figure 4.3.2: Simplified UML Class diagram of the CoolObs objects arquitecture

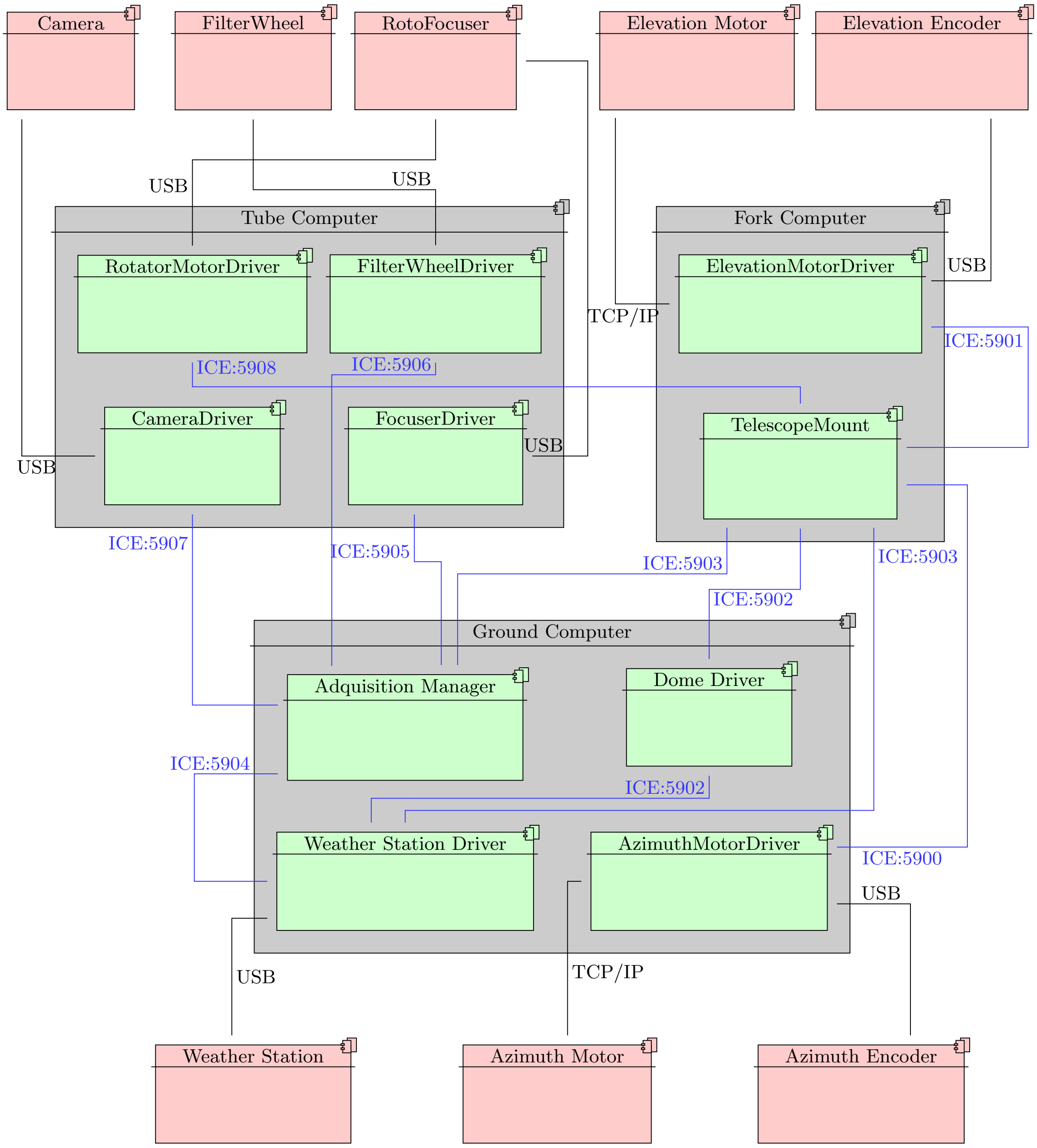


Figure 4.3.3: UML deployment diagram of the CoolObs objects arquitecture

in an XML document in each computer. The software will then first look for the configuration file on the local drive to get the parameters of the location and credentials of the database. Then use them to connect the database and retrieve the configuration of the wanted peripheral.

Then, we present the implementation of our configuration system in the 40cm Telescope at Santa Martina's Observatory (Pontificia Universidad Catolica de Chile, PUC). The XML file is stored in each computer in the folder `$HOME/.CoolObs/Config.xml`, encoded as shown in Fig.4.3.4, which is the usual configuration in the mentioned telescope.

```xml
<?xml version="1.0" encoding="UTF-8"?>

<CoolObs>
  <DataBase>
    <Name>puc40</Name>
    <User>puc40</User>
    <Password>pucobs1</Password>
    <Ip>192.168.0.59</Ip>
    <Port>3306</Port>
    <Table>Puc40Peripherals</Table>
  </DataBase>
</CoolObs>
```

Figure 4.3.4: Example of XML configuration file for the PUC40 telescope installed at Santa Martina's Observatory.

Regarding the implementation of the database itself considering a site with multiple telescopes, we decided to group only one database per site, which would have one master table of peripherals per telescope. This master table of peripherals would group every device composing the telescope together with the table where we can find its extended definition. Then we can define one shared table format per each type of peripheral.

The figure 4.3.5 shows a real example of the database implemented at the Santa Martina's Observatory in Chile. Every table represents a type of peripheral, in which each line represents one peripheral with its associated properties. All the peripherals of the observatory are expressed this way in the same database. It is then essential to notice the existence of two different types of table:

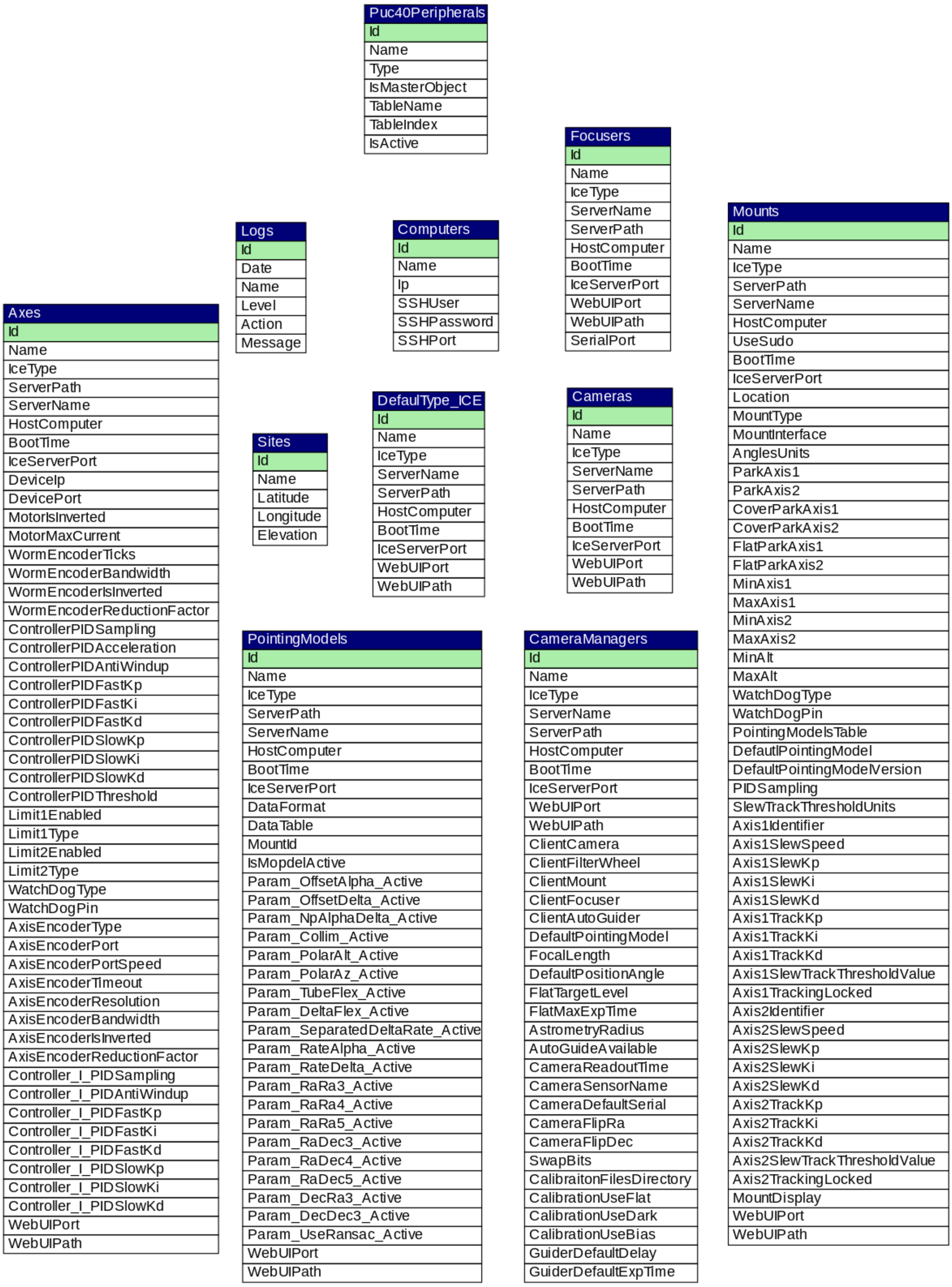
The Puc40Peripherals Table: We define one specific table like this one per telescope on the site. In this table, a list of all peripherals related to one single telescope is detailed. As a consequence, the software in charge of starting the system would only have to look over this table to start the necessary components. The table has one line per telescope component, and we define each component by five fields:

- Name, which is a unique identifier for each peripheral.
- The type of component.
- IsMasterObject is an Integer value defining if the object has slaves. A value of "0" means this object does not need to be started at first, "1" means that the object will start the complete system, "2" means it will be started by objects of index "1" after these are ready, "3" means it will be started by objects of index "2" and so on.
- TableName is a text defining the name of the table in which we can find the complete description and parameters of the related object.
- Table Index is an Integer value showing the index in which the proper definition of the object is, in case there would be multiple configurations. A -1 value means we always take into account the latest configuration responding to the object name.
- IsActive is a boolean value defining whether the object must be started or not.

A real example of the Puc40Peripherals table from Santa Martina's Observatory, where all the components of a 16" telescope are properly stored is presented in Fig.4.3.6 to show a practical implementation of the proposed architecture.

The Logs Table: All the logs of all peripherals are stored in this table in their order of appearance, with one line per log. The table is composed of 5 columns:

- Date represents the instant of the event
- Name is the name of the peripheral initiating the current log line
- Level represents the type of log information (INFO, WARNING, ERROR, etc.)
- Action field is a text backing up the function from which the log was initiated
- Message is the corresponding logged message



Puc40Peripherals
Id
Name
Type
IsMasterObject
TableName
TableIndex
IsActive

Focusers
Id
Name
IceType
ServerName
ServerPath
HostComputer
BootTime
IceServerPort
WebUIPort
WebUIPath
SerialPort

Logs
Id
Date
Name
Level
Action
Message

Computers
Id
Name
Ip
SSHUser
SSHPassword
SSHPort

Mounts
Id
Name
IceType
ServerPath
ServerName
HostComputer
UseSudo
BootTime
IceServerPort
Location
MountType
MountInterface
AnglesUnits
ParkAxis1
ParkAxis2
CoverParkAxis1
CoverParkAxis2
FlatParkAxis1
FlatParkAxis2
MinAxis1
MaxAxis1
MinAxis2
MaxAxis2
MinAlt
MaxAlt
WatchDogType
WatchDogPin
PointingModelsTable
DefautlPointingModel
DefaultPointingModelVersion
PIDSampling
SlewTrackThresholdUnits
Axis1Identifier
Axis1SlewSpeed
Axis1SlewKp
Axis1SlewKi
Axis1SlewKd
Axis1TrackKp
Axis1TrackKi
Axis1TrackKd
Axis1SlewTrackThresholdValue
Axis1TrackingLocked
Axis2Identifier
Axis2SlewSpeed
Axis2SlewKp
Axis2SlewKi
Axis2SlewKd
Axis2TrackKp
Axis2TrackKi
Axis2TrackKd
Axis2SlewTrackThresholdValue
Axis2TrackingLocked
MountDisplay
WebUIPort
WebUIPath

Axes
Id
Name
IceType
ServerPath
ServerName
HostComputer
BootTime
IceServerPort
DeviceIp
DevicePort
MotorIsInverted
MotorMaxCurrent
WormEncoderTicks
WormEncoderBandwidth
WormEncoderIsInverted
WormEncoderReductionFactor
ControllerPIDSampling
ControllerPIDAcceleration
ControllerPIDAntiWindup
ControllerPIDFastKp
ControllerPIDFastKi
ControllerPIDFastKd
ControllerPIDSlowKp
ControllerPIDSlowKi
ControllerPIDSlowKd
ControllerPIDThreshold
Limit1Enabled
Limit1Type
Limit2Enabled
Limit2Type
WatchDogType
WatchDogPin
AxisEncoderType
AxisEncoderPort
AxisEncoderPortSpeed
AxisEncoderTimeout
AxisEncoderResolution
AxisEncoderBandwidth
AxisEncoderIsInverted
AxisEncoderReductionFactor
Controller_I_PIDSampling
Controller_I_PIDAntiWindup
Controller_I_PIDFastKp
Controller_I_PIDFastKi
Controller_I_PIDFastKd
Controller_I_PIDSlowKp
Controller_I_PIDSlowKi
Controller_I_PIDSlowKd
WebUIPort
WebUIPath

Sites
Id
Name
Latitude
Longitude
Elevation

DefaulType_ICE
Id
Name
IceType
ServerName
ServerPath
HostComputer
BootTime
IceServerPort
WebUIPort
WebUIPath

Cameras
Id
Name
IceType
ServerName
ServerPath
HostComputer
BootTime
IceServerPort
WebUIPort
WebUIPath

PointingModels
Id
Name
IceType
ServerPath
ServerName
HostComputer
BootTime
IceServerPort
DataFormat
DataTable
MountId
IsMopdelActive
Param_OffsetAlpha_Active
Param_OffsetDelta_Active
Param_NpAlphaDelta_Active
Param_Collim_Active
Param_PolarAlt_Active
Param_PolarAz_Active
Param_TubeFlex_Active
Param_DeltaFlex_Active
Param_SeparatedDeltaRate_Active
Param_RateAlpha_Active
Param_RateDelta_Active
Param_RaRa3_Active
Param_RaRa4_Active
Param_RaRa5_Active
Param_RaDec3_Active
Param_RaDec4_Active
Param_RaDec5_Active
Param_DecRa3_Active
Param_DecDec3_Active
Param_UseRansac_Active
WebUIPort
WebUIPath

CameraManagers
Id
Name
IceType
ServerName
ServerPath
HostComputer
BootTime
IceServerPort
WebUIPort
WebUIPath
ClientCamera
ClientFilterWheel
ClientMount
ClientFocuser
ClientAutoGuider
DefaultPointingModel
FocalLength
DefaultPositionAngle
FlatTargetLevel
FlatMaxExpTime
AstrometryRadius
AutoGuideAvailable
CameraReadoutTime
CameraSensorName
CameraDefaultSerial
CameraFlipRa
CameraFlipDec
SwapBits
CalibraitonFilesDirectory
CalibrationUseFlat
CalibrationUseDark
CalibrationUseBias
GuiderDefaultDelay
GuiderDefaultExpTime

Figure 4.3.5: Observatory's Database tables structure at Santa Martinas Observatory

Id	Name	Type	IsMasterObject	TableName	TableIndex	IsActive
1	Puc40Mount	GenericMount	0	Mounts	-1	1
3	Puc40IpecHa	Ipec	0	Axes	-1	1
4	Puc40IpecDec	Ipec	0	Axes	-1	1
5	Puc40PointingModel	PointingModel	0	PointingModels	-1	1
6	Puc40GuiderPointingModel	PointingModel	0	PointingModels	-1	1
7	Puc40C11PointingModel	PointingModel	0	PointingModels	-1	1
8	SbigSpectroServer	CoolObsSimpleCamera	0	Cameras	-1	1
9	SbigServer	CoolObsCamera	0	Cameras	-1	1
10	STL1001Manager	CoolObsCameraManager	1	CameraManagers	-1	1
11	STL1001GManager	CoolObsCameraManager	0	CameraManagers	-1	0
12	STiGManager	CoolObsCameraManager	0	CameraManagers	-1	1
13	ST7Manager	CoolObsCameraManager	2	CameraManagers	-1	0
14	ST7GManager	CoolObsCamraManager	0	CameraManagers	-1	0
15	Puc40Focuser	Focuser	0	Focusers	-1	1
16	SpectroFocuser	Focsuer	0	Focusers	-1	0
17	ObsPuc	Site	0	Sites	-1	1
18	Puc40Ha	Computer	0	Computers	-1	1
19	Puc40Dec	Computer	0	Computers	-1	1
20	Puc40Tcs	Computer	0	Computers	-1	1
21	Puc40Nuc	Computer	0	Computers	-1	1
22	Puc40Guider	Computer	0	Computers	-1	1
23	ODUCLogger	Logger	0	Logs	-1	1
24	Puc40Focuser2	Focuser	0	Focusers	-1	1

Figure 4.3.6: Content of table Puc40Peripherals

4.3.4 Browser Access

Once a logical and standardized architecture has been implemented, the last step of the optimization of the complete system involves improving the interaction of the hardware with the user, that is, improving the user interface.

On Windows-based systems, conventionally each **ASCOM** driver would provide its interface. It is possible to access it from the computer it is running on, with the evident complications in a distributed system like the one we are proposing. On the other side, on Linux-based Systems, the client itself would provide its interface. We can access it from the machine the client is running on. The user thus needs to install the client software and dependencies on its machine to access the interface of each peripheral.

To make every possible peripheral of the observatory genuinely network-friendly we propose that each server proposes its web-based interface compatible with a list of simple requirements for improved usability listed next:

1. Every interface should potentially be opened from any web-enabled device, and equipped with a standard modern web browser without the need of adding plug-ins or add-ons (including at least Windows/OSX/Linux/Android/iOS systems)
2. Some server must be able to display the data and the description of some critical internal variables several times per second
3. Each web interface should give access to commands corresponding to most of the slice functions it exports. Each server must also be able to give access in its interface commands to the **SLICE** function of its children clients whenever it could be useful
4. Interfaces must be easily customizable, of high-end quality and user-friendly

In the first approach to a browser implementation, frameworks like Tornado or Django seemed to be the best option.
However, first tests showed that its standard configuration using post requests becomes too slow to smoothly display the data evolution in real time. As a result we finally based our user interface on a simple WebSocket handler class so each **ZeroC-ICE** server should provide the following:

- A webpage running on an Apache web-server and a dedicated port (requirement #1)
- A Javascript applet will provide a websocket linked with the **ZeroC-ICE** (requirement #2)
- A Javascript applet will provide a list of function which can be called from the server for visualization updates (requirement #3)
- The same Javascript should provide a list of functions which can be called from HTML5 buttons, which will transcript in a visual way the functions of the slice interface and will communicate to the server accordingly (requirement #3)
- Finally, the webpage should be featured using HTML5 and CSS3 (requirement #4)

From a practical point of view, JavaScript functions handle the visualized data on the webpage of the peripheral. These functions along with their respective parameters are called from the Python server's WebSocket to each connected client browser. The advantage of using the WebSocket is its speed since this type of functions only refreshes numerical or text values on the pages, so they can be executed more than ten times a second, transforming a webpage into a real-time display for the user. Further, the optimal way to send a list of parameters from the Python server to the WebSocket is using a single parameter sent to the refresh function which is a JavaScript Object Notation (JSON) formatted string. This string can be easily created from the Python side by concatenating dictionaries or sets of parameters using the Python JSON library. On the web side, it can be easily parsed using the JSON.parse( ) Javascript function or created using the JSON concatenator function JSON.stringify( ).

As the main conclusion of this browser subsection, an embedded web server using the WebSocket protocol for each of the peripherals successfully allows the user to have a shared, user-friendly and recognizable User Interface regardless the operating system he is using, and without any need to install additional software, add-ons or plugins. Any recent browser would be able to give access to the complete system in real time, so the system adds the simplicity of technical implementation to a speedy response which enables data display in real time.

4.3.5 Discussion

Within this complete section, we have successfully defined and implemented an object-oriented software architecture which can be used to interact with any observatory peripheral. This software allows programming servers and clients in a variety of programming languages at the convenience of the programmer. It is even possible to select the optimal programming language depending on the performance of the hardware or the performance of the algorithmics. This becomes an advantageous approach for the integration of the servers in a community, as clients become independent from the hardware they are used for, as the programmer sees peripherals as black box objects from the client side, so that they will have a similar behavior from the server side point of view whatever is the hardware connected.

One of the main advantages of the proposed architecture is that the deployment of new systems can quickly be done using automation scripts and modern programming tools such as GitHub, allowing a highly dynamic development inside large software teams, keeping updates controlled and making the incoming tools being deployed to be transparent and user-friendly.

Beyond this modularity and transparency, one of the primary goals of this work was proposing highly accessible visualization methods of the indicators of the performance of the system. We successfully managed to define a method which would not need any installation of third party tools or plug-ins for the user keeping the ability to interact with every peripheral of the observatory in real time, equivalently as if it was on a local GUI.

Additionally, our method also allows a total abstraction from the network layer so the use of a VPN connection can even allow opening peripheral interface over a Wide Area Network (WAN) from any smartphone/tablet computer using minimal bandwidth. This is a very desirable feature for robotic and automated control of remote observation stations.

4.4 The SAPACAN Hexapod system

In the field of telescope optics, accurately positioning mirrors or detectors at the focal point of the primary mirror of the telescope (the main collector of light) is often limited by the mechanical strength of the support structure. In order to be stable, the telescope mount needs to be rigid, and thus expensive. This is specially true considering the telescope needs to precisely point at celestial objects along many hours in long exposures, tracking them as they move across the sky while avoiding flexures which defocus the image.

Telescope optics are amazingly simple, but need an extremely precise alignment and positioning of the focus of a large mirror (the primary mirror, which defines telescope class and performance through its light gathering capability). The primary mirror collects all the light coming from the object of interest and concentrates it on a distant secondary mirror, or, in the case of a prime focus configuration, directly onto a solid-state detector (e.g. a high sensitivity imaging sensor, usually a CCD -Charge-Coupled Device-) which collects the desired images. Fig.4.4.1 represents a lateral view of a a prime focus telescope with the relative positioning of the main elements such as the primary mirror and the detector. Telescopes are typically eagering for more light to improve the SNR (Signal to Noise Ratio) of distant or faint objects, and thus long exposure times with the telescope tracking the object of interest across one (or several) full nights are usual in fields such as exoplanet detection.

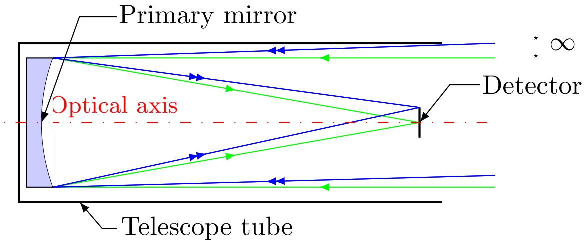


Figure 4.4.1: Lateral view of a prime focus telescope. Chief (green) and marginal (blue) rays coming from stars at infinity are represented.

However, keeping that positioning accuracy of the focus of the primary mirror while the telescope tracks a field in the moving night sky is a mechanically demanding problem, as flexures and misalignments happen while the telescope moves [115][116]. This is usually solved by spending a large amount of money in a rigid, non-flexible mechanical mount, or by using hexapod systems which reposition the secondary mirror or detector to meet the focus of the primary mirror. Such arrangements are thought to relax the rigidity requirements of the mounts of the telescopes. The most popular hexapod system in telescope optics used with this purpose is the Stewart platform [41], which has been described and analysed extensively. [117][118][46]. However, such approach is usually limited to telescopes with primary mirrors over 2m[116]. For smaller telescopes the relative cost of installing a Stewart platform behind the secondary mirror becomes too large to be considered when compared to the total cost of the project. Additionally, the increase of weight and size at the primary focal plane due to such device implies the use of heavier and stronger mechanics. As a result, in small class telescopes only focus is usually controlled, while the other four possible collimation positioning axes are only manually adjustable with tilt screws, which become fixed once the observation starts.

Within this Chapter we present a new, simple, 5-DOF (Degrees Of Freedom) parallel mechanism baassed on a cost-effective and simple mechatronic manipullator [119][120] which works in open loop conditions and may be used for positioning precisely secondary mirrors or detectors in multiple mirror systems [121] on the main axis of a telescope at the desired focal point. The proposed approach overcomes the cost limitations which prevent the use of hexapod systems such as the Stewart platform in small telescopes. The proposed mechanism is a 3T2R (3 Translations, 2 Rotations) device intended to position and orient a plane in space which works under open loop conditions. The structure provides the 5 degrees of freedom required to position the hardware with an active adjustment of tip, tilt, defocus, and lateral shift along two directions, while leaving a minimal footprint in the telescope light path.

The device is proposed as a low cost, feasible alternative to the hexapod systems widely used in large telescope active optics systems [48], but suited to its use in smaller telescope classes. Next Section briefly discusses the requirements for the application of the system to a prime focus telescope, and the description of the mechanical device to be set up and its mechanical restrictions. Section III will describe the main equations used to calculate the position of the detector out of the positions of the actuators of the device, defining the static positioning algorithm. Section IV describes the strategy used to bring the system from an initial position to a desired one, while Sections V and VI are devoted to the analysis of the nonlinear resolution, accuracy and range of the system. Before ending with the conclusions, results on sky obtained using the constructed prototype adapted to a 50cm diameter telescope will be shown, showing the capability of the system to properly compensate for focus, lateral displacement, and tilt of the detector plane using a quite simple and low cost mechatronic setup.

4.4.1 Description of the proposed system

The solution will be designed and built for a prime focus telescope, meaning that a high resolution CCD camera needs to be precisely positioned at the focal point of the primary mirror of the telescope, coincident with the image plane. Repositioning of a secondary mirror which redirected the light towards a detector would be completely equivalent. Besides, it is stressed that although the system has been originally built as a solution to a problem in telescope optics, it can be used as part of any active optical system where repositioning of optical components at low frequencies is required, covering several applications in large-scale active optics.

This definition of the problem to be solved drives us to the subsequent mechanical constraints:

1. The device to be used for positioning is partially obscuring the optical path of the telescope, thus preventing a portion of valuable light to reach the primary mirror. The system thus should be kept as simple as possible, avoiding the presence of mechanical components in the optical path or reducing its footprint to the minimum.

2. Assuming Z axis to be the optical axis of the telescope, the maximum acceptable error for lateral shift (XY plane) and defocus (Z axis) is set to 20μm. For the case of the lateral shift, this corresponds to the typical size of a pixel in a CCD detector. For the acceptable focus error, we considered the amount of defocus needed to spread a PSF over twice its optimal size in pixels using a 50cm primary mirror at f/5 (a 1 to 5 ratio of mirror diameter to focal length), resulting in the mentioned 20μm value.

3. The maximum acceptable error of positioning for compensating tilts along X and Y axes in the telescope primary focus is set to 1 arcmin. This constraint is obtained by considering the tilt angle of a CCD which generates a visible defocus of 20μm (again, one pixel) from the optical axis in the previously defined optical configuration.

4. As far as tracking is a slow, steady pace process, we only want to control changes in position at low frequencies, in order to compensate movement and misalignment of the primary focus due to mechanical flexures. Therefore, the maximum positioning speed will be set to 0.1Hz. Thus, atmospheric turbulence compensation or comparable adaptive optics approaches are explicitly disregarded.

The concept of the device [121] contains three arms hanged by three attachment points, which define the links to the telescope body of the positioning device. Each of these three attachment points is linked to a 2D (two-dimensional) positioning mechanism situated outside of the optical path of the telescope in a 120° even pattern, as shown in Fig.4.4.2. We propose to use three independent 2D actuators so each of them can evolve on a different 2D plane normal to the XY plane. Such 2D movement of each actuator is achieved using two different stepper motors, which perform in a coordinated but independent manner. In order to respect constraints 2) and 3), we will use a scissor design for the 2D positioning mechanism, in which each motor drives an independent worm screw in order to control the scissor angle with a large reduction factor, as shown in Fig.4.4.3.

Fig.4.4.2 and Fig.4.4.3 also detail the position of the important points used in the calculations of the positioning algorithms. So, P_1, P_2 and P_3 are the contact points of each arm with the central ring; O_1, O_2, and O_3 are the points of fixation of the scissor on the telescope body, while S_1, S_2 and S_3 are the attachment points of each flexible lame on its corresponding 2D actuator. We also define the points B_i, C_i and I_i, J_i whose relative distances define angles between the lower scissor and the telescope tube, and the relative angle between the lower scissor arm and the top scissor arm. Each motor can perform 200 individual steps per turn, and each actuates on a worm screw of 2mm per turn, so the minimum possible displacement along the distances I_iJ_i and B_iC_i is 0.01mm. Three flexible lames make the union between the central part where the detector will be fixed and the three actuators. The lames were obtained using stainless steel sheets of thickness 0.1mm. The length of the sheets was cut to link the central ring supporting the detector with each of the arms of the device placed at its middle range position. Width of such lames was chosen to be strong enough to support the weight of the detector and associated mechanics without flexures. Fig.4.4.3 shows the detail of all the components of one of the three independent actuators involved. The system is thus simple, efficient, suitable for small telescopes and very cost-effective.

The main difference between this device and other previously defined 5 or 6-DOF parallel systems previously proposed [122][123] is its simplicity and cost-effectiveness, plus the presence of the flexible lames, which allow the system to be driven to released (subtense) positions before moving, so each motor can be moved independently from the others over small distances, and subsequently can be run in open loop as if the lames were flexible strings [124]. Obviously, the performance of the system regarding the attainable positioning range, resolution and accuracy will not match that of a Stewart platform, although it will be enough to achieve the mentioned specifications in a small size telescope, as will be discussed later along Sections V and VI of this Chapter.

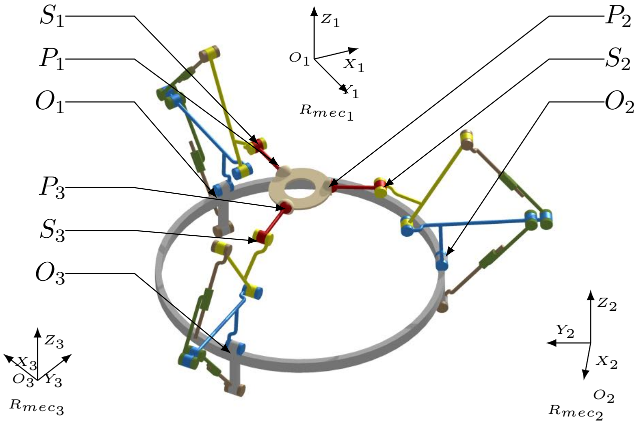


Figure 4.4.2: Kinematic of the device, including indication of the respective local reference frames of each scissor.

4.4.2 Static positioning algorithm

The static positioning algorithm consists in the computation of the proper absolute position of each motor, in units of number of steps from their respective home position. First we have to compute the positions of P_1, P_2 and P_3 in their respective reference frames R_{mec1}, R_{mec2} and R_{mec3} attached to O_1, O_2 and O_3. The positions P_i and the reference frame R_{meci} are defined in Fig.4.4.2 and Fig.4.4.3, in the latter case detailing the reference points on the left and the related mechanical components involved on the right. These coordinates are expressed as a function of the 6 parameters required to reposition the detector, defining the position of the primary focus (t_z), the lateral shift (t_x, t_y), the two tilt parameters (α_x, β_y) and the radius R_P, which is defined as the radius of the circle passing through points P_1, P_2 and P_3.

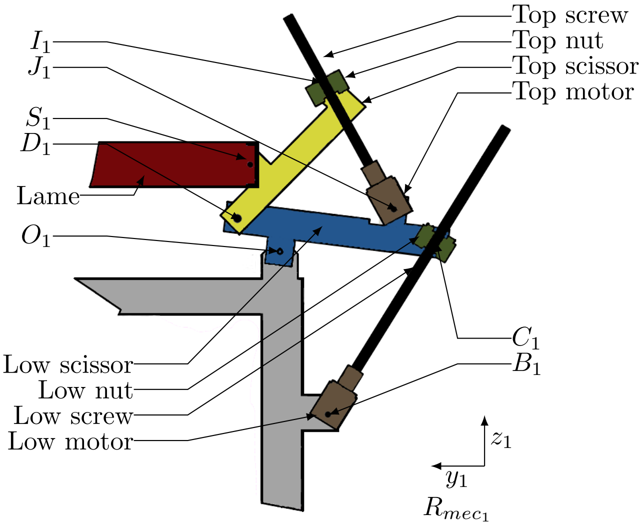


Figure 4.4.3: Lateral view of the device with description of its mechanical components with all relevant points used in the calculations detailed. Only Arm 1 is presented, Arms 2 and 3 are fully equivalent.

The relationship of the target coordinates of the hexapod and the parameters required to reposition the detector is established in equations (4.4.1) to (4.4.3) using trigometric relationships.

$$P_1^{Rmec_1} = \begin{pmatrix} -\sin\left(\beta_y\right)\sin\left(\alpha_x\right)R_P + t_x \\ \cos\left(\alpha_x\right)R_P + t_y \\ \cos\left(\beta_y\right)\sin\left(\alpha_x\right)R_P + t_z \end{pmatrix} \tag{4.4.1}$$

$$P_2^{Rmec_2} = \begin{pmatrix} \frac{\cos(\beta_y)R_P\sqrt{3}}{4} - \frac{\sin(\beta_y)\sin(\alpha_x)R_P}{4} - \frac{t_x}{2} + \\ \frac{\sqrt{3}(-1/2\ \cos(\alpha_x)R_P + t_y)}{2} \\ -\frac{\sqrt{3}}{2}\left(\frac{\cos(\beta_y)R_P\sqrt{3}}{2} + \frac{\sin(\beta_y)\sin(\alpha_x)R_P}{2} + t_x\right) + \\ \frac{\cos(\alpha_x)R_P}{4} - \frac{t_y}{2} \\ -\frac{\sin(\beta_y)R_P\sqrt{3}}{2} - \frac{\cos(\beta_y)\sin(\alpha_x)R_P}{2} + t_z \end{pmatrix} \tag{4.4.2}$$

$$P_3^{Rmec_3} = \begin{pmatrix} \frac{\cos(\beta_y)R_P\sqrt{3}}{4} - \frac{\sin(\beta_y)\sin(\alpha_x)R_P}{4} - \frac{t_x}{2} - \\ \frac{\sqrt{3}}{2}\left(\frac{\cos(\alpha_x)R_P}{2} + t_y\right) \\ \frac{\sqrt{3}}{2}\left(\frac{\cos(\beta_y)R_P\sqrt{3}}{2} + \frac{\sin(\beta_y)\sin(\alpha_x)R_P}{2} + t_x\right) + \\ \frac{\cos(\alpha_x)R_P}{4} - \frac{t_y}{2} \\ \frac{\sin(\beta_y)R_P\sqrt{3}}{2} - \frac{\cos(\beta_y)\sin(\alpha_x)R_P}{2} + t_z \end{pmatrix} \tag{4.4.3}$$

Using these equations we can compute the position of points S_1, S_2 and S_3 in their respective coordinate reference systems R_{mec1}, R_{mec2} and R_{mec3}. Every YZ mechanical stage can move on the plane YZ of its corresponding reference frames. The setpoints S_1, S_2 and S_3 are each linked to P_1, P_2 and P_3 respectively by the three metallic flexible lames of equal length L. We define R_S as the radius of the circle passing through the points S_1,S_2 and S_3. We also introduce the tensioning offset parameter ϵ, which is a virtual value we substract from the previously defined radius R_P in order to be able to adjust the tensioning of the lames. The same offset value ϵ is applied to the three lames. As a result we obtain a new virtual radius value R', which we will call the tensioning radius, defined by $R' = R_P - \epsilon$. If $\epsilon > 0$ the algorithm will compute positions for the points S_1, S_2 and S_3 so that the respective distances S_iP_i will be smaller than the actual length L of the flexible lames, so the whole system will not be completely tensed in this configuration. If ϵ is null the system will be perfectly adjusted to its nominal position and the points S_i and P_i will be coplanar. It is important to notice than ϵ should not lead to a negative value since in this case the distances S_iP_i would become bigger than the actual length L. The lames are flexible but not elastic, so this is considered an unreachable position. Fig.4.4.4 schematizes the positions of O_1, S_1, and P_1 in a lateral cut of the first arm in order to identify the distances ϵ, L, R', and R when the system is not tensed. This figure also identifies the radius R_S, which will be used later in Section IV.

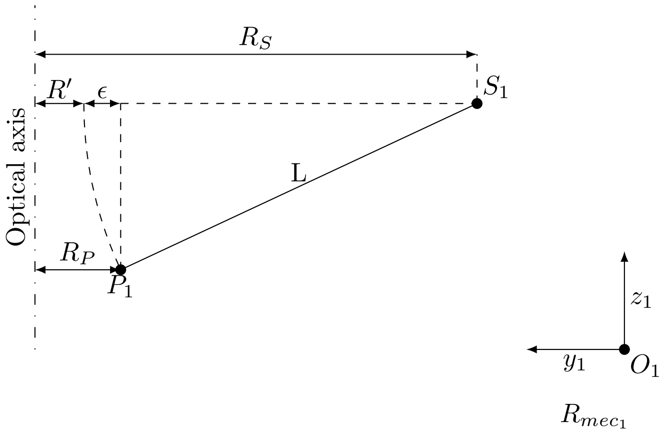


Figure 4.4.4: Lateral view of the system in an subtensed condition. Only one arm is represented with the definition of the distances described in the text.

The coordinates of S_1, S_2 and S_3 in the X axis of their corresponding reference frames are set to be null. This arrangement leads to the coordinates of the setpoints S_1, S_2 and S_3 to be expressed as in equations (4.4.4) to (4.4.9).

$$S_1^{Rmec_1} = \begin{pmatrix} 0 \\ \cos(\alpha_x) R' + t_y + a \\ \cos(\beta_y)\sin(\alpha_x) R' + t_z \end{pmatrix} \tag{4.4.4}$$

$$S_2^{Rmec_2} = \begin{pmatrix} 0 \\ -\frac{\sqrt{3}}{2}\left(-\frac{\cos(\beta_y)R'\sqrt{3}}{2}+\frac{\sin(\beta_y)\sin(\alpha_x)R'}{2}+t_x\right)+ \\ \frac{\cos(\alpha_x)R'}{4}-\frac{t_y}{2}+b \\ -\frac{\sin(\beta_y)R'\sqrt{3}}{2}-\frac{\cos(\beta_y)\sin(\alpha_x)R'}{4}+t_z \end{pmatrix} \tag{4.4.5}$$

$$S_3^{Rmec_3} = \begin{pmatrix} 0 \\ \frac{\sqrt{3}}{2}\left(\frac{\cos(\beta_y)R'\sqrt{3}}{2}+\frac{\sin(\beta_y)\sin(\alpha_x)R'}{2}+t_x\right) \\ +\frac{\cos(\alpha_x)R'}{4}-\frac{t_y}{2}+c \\ \frac{\sin(\beta_y)R'\sqrt{3}}{2}-\frac{\cos(\beta_y)\sin(\alpha_x)R'}{2}+t_z \end{pmatrix} \tag{4.4.6}$$

with:

$$a = \sqrt{L^2 - \left(-\sin(\beta_y)\sin(\alpha_x) R' + t_x\right)^2} \tag{4.4.7}$$

$$b = \left(L^2 - \left(\frac{\cos(\beta_y) R'\sqrt{3}}{4} - \frac{\sin(\beta_y)\sin(\alpha_x) R'}{4} \right.\right. \\ \left.\left. - \frac{t_x}{2} + \frac{\sqrt{3}}{2}\left(-\frac{\cos(\alpha_x) R'}{2} + t_y\right)\right)^2\right)^{\frac{1}{2}} \tag{4.4.8}$$

$$c = \left(L^2 - \left(-\frac{\cos(\beta_y) R'\sqrt{3}}{4} - \frac{\sin(\beta_y)\sin(\alpha_x) R'}{4} \right.\right. \\ \left.\left. - \frac{t_x}{2} - \frac{\sqrt{3}}{2}\left(-\frac{\cos(\alpha_x) R'}{2} + t_y\right)\right)^2\right)^{\frac{1}{2}} \tag{4.4.9}$$

Once the local coordinates of S_1 ,S_2 and S_3 are known, we can obtain the positions of the lengths I_1J_1, I_2J_2, I_3J_3, and B_1C_1, B_2C_2, B_3C_3, which fix the position of the mechanical axis on the corresponding worm, and are directly related to the distance to the home position of the corresponding stepper motor. Since the computation is equivalent for the three reference frames we will only express their equations for computing the distances I_iJ_i and B_iC_i as a function of the coordinates of S_i in the reference frame $RMec_i$. Obtaining this distances can be performed as follows. First we compute the position of the points D_i indicated in Fig.4.4.3 by solving (4.4.10), and then we solve the coordinates of I_i,J_i and C_i using (4.4.11), (4.4.12), and (4.4.13) respectively.

$$\begin{cases} \overrightarrow{O_iS_i} + \overrightarrow{S_iD_i} + \overrightarrow{D_iO_i} = \overrightarrow{0} \\ \overrightarrow{O_iD_i} \cdot \overrightarrow{y_i} > 0 \end{cases} \tag{4.4.10}$$

$$\begin{cases} \overrightarrow{O_iI_i} + \overrightarrow{I_iD_i} + \overrightarrow{D_iO_i} = \overrightarrow{0} \\ \overrightarrow{O_iI_i} \cdot \overrightarrow{y_i} < 0 \end{cases} \tag{4.4.11}$$

$$\begin{cases} \overrightarrow{O_iJ_i} + \overrightarrow{J_iD_i} + \overrightarrow{D_iO_i} = \overrightarrow{0} \\ \overrightarrow{O_iJ_i} \cdot \overrightarrow{y_i} < 0 \end{cases} \tag{4.4.12}$$

$$\begin{cases} \overrightarrow{O_iC_i} + \overrightarrow{C_iD_i} + \overrightarrow{D_iO_i} = \overrightarrow{0} \\ \overrightarrow{O_iC_i} \cdot \overrightarrow{y_i} < 0 \end{cases} \tag{4.4.13}$$

$$\begin{cases} \overrightarrow{O_iB_i} + \overrightarrow{B_iC_i} + \overrightarrow{C_iO_i} = \overrightarrow{0} \\ \overrightarrow{O_iB_i} \cdot \overrightarrow{y_i} < 0 \end{cases} \tag{4.4.14}$$

The inverse transformation permits to obtain the five parameters of the system and is performed in a similar way. At first, the positions of the points C_i can be obtained by solving (4.4.14) according to the coordinates of B_i and O_i and lengths B_iC_i, then the coordinates of D_i can be obtained by solving (4.4.13). Given this we can recover I_i according to (4.4.11), which leads us to obtain J_i from (4.4.12). Finally we can deduce the coordinates of S_i by solving (4.4.10).

Once the coordinates of the points S_i are known, the vector $\left[\alpha_x,\, \beta_y,\, t_z\right]$ is obtained by solving the coordinates of the center O_c of the circle passing through S_1, S_2, and S_3, α_x (x tilt) and β_y (y tilt) are obtained according to (4.4.15) and (4.4.16).

$$\alpha_x = \arctan\left(\frac{(\overrightarrow{S_1S_2} \wedge \overrightarrow{S_1S_3}) \cdot \overrightarrow{y_1}}{(\overrightarrow{S_1S_2} \wedge \overrightarrow{S_1S_3}) \cdot \overrightarrow{z_1}}\right) \tag{4.4.15}$$

$$\beta_y = \arctan\left(\frac{(\overrightarrow{S_1S_2} \wedge \overrightarrow{S_1S_3}) \cdot \overrightarrow{x_1}}{(\overrightarrow{S_1S_2} \wedge \overrightarrow{S_1S_3}) \cdot \overrightarrow{z_1}}\right) \tag{4.4.16}$$

4.4.3 Path computation

Each position of the device to be placed in our optical system is defined by the six degrees of freedom in which the system may be adjusted (defocus t_z, shift along x t_x, shift along y t_y, tilt on x α_x, tilt along y β_y, and radius R'). Notice R', the distance as defined on Fig.4.4.4 is now considered a degree of freedom for path computation purposes, while in past equations its was assumed to equal R_P, that is, the lames were assumed to be in tensed condition. We will now consider each step in the path of the system as a sequence of these six parameters. In order to be moved from a position set A to a position set B, the device must be sequentially moved to a set of intermediate position sets between A and B. In this way we will avoid to overconstrain the motors using unreachable positions out of their range yielding to maintenance problems in the device.

A complete flowchart of the algorithm used to drive the system from a position A to a position B is shown in Fig.4.4.5. A path is a list of consecutive positions which will be reached by the system. Each position is defined by its set of six parameters presented previously and an index which is an integer value representing its position inside the path. The fist position will be assigned the index $Index_{start}$ and the last postion will have the index $Index_{end}$.

Each path can then be split into 3 sub-paths from position A to position B. First the system will be driven from position A to A', where A' has the same five positional parameters of position A, but with $\epsilon > 0$ (see Fig. 4.4.4), meaning first the lames are set to an undertensed position. In practice, setting ϵ to a value of 3mm allows every movement to be performed without overconstraining the motors because of the potential existence of paths containing unreachable position. This way the tension applied to the three lames is slightly relaxed and motors can move without the risk of overload. Position A' is then driven to position B' which is described by the position of B, but with the same R' value of A'. In the last step, the system is driven from position B' to position B by progressively driving the value of ϵ to zero, so lames are again tensed.

As we can see in Fig.4.4.5 we use a recursive dichotomy algorithm [125] to find the intermediate positions of the path between two given positions. The algorithms developed permit to generate a path driving the system from a position N_m (characterized by the abovementioned six parameters) to a position N_n following recursively a path $\{N_1, [...], N_m, N_n[...], N_k\}$. m and n are respectively the indexes of the start and end positions of the path chain to be generated. The algorithm first calculates the maximum distance d_{max} to be covered by every motor in a given number of steps to go from N_m to N_n. When the maximum number of steps to be done by a motor is smaller than the maximum distance allowed for a single iteration, it means the system reached its final position and the algorithm ends returning the generated path. If the maximum number of steps to be done by a a motor is higher than this distance, a 6 axis position N_j between N_m and N_n is interpolated linearly and added to the path in between these two positions. The algorithm is then called recursively twice with this update path, using as check points the pairs (N_m, N_j) and (N_j, N_n).

Due to the loose constraint in terms of speed we specified for the system, it is important to see that we do not include dynamic analysis in the scope of this document. Positioning corrections will be done when the detector is not collecting light between observations. As a result the movements can be considered as a simple sequence of semi-static positions. Higher frequency effects susceptible of degrading the image quality, such as windbursts, atmospheric turbulence or drive dynamics will be corrected, if necessary, by separate subsystems.

4.4.4 Resolution analysis

Next, we want to analyse the resolution of the system, understood as the minimal movement which can be done along each axis. Due to its simplicity, the device presents a strongly non-linear behaviour, and the minimal possible movement depends on the position of the system along the five axes which define the degrees of freedom of the movement $(t_z, t_x, t_y, \alpha_x, \beta_y)$ that is, focus, x shift, y shift, x tilt and y tilt.

Let's call $\mathcal{R}_{a_n}$ the resolution of axis a_n at a position $a = [a_1...a_5]$ and $[m_1...m_6]$ the corresponding position of the six motors of the system at one defined position a. The resolution of the system over one of the five axes is then defined as the movement generated on this axis when we move every motor by their minimal amount, which is considered one step in our case.

In order to make the analysis fit with the extension of this paper, we will present in detail the method to validate the resolution available in the positioning of the primary focus, against each of the other four terms. An equivalent methodology may be applied to the five axes against the remaining four. We thus will display four graphs, where we numerically plot the results of the change in positioning of the primary focus when the other four axes are not changed. As mentioned in Section 2, the minimum movement of the mechanism is obtained by a single step of a simple stepper motor of 200steps per turn, coupled to a low cost standard worm screw of 2mm per turn, yielding a minimum movement of 0.01mm, corresponding to the advance of one step of the motor. This value is a compromise between a good enough resolution in positioning and a fast enough displacement (a smaller step yields better resolution, but becomes proportionally slower as it needs a larger number of turns to achieve the same displacement). The length of the worm screws was chosen in order to give a suitable range of movement of the detector according to its intended use. Lengths and sizes of the scissors were selected so one step of each motor resulted in a focus change smaller than 20μm, according to the specifications described in Section 2. In addition, as even if the mechanism may be seen as a parallel system the presence of the flexible lames and the dychotomic algorithm just presented allows to drive each motor independently of the others, as in

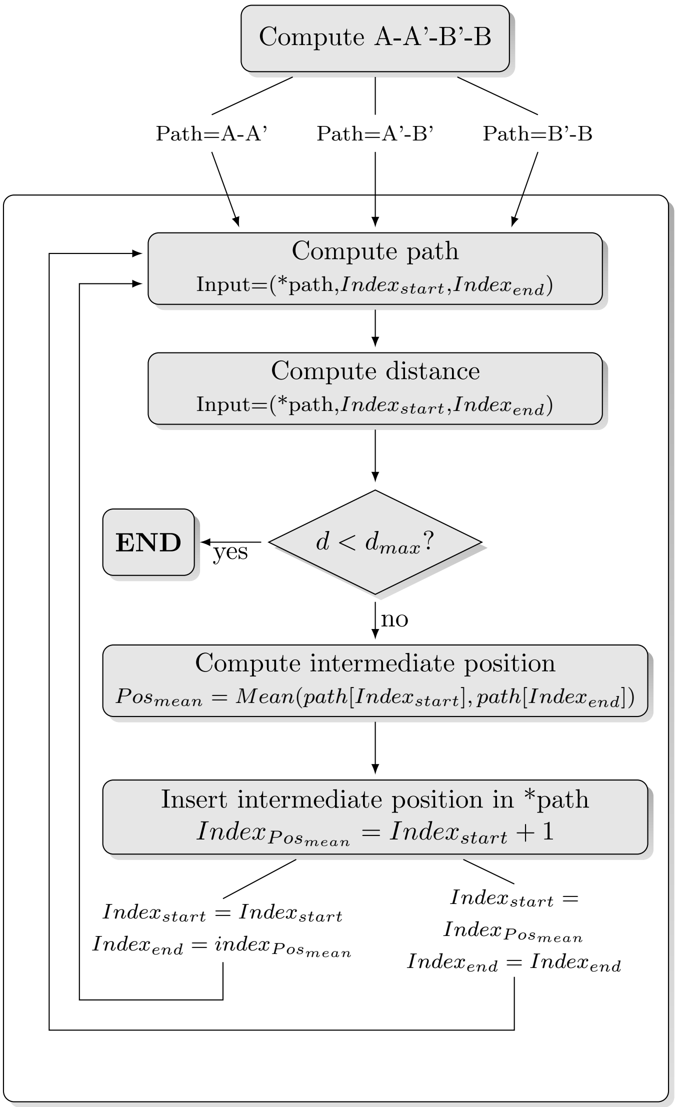

Compute A-A'-B'-B
Path=A-A'
Path=A'-B'
Path=B'-B
Compute path
Input=(*path,$Index_{start}$,$Index_{end}$)
Compute distance
Input=(*path,$Index_{start}$,$Index_{end}$)
END
yes
$d < d_{max}$?
no
Compute intermediate position
$Pos_{mean} = Mean(path[Index_{start}], path[Index_{end}])$
Insert intermediate position in *path
$Index_{Pos_{mean}} = Index_{start} + 1$
$Index_{start} = Index_{start}$
$Index_{end} = index_{Pos_{mean}}$
$Index_{start} = Index_{Pos_{mean}}$
$Index_{end} = Index_{end}$

Figure 4.4.5: Path computation flowchart. The algorithm can be assimilated to a recursive dichotomy. The output is the path made of a list of positions.

a serial system. This has the additional advantage that equations describing restrictions between axes are not required simplifying notably the mechanical behavior of the system.

Thus, for each position of the other axes we will find a change in focus, according to the inverse of the transformations described in equations (4.4.1) to (4.4.16), which may be plotted. Inverse equations can be analytically obtained in a straightforward manner using the same approach used for the direct equations. In such a graph, non-plotted points will be regions where the device cannot reach due to mechanical constraints, so the graphs will also depict the range of actuation of the device on the displayed axis. Fig.4.4.6 presents the resolution in focus displacement and the range of allowed values of the primary focus against the range of X or Y lateral shifts, when X and Y tilts are null (meaning they correspond to a geometrical focus plane perfectly normal to Z axis). In the graphs of Fig.4.4.7 we display the resolution of the displacement and range of positioning of the primary focus against the range of X or Y tilt when X and Y lateral shifts are null. As an example, we can read on Fig.(4.4.6a) that at the position where focus=$+5mm$, $x$shift= $+30mm$, $y$shift= $0mm$, $x$tilt= 0 and $y$tilt= 0 is not coloured, meaning it is unreachable, while on Fig.(4.4.6b) we can read that at the position where focus=$+5mm$, $x$shift= $0mm$, $y$shift= $+20mm$, $x$tilt= 0 and $y$tilt= 0, the movement of one step of each motor will give a focus variation of approximately 2.2μm.

The maximum change in focus position at all positions is below 4μm, exceeding the specification presented for focus adjustment.

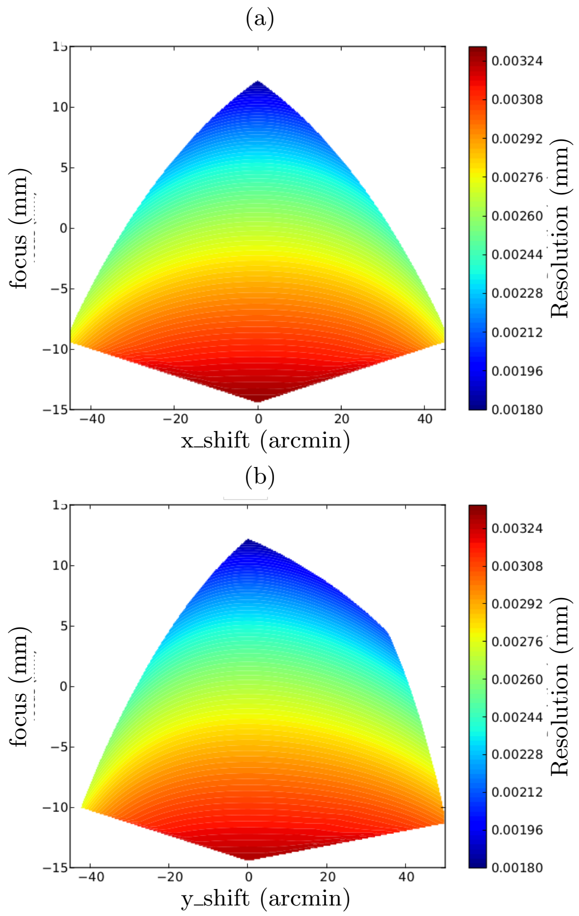


Figure 4.4.6: Focus range and resolution against a) X shift; and b) Y shift.

It may also be appreciated how the smallest possible displacement for a motor step (and thus, the best resolution) in focus positioning is obtained when the focus is set at Z close to its maximum value, while the largest one (the worse) is obtained when Z is close to its minimal position. This is in close relationship with the lengths B_iC_i in these respective positions. Since they are maximal when the focus position Z is at the lower regions, and the angle of the scissor at this position is close to 0, a single step of the motor will induce a bigger displacement in this position compared to the one performed when Z is close to its maximum, and then B_iC_i is minimal. In this case the angle of the scissor is bigger and the variation provided by a single step of the motor is also bigger, since it depends on the arctangent of the angle of the scissor. Thus, close to the maximum in Z we reach the best resolution, although also the minimum movement range, something coherent with the mechanical constraints of the proposed mechanical design. An identical situation, for equivalent reasons, happens when comparing focus with x_{tilt} and y_{tilt} (see Fig.4.4.7), yielding minimum (more precise) resolution values at the highest focus positions and much larger displacement values when the scissor is at the lower points of its trajectory.

4.4.5 Accuracy analysis

The evaluation of the accuracy with which the device is able to set a given position will be run over its five positioning axis, plus the accuracy of the tensioning radius R', defined in Fig.4.4.4. The inclusion of the analysis of this parameter is related to its relevance to image quality when compared to the effect of the remaining axes of the system. Such parameter is critical, as far as

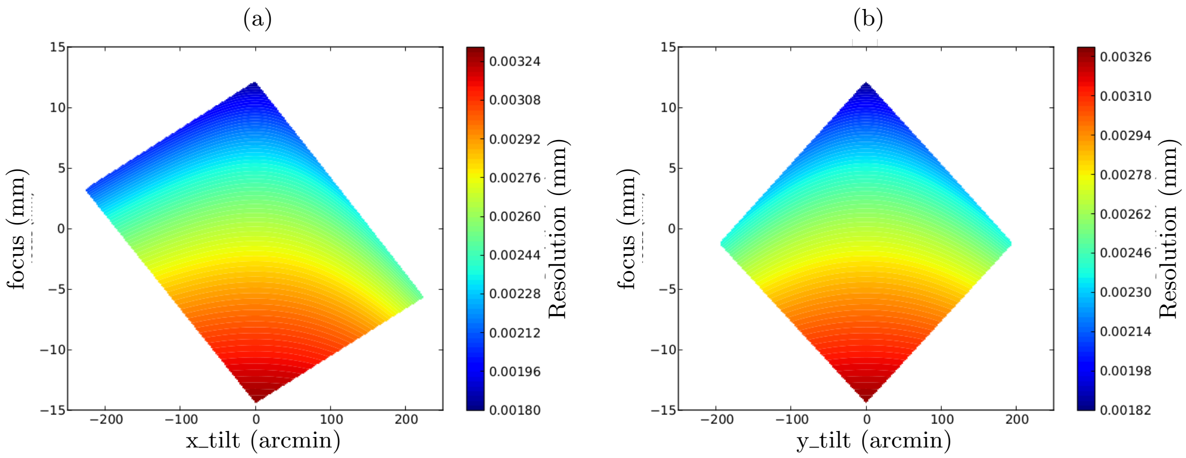


Figure 4.4.7: Focus range and resolution against a) X tilt; and b) Y tilt.

the accuracy in focus position has obvious effects in the overall image of the observed field, and the accuracy with which the tensioning radius is fixed leads to lames under or overtensed and thus to potential focus errors. A situation with overtensed lames will keep the system in a proper position but with the side effect of losing the exact homing position of the motors. A configuration with undertensed lames will make the system to slightly flex under the effect of gravity and as a result the points S_i and P_i will not be coplanar anymore, introducing a critical error between the theoretical and the actual focus position, and a subsequent degradation in image quality.

In order to perform the analysis of the accuracy of the system, we will numerically induce various sizes of mechanical homing errors to the distances B_iC_i and I_iJ_i through errors in the positions of the motors $[m_1..m_6]$. Homing errors and lost motor step counts are the main source of inaccuracies in the system, if mechanical elements are stable, and they are conditions prone to happen if overtensed lame conditions are achieved, which is a realistic situation. The effects of these errors across equally distributed sets of positions of x shift, y shift, x tilt, y tilt, and focus within the possible actuation range have been studied according to the inverse transformations previously described in equations 4.4.1) to (4.4.16. In particular, we will concentrate in the measurement of the absolute error in focus position and the absolute error on tensioning radius they induce, as errors on these two parameters have a dramatic effect on image quality. Such errors are measured in mm, and they are physically related to problems in homing or in positioning of the motors, as stated above.

The error in positioning along each of the six axes is computed using the inverse transformations so the distances B_iC_i and I_iJ_i may be computed in a perfect system and then in a system with positioning misadjustments due to lost motor counts or bad homing. This gives us the associated position of the points $S_i^{Rmec_i}$. Using these points we compute the 3D circle they define. Tilts over X and Y are computed from the 3D normal vector to the plane in which the circle is drawn, and lateral shifts and defocus are deduced from the position of the centre of the mentioned circle. We then define a total RMS homing error as the root mean square of the errors of the six individual motors. The accuracy of compensation of tilts, lateral shifts and defocus is obtained by respectively comparing the values obtained in the perfect system and in the mechanically misadjusted one. The accuracy of R' (that is, the accuracy in lame tensioning), will be defined by the difference of the value R_P and the theoretical radius R' obtained through $R' = R_S - L$ where R_S is the radius of the circle passing through the points S_1,S_2 and S_3.

Fig.4.4.8 shows that the absolute positioning errors in focusing and in tensioning radius are in the same order of magnitude. Positioning errors may be seen to be unacceptably large. However, since the positioning errors of the motors are mostly due to bad trigger distance estimation and/or repeatability of the magnetic proximity homing sensors, these aberrations can be considered as constant once the system has been homed. The range of variability of triggering of the sensors we used was measured using a distance measuring gauge and was estimated to ±200μm. It should be noted that relevant RMS motor positioning errors may result in small focus or tensioning errors by compensation of the errors introduced in multiple positioning axes.

As a consequence, it was needed to add a new step to the homing procedure in order to consider this effect, as an alternative to include antibacklash arrangements in the worms which unnecessarily raise the cost involved. The conventional homing procedure should now be followed adding a new step where the system goes to the first reference position to be used for tracking with a loose ring ($\epsilon > 0$), to gradually bring it to be overtensed applying smaller and smaller values of ϵ, even if ϵ gets to be negative. We then continue the process until being sure that each motor got overtensed and ran within its current limit at least once. Once this is done, we can be sure that the spider is perfectly tense and we can move to any set of tilt in x,y, lateral shifts and defocus. As far as errors in manufacturing and mounting are negligible compared to the precision of zeroing of the motors, we will focus the forthcoming accuracy analysis in the quantification of the errors induced by inaccurate homing.

Given the described additional step in the homing procedure is applied, Fig.4.4.9, quantifies the effect of homing errors in the focus and tensioning radius errors. Fig.4.4.9 represents in the vertical axis the absolute position of the center of the device (being 0 the exact focal point), while showing in colour scale the amount of focusing error and tensioning radius error introduced by a given homing error value. The value for the homing error is calculated as the RMS (root mean square) value of the vector composed by the respective homing errors of the six motors present in the system. Thus, Fig.4.4.9 shows the tensioning and focusing accuracies on a relative movement from the new origin defined by the homing procedure.

After a series of measurements with a distance gauge, it could be estimated that the initial distances B_iC_i and I_iJ_i are repeatable within less than $10\mu m$, but with an accuracy of $\pm 0.5mm$, with this value including potential homing errors.

We can observe in Fig.4.4.9 that the focusing and tensioning induced aberration by homing errors in this case stays below $10\mu m$ across a 5mm excursion around the focal point region, given we keep the homing error below $\pm 0.5mm$, which is perfectly feasible in practice. Please notice in Fig.**??** the vertical axis is the absolute position of the focus measured from the ideal focal point, while the horizontal axis describes the random homing errors. The focus and tensioning errors are now plotted as colormaps instead of being contained in the vertical axis of the plot as in Fig.4.4.8.

As an example, we can see that with a RMS homing error of $0.6mm$ at $+2mm$ of absolute focus, the error in focusing is $3.5\mu m$ while the error over R' is about $0.45\mu m$, which agrees well with the specifications proposed in section 4.4.1.

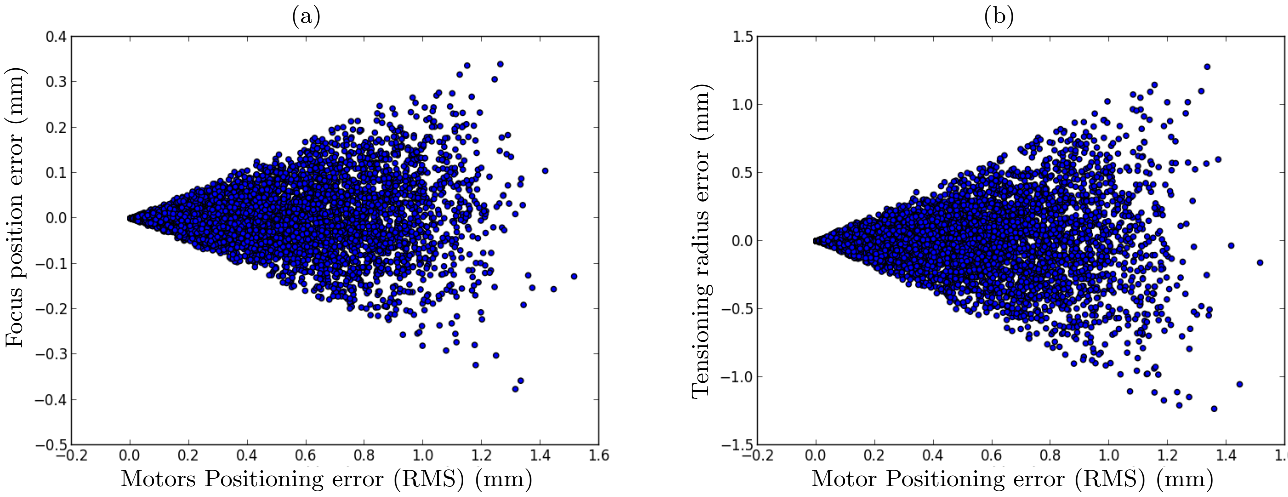


Figure 4.4.8: Accuracy of a) focusing; and b) tensioning radius prior to the homing procedure described in the text.

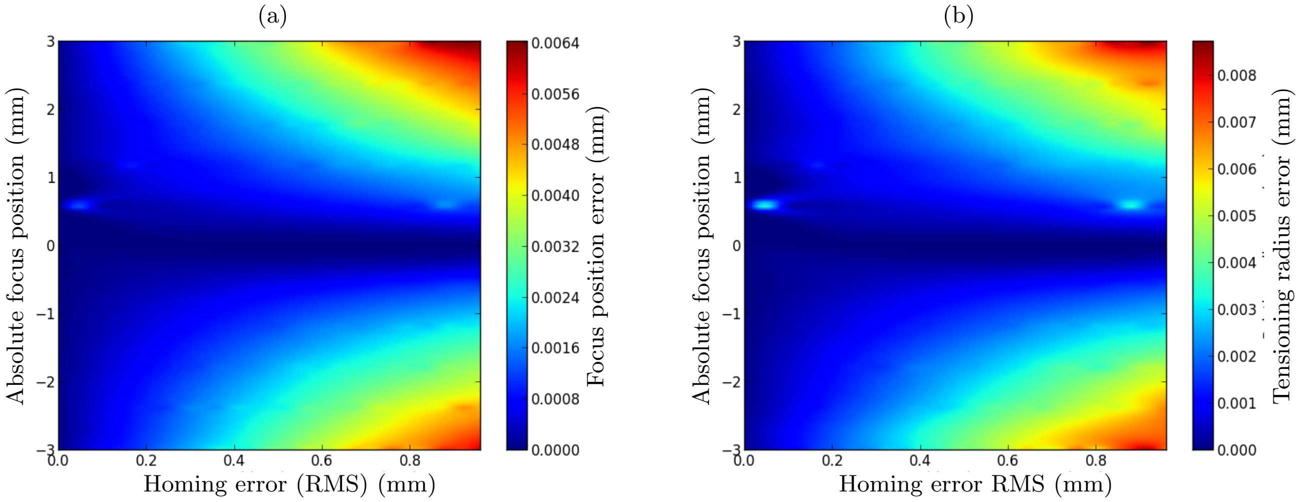


Figure 4.4.9: Accuracy of a) focus position; and b) tensioning radius for different homing errors once the homing procedure described in the text has been applied.

4.4.6 Results

A prototype of the system was adapted to a 50cm telescope using the usual process of design (Fig.4.4.10) and construction (Fig.4.4.11). The telescope mount was built using conventional commercial aluminium profiles, yielding a very light and economic shape but with less rigidity than a conventional telescope. In order to check the performance of the device on such a small telescope, we verified the system was performing as desired by observing series of open and globular star cluster fields, in order to obtain a sufficient star density all over the detector so image quality could be evaluated. We show as an example one of the results obtained in Fig.4.4.12 and Fig.4.4.13. Both images show a mosaic formed by an extraction of subimages of 300x300 pixels selected at the four corners of the detector and the central area of the image, so the image aberrations at the edges are properly highlighted. Fig.4.4.12 was taken with the hexapod device in its initial homing position. The stars imaged at the edge of the field of view show very relevant defocus and even vignetting may be appreciated in the left corner, due to misalignment of the optical axis. The size of the stars in all corners is also very variable, due to improper tilt of the detector. The corrections to be done were obtained by visually inspecting the evolution of defocus and coma of the stars across the field in order to obtain the tilt and lateral shift parameters, while the best focus position was obtained also by analysing the size of the stars. Fig.4.4.13 shows the same field after applying the proper corrections to the device, resulting in the following parameters in the axes: focus offset: $500\mu m$, $x$shift: $-1000\mu m$, $y$shift: $150\mu m$, $x$tilt: 58 arcmin, $y$tilt:-28arcmin. The images show now the appearance of unvignetted stars, and a comparable size of objects in all areas of the image. It is important to notice that the two images seems to aim at slightly different fields of stars, due to the fact that the observed point was slightly modified by adjusting the X_{tilt} and Y_{tilt} parameters of the device.

To finalize, we show in Tab.4.4.1 a comparison of the main features of the presented device compared to a standard commercial hexapod based on a Stewart platform of similar size and weight. It is important to notice that although the presented device cannot reach the features proposed by a commercial one in terms of range, repeatability or speed, the cost has been dramatically reduced, making it feasible for its use in the considered range of telescopes, while mechanically keeping the optical path clean as the actuators are left in the outer part of the telescope tube.

Additionally, the range of movements we need for our device is still much smaller in comparison to what the device offers since we only want to compensate for flexures due to the mechanical mount proposed. The property of having the actuators outside the optical path offers also the potential of using larger, even more demultiplicated hardware for higher resolution positioning at low cost of manufacture. The gain in resolution in this case, however, would be paid off with a slower system response and a smaller actuation range which would not be useful for the telescope application we

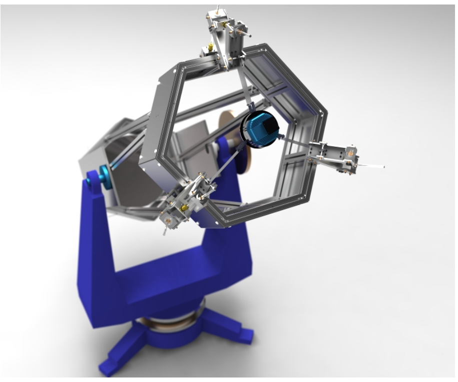


Figure 4.4.10: 3D render of the designed system.

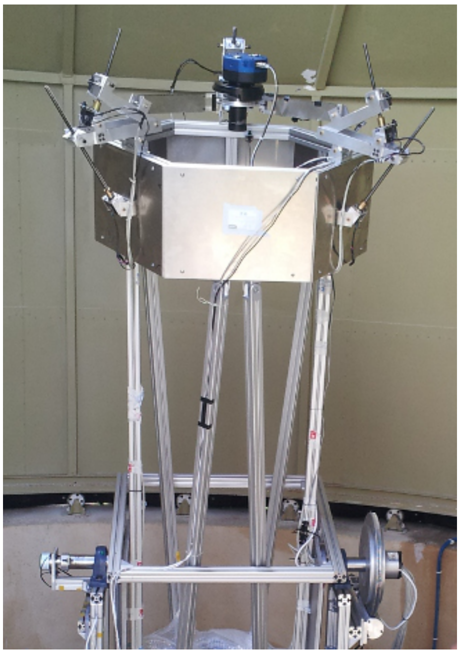


Figure 4.4.11: First prototype built.

have described, although possibly could be useful in other applications.

4.4.7 Discussion

We have proposed a low cost hardware system able to position efficiently an optical device over 5 axis of freedom. The system has been adapted to a small telescope showing the capability of adapting the focus point position along a tracking process, a procedure unavailable to this telescope class due to the large cost of the associated Stewart platforms. The system has been designed to use a minimal number of actuators and a mechanical arrangement which minimizes the footprint of the device in the optical path of the telescope, using a set of flexible lames between the actuators and the central ring containing the sensor. The proposed geometry removes the need of constraining all actuators together and enables the device to perform adequately in open-loop conditions. The description of the main equations and algorithms used in the control software has been presented, as the algorithmics for path computation between neighbouring points. A methodology for computing both the resolution and the accuracy of the positioning system proposed has been discussed, showing its agreement with the desired specifications but also the nonlinearity and limitations in

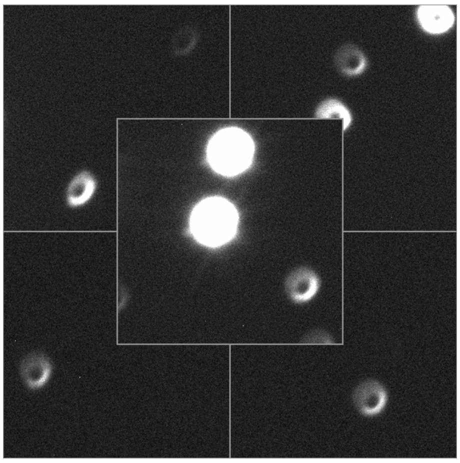


Figure 4.4.12: Field before correcting shift, tilt and focus.

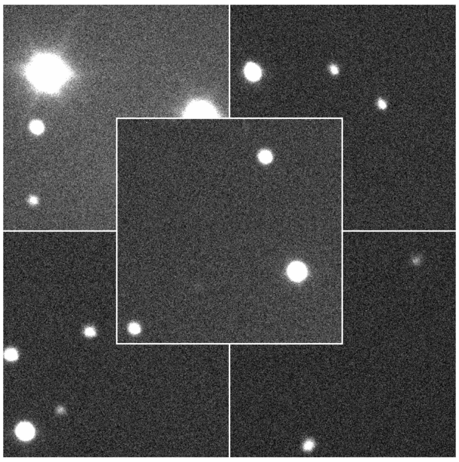


Figure 4.4.13: Field after correcting shift, tilt, and focus.

range imposed by the low cost approach used. The proposed prototype has been built and tested on sky, showing significant improvements in images when the position of the device is optimized.

As future work, we plan to integrate the one shot focusing algorithm described in next Section, which extracts aberration information directly from the acquired images in the CCD [126], so it can automatically measure and compensate the aberration values after each image acquisition so the detector is kept at in its optimal position in all moments.

Table 4.4.1: Comparison of the proposed device and a commercial hexapod based on a Stewart platform of comparable features.

	Our device	Hexapod
Supported weight	10Kg	10Kg
Device weight	10Kg	10Kg
Size	3 arms 200x300x50mm	⌀300mm x 200mm
Obstructs optical field	no	yes
Shift range	±10x10x15mm	±50x50x25mm
Angular range	$\pm 3^o$x3^ox0^o	$\pm 15^o$x15^ox30^o
Repeatability	$3-10\mu$m	$1-2\mu$m
Speed	<0.1mm/s	>20mm/s
Price	<1000 USD	>15000 USD

4.5 Controlling and measuring collimation from single images

Within this final Section of this Algorithmics chapter, we present a method for measuring focus aberrations on wide-field telescopes based on the analysis of the entropy of a single image. First, we calibrate the system using the evolution of the entropy as a function of the position in the field and the position of the focuser. This calibration gives us a model defining the tilt of the sensor and the field curvature. Then, using a single image at a given position of the focuser in which the mean defocus is unknown, we can compute the optimal focuser's position to minimize the focus aberration over the whole field. In practice, the system becomes a sensorless adaptive optics approach using images to compute the aberrations in them and to actuate on the telescope optics to compensate them actively.

In practice, one of the most basic features one often desires in telescope operations is to keep the system at or as near as possible to the position of the optical focus along an observation, or along an image acquisition in an extended exposure integration including tracking. Historically, this has been done by manually adjusting the focus such that the narrowest point spread function possible was achieved, based on the judgment of the eye on stellar images. Large modern telescopes rely on improved automated systems, such as those defined by, e.g., Mc Leod et al. [23] or Kuehn [127].

In this Section, we will present an algorithm to find the best focus position of a telescope delivering wide-field astronomical images in a non-iterative way. Debarre et al. [24] [128] [34] showed that given $N + 1$ measurements of some merit function of a system aberrated by N offsets applied on N different optical modes, it was possible to minimize the aberration of these N modes. In our case, we want to minimize a single optical aberration on a wide-field image which is the defocus, as it is the critical aberration in telescope optics and we want to avoid iterative image acquisitions to keep fast feedback on the system. To be able to find the best mean value of the defocus using a single image, we consider the field curvature as a radial function of the distance to the optical center and proceed in 4 steps:

- First, we calibrate the field curvature, sensor tilt, and the position of the optical center using a set of images which are affected by different amounts of known mean defocus.
- In step two, we determine a model function that allows us to associate an absolute defocus value with a set of values of a given image quality metric.
- The third step consists in measuring the value of the image quality metric chosen in step 2 in different parts of the image.
- Finally, a minimization algorithm will find the mean defocus value which best fits the data obtained in step 3 using the model obtained in step 2.

Steps 1 and 2 can be done a single time for the full instrument lifetime (as long as it is stable and the user does not intervene), while we will run steps 3 and 4 every time we want to compute the best focus position as determined by a single image. In this work, we use data of focus series taken with four telescopes of the HAT-South[1] project [129] to illustrate the procedure outlined above. The four HAT-South telescopes are based on the same optical design but may be affected by different alignments of the sensors relative to the corresponding optical axes. We show that the local entropy of a sub-image around the detected stars is the best image metric for our application, rather than other image metrics more usual in the definition of astronomical image quality, such as, for example, full-width at half maximum. Finally, we briefly discuss how stable is the proposed technique under variations of the model such as, e.g., variations in the sensor tilt which can arise

[1] The HATSouth hardware was acquired by NSF MRI NSF/AST-0723074, and is owned by Princeton University. The HATSouth network is operated by a collaboration consisting of Princeton University (PU), the Max Planck Institute for Astron-omy (MPIA), and the Australian National University (ANU). The station at Las Campanas Observatory (LCO) of the Carnegie Institution for Science, is operated by PU in conjunction with collaborators at the Pontificia Universidad Catolica de Chile (PUC), the station at the High Energy Spectroscopic Survey (HESS) site is operated in conjunction with MPIA, and the station at Siding Springs Observatory (SSO) is operated jointly with ANU.

due to instrumental flexure. We will analyze the four steps we have described in detail along the text.

4.5.1 Step 1: Calibration algorithm

The first step in the procedure is the calibration of the field curvature, sensor tilt, and the position of the optical center, using a set of images with known mean defocus. We tested the calibration algorithm we propose on a focus sequence of thirty images taken with the HS1.4 telescope of the Hat-South Las Campanas Observatory (LCO) station. We took every image with the focuser in a different known position. We thus know the absolute focus position in coding units of stepper motors, and we want to find the parameters $z, \vec{O}, \vec{m}, \vec{t}$ which we define as:

- Defocus offset z (offset from the optical focus position)
- The position of the center of the field$\vec{O} = (O_x, O_y)$
- The value of field curvature, which we can express as a quadratic function from the distance to the field center, defined by the scalars $\vec{m} = (m_1, m_2)$
- The detector tilt, which is defined by a linear function defined by the parameter $\vec{t} = (t_x, t_y)$

The computational procedure to obtain these parameters is described in detail next.

4.5.1.1 Single image analysis

For each of the images to be used in the calibration we measure a given number of metric values around every detected star. To perform star detection in our frames, we use the SExtractor [130] software, freely available on the internet at, e.g., *www.astrometric.net*. The metrics we extract for each star are

- Full width at half-maximum (FWHM).
- Stellar ellipticity.
- Entropy of a 10x10 pixel^2 box around the star.

We define entropy according to Equation 4.5.1, where P is the normalized binned histogram of the values in a 10x10 pixel^2 box defined around the star:

$$e = \sum_{i=0}^{len(P)} P_i \log(P_i) \tag{4.5.1}$$

We define the box's size so that the star at the maximum defocus image we use still fits inside the box:The measured values for each star are then smoothed using a 10x10 zone grid covering the full field of view. In practice, we take the available field of view and divide it in a mesh of 10x10 independent boxes. We present the organization of the boxes within the visible field of view and a typical density of detected stars in Fig.4.5.1

Every box of the mesh has an associated value of the metric at its center, which is the mean of all values of the sources present in the considered box. The value of the metric at an arbitrary position in the box is obtained by interrelating from the grid. An example of a calculated entropy map in a single imageis presented in Fig 4.5.2.

An extended analysis of several images affected by different values of mean defocus shows that images before focus tend to present lower entropy values close to the optical center rather than on the edges, while images taken behind the focus tend to present better results on the edges rather than on the center of the field. This asymmetry is due to the effect of field curvature, and as a result, a detailed analysis of *any* defocus image that can measure this asymmetry should in principle be sufficient to determine on which side of the focus we are and how far we are from the best focus position. Thus, the local behavior of entropy becomes a vital issue in the process as it potentially contains information to ascertain the position of the focus, and deserves further attention.

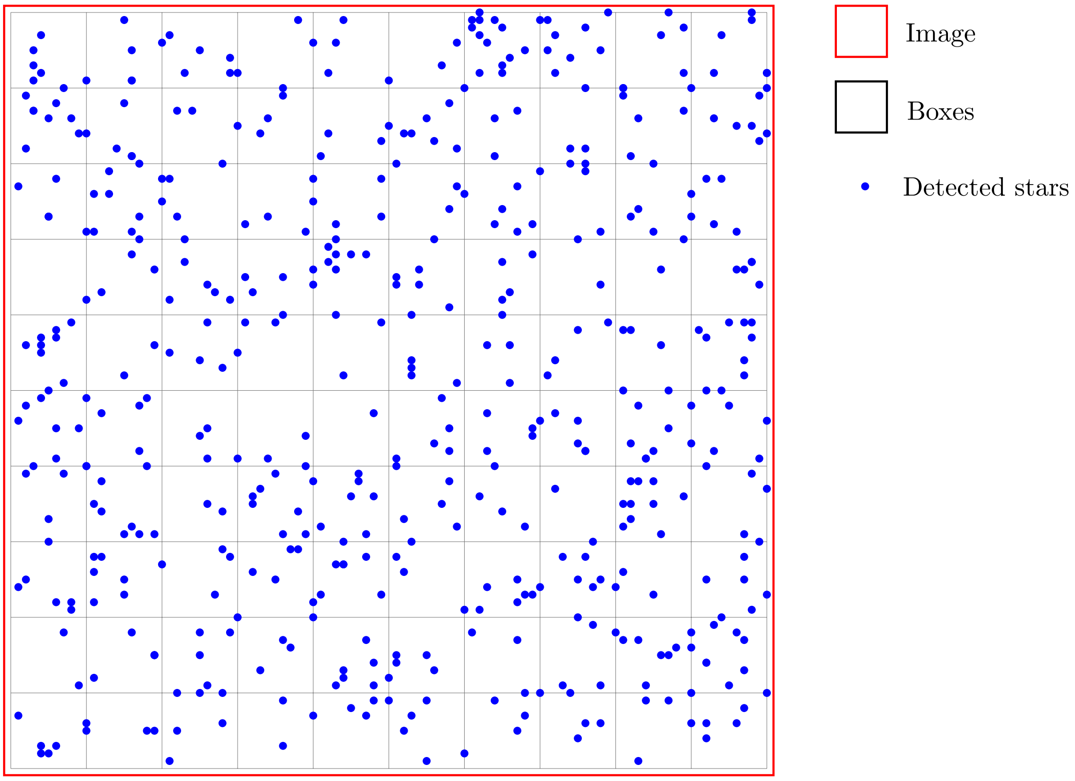


Figure 4.5.1: Organization of boxes in the image.

4.5.1.2 Local evolution of entropy

When we divide a given image in a 10×10 mesh map as we presented in Fig.4.5.1, we can probe the evolution of the mean entropy (or any other image metric) as the defocus is changed to evaluate which is its behavior. The graph shown in Fig 4.5.4 shows the evolution of entropy as a function of focus position for 10 zones in the HAT-South images series at different focus positions. Entropy consistently behaves as smoothly varying with focus change, and it obtains its minimum value at the best focus position for each zone considered. However, this best focus position is not the same for all the zones of the images. However, the behavior of the entropy as a smooth, derivable function of the position of the focuser is equivalent from zone to zone. Such differences in different zones are perfectly acceptable as they only reflect better compensation of local aberrations present in the image in different regions of the image. Fig. 4.5.3 shows the selection of boxes used to generate Fig 4.5.4. We deliberately decided to limit the number of boxes displayed in Fig 4.5.4 to clarify the visualization.

4.5.1.3 Optical instrumental model

Our next step will be the development of an instrumental model of the focusing of the telescope to perform focus compensation from the calculations of the metrics described above, including entropy.

To further characterize the variation of a given metric on a given zone we choose to fit Gaussian functions to the associated measurement values at each zone.

We then fit a Gaussian function of the form

$$f(x) \quad = \quad ae^{-\frac{(x-b)^2}{c}} + c \qquad (4.5.2)$$

to the variation of a given metric in each zone of the image. We then can extract from the fit the value of the Gaussian mean b (i.e., the decentering of the Gaussian of each metric), and plot

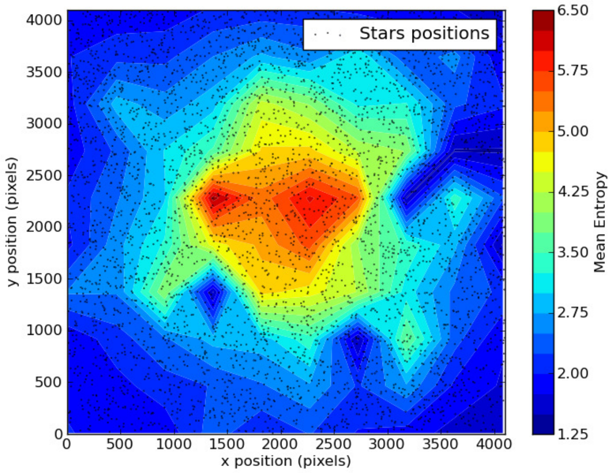


Figure 4.5.2: Entropy measured on one image.

its value over the entire field of view of the image. Since b corresponds to the decentering of the Gaussian, the fitted value of b for a given zone will give the absolute best focus position for this position in the field. We show the best fit value of b for entropy over the field obtained with our set of focus images for HS1.4 in Fig 4.5.5. It is now clear how the best focus position is not constant over the field. We can see the radial distribution nature of the focus and how the focal plane is curved.

The map obtained from this figure gives us the value of the best focus as a function of the position in the field (x, y) calculated out of the entropy value. This shows how it is possible to make a radial model of best focus as a function of the distance to the optical center. From this information we can estimate the parameters $\{z, \vec{O}, \vec{m}, \vec{t}\}$ defined previously. To achieve this, we use the map of b values obtained to fit the following model:

$$b \quad = \quad z + t_x (x - O_x) + t_y (y - O_y) + m_1 r_{(x,y)} + m_2 {r^2}_{(x,y)} \tag{4.5.3}$$

where:

$$r^2 \equiv (x - O_x)^2 + (y - O_y)^2. \tag{4.5.4}$$

The multidimensional model for HS1.4 and we show its residuals in Fig. 4.5.6, where we see that a reasonable quality fit is obtained enabling all parameters to be calculated, including the best focus position for the image (in this case, 264.99mm). We could, of course, obtain a better fit with a 3rd or higher order model, but this would have made the fitting and later evaluation processes slower and, as we will show in the following, a second order model proves to be accurate enough for our purposes.

Thus, we obtained a consistent model of the focus position of the telescope based on the analysis of the evolution of the entropy function. The model allows the determination of the best focus for the image plus the tilt, vertex position, and field curvature. Fig.4.5.5 shows a map of the fitted value b as a function of the position of the field. Since b represents the offset of the Gaussian fit of the evolution of the entropy when the focal plane moves across the focus, the value of b gives the focus position of the entropy minima, and thus the best focus for the corresponding measured zone of the field. It is now clear that the best focus is obtained at different positions of the focuser depending on which part of the focal plane we are watching. The evolution is radial, and we can also see a linear evolution from the bottom of the field to the top which is due to the focus tilting. The same work has been performed using the ellipticity and the measured Full Width at Half

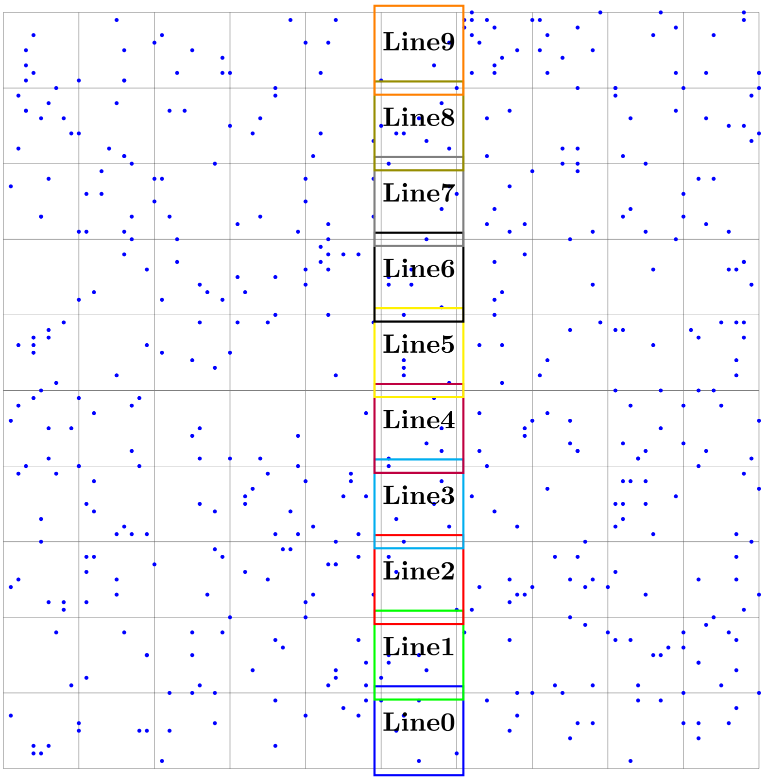


Figure 4.5.3: Selection of boxes of measurements displayed in Fig.4.5.4

Maximum (FWHM) of the stars, and we plot their respective evolution across focus in Fig 4.5.7 and 4.5.8. We can naturally assimilate the FWHM to a Gaussian, and its minima is related merley to the best focus position. However, it is possible to see in Fig 4.5.7 that the measurement quickly deviates from any possible Gaussian when we get away from the best focus because the shape of the stars is not gaussian anymore and the annular pupil starts to be visible.On the other side, we can still assimilate the evolution of ellipticity to a Gaussian was we get away from the best focus position, but the minima is not related to the real best focus.

As a consequence, we decided to use only the entropy as a measurement metric since it remains robust as we get far away from the focus and is strongly related to the evolution of the entropy when we are close from the focus. Which lets us conclude that the entropy is, in fact, the best metric of focus quality both for out-of-focus and in-focus images.

4.5.2 Step2: Model of the Behaviour of Entropy as a function of Absolute Defocus

4.5.2.1 Analysis of different image metrics

At this point, we have a model of our telescope which, out of its images, enables the characterization of field curvature, CCD tilt and position of the optical axis. This allows us to predict the best focus position b for each star on a given image out of its position in the field.

From now on we will introduce the term *Absolute Defocus*. While we define a *Focus position* the position, in steps, of focuser motor, the *Absolute Defocus* represents, for a given zone of the field, the difference in focuser 's steps, from the actual position to reach the best focus position for this zone.

We can then plot the evolution of any given metric measured on a per star basis as a function of its *Absolute Defocus*, (i.e., the value of its defocus relative to the best focus position at the position

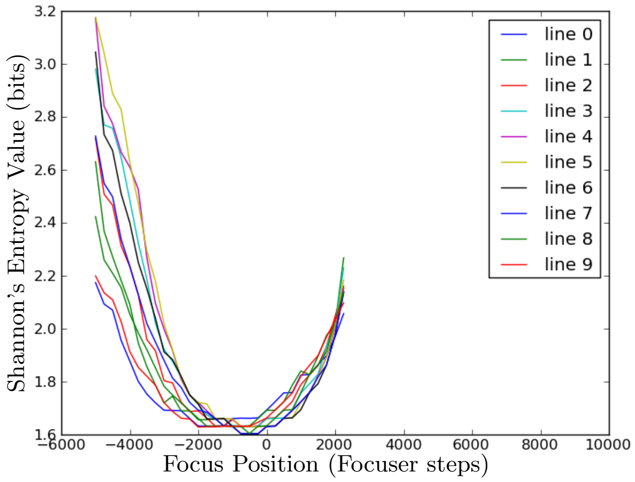


Figure 4.5.4: Evolution of the mean entropy on 10 zones of the field as a function of the focus position. The 10 selected zones are presented in Fig.4.5.3

of the source). The best metric will be the one which shows a nearly constant behavior across the whole field of view, as far as this will allow using this single function to model the effects of defocus on our instrument. We carried out this exercise for the three image metrics which we mentioned above, namely FWHM, ellipticity, and entropy.

We display the results in Fig 4.5.7, 4.5.8 and 4.5.9. It is clear from the figures that entropy is a significantly more stable metric at all positions of the field of view. Its behavior as a function of absolute defocus is more homogeneous across the field, making it the best metric for our purposes and enabling to predict the best focus position of the telescope out of single images. Just as a comparison, ellipticity is strongly unstable as a clear minima region where to set the focal point is hard to find.

4.5.2.2 Entropy behavioral model

Once entropy has been fixed as the optimal image metrics for our purposes, it will now be straightforward to define a behavioral model which will associate an entropy value to a known absolute defocus. We based this model on a functional fitting of the entropy measurements shown in Fig. 4.5.9. When testing quadratic, cubic and Gaussian functions as basis functions for the fit the results showed an insufficient quality fit to the set of points, so we turned into non-parametric functions to model entropy. Our first trial was a 1-dimensional smoothed spline. We can see this model overlaid on the experimental data points in Fig. 4.5.10.

If we plot the evolution of entropy in a 3D plot including absolute defocus and distance to the center of the image, it may be seen how the behavior of the focal point has a particular dependence on the distance to the optical axis (see Fig. 4.5.11). Since we are plotting the entropy as a function of the absolute defocus, it appears the distance to the optical center is the only parameter which still has an influence on the shape of the evolution of entropy when we are getting away from the absolute focus. This is because the distance to center does have an effect on the coma and vignetting which may not be constant nor symmetric when we get away from the best focus on one side or another. However, since we measure as a function of the absolute defocus, which is the distance from the best focus position in this zone of the image, the tilt parameters do not have any influence since they are included into the absolute defocus.

To further strengthen the model and to make it more accurate, we also tried the fit of a 2-dimensional model based on a smoothed B-spline which simultaneously fits entropy as a function of defocus and of the distance to the center. We present the fit of the 2D B-spline model and the

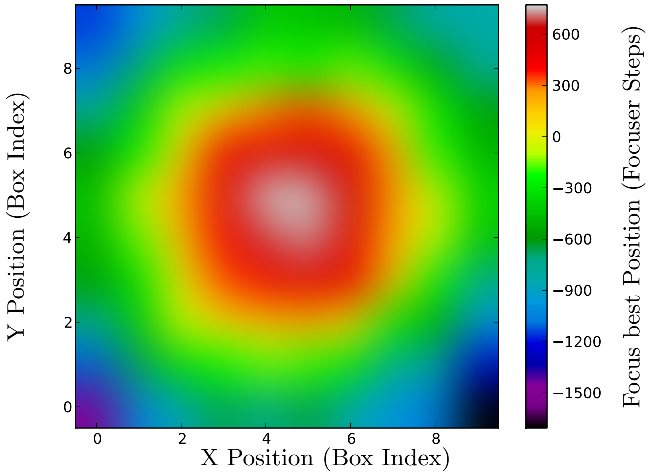


Figure 4.5.5: b value (mean of the Gaussian fit of the entropy metric) over the field of HS1.4

associated data in Fig. 4.5.11, enabling to see the dependence of entropy with the distance to the center and how the B-spline enables a proper fit of the data. It is possible to see the residuals obtained after the fit for both the 1D and the 2D smoothed spline fits in Fig. 4.5.12, where, clearly, the 2D spline shows a better behavior.

4.5.3 Step3: Determination of the position of the focus

Once we constructed the optical model and the behavioral model as described in the previous Sections, we are now able to associate the modeled entropy in any position of the field with an absolute defocus value, so optimal focusing is consistently attained based on the analysis of the images obtained. We can now use this information to determine the mean defocus of an image for which this quantity is unknown. After acquiring such image, we first find the stars present in it and compute their associated entropy in a box of 10x10 pixels[2] around each star, generating a vector of measured entropies for the image, $\vec{e} = (e_1, \ldots, e_n)$. Using the model of the instrument which describes the evolution of the entropy as a function of the focus position, and the position of the measured object in the field, which we have described as $\vec{m}(b)$, we can find the value of absolute defocus which minimizes the amount of defocus of all of them:

$$X^2(b) = \sum_i (e_i - m_i(b))^2,$$

This result gives us directly the position of the best focus for the global image. Figure 4.5.13 shows the evolution of $\|\vec{e} - \vec{m}(b)\|$ as a function of b for a real image obtained with HS1.3. For this image, it was necessary to move the focus −6000 steps to get the best overall focus in the image. We obtained Figure 4.5.14 from another image taken with HS1.3. In this case, it was necessary to move the focus by 1000 motor steps to obtain the best overall focus of the image. In both cases, we can see that the absolute minima of the root mean square of comparison of the vector of entropies of stars obtained by the model at a given focus position. Moreover, as we expected, the actual entropies measured at this same position reach an absolute and stable minimum at the same point we had to move the focus to get to the best focus in practice.

Figure 4.5.13 and Figure 4.5.14 show how, both for large and small values of defocus, the minima of the norm is coincident with the amount of defocus present in the system. Thus, the calculation of the norm is enabling to estimate correctly the value of defocus to be corrected from the images. In the case of small defocus, however, the model presents a broad plateau around the best focus value we expect, so in practice, many positions of the focus could be used equivalently.

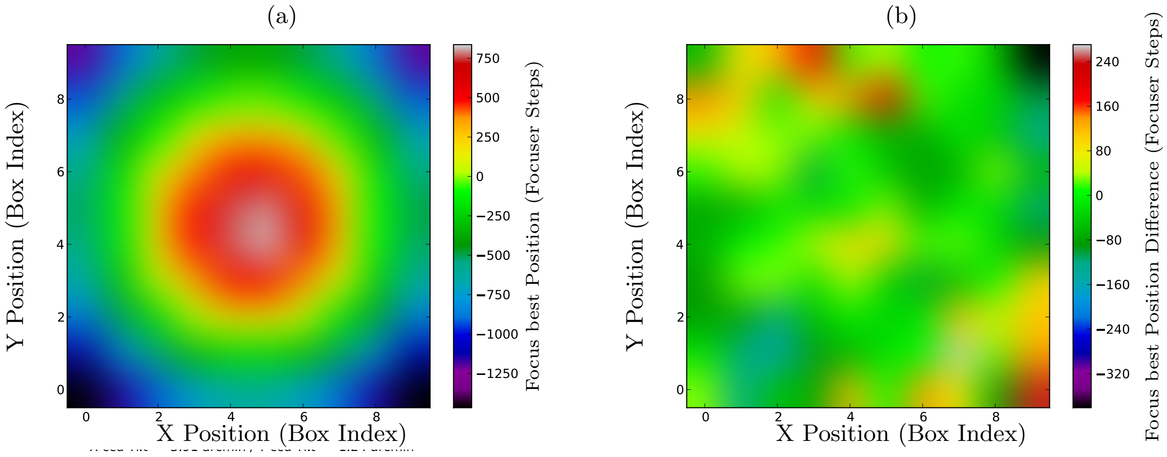


Figure 4.5.6: Optical *zOmt* model, or *Field curvature model* of HS1.4 (a) and residuals (b). Fig.(a) represents the map of the expected best focus positions over the field, in focuser steps, according to the *zOmt* model. The lowest RMS defocus over the field is obtained for the focus position $z = -264.99$ steps. The model converged to an optical center at coordinates $(X_{Shift}, Y_{Shift}) = (-273.55, -131.16)$ pixels off the geometric image center. The tilt found was $(X_{Tilt}, Y_{Tilt}) = (3.91, -1.24)$arcmin. Fig.(b) represets the difference, in focuser steps, between the model presented in Fig.(a) and the measurement of the best focus positions over the field presented in Fig.4.5.5

As a last phase of the process, we implemented a minimization algorithm for finding the best fit between the data in Step 3 and the model in Step2. We used a variety of algorithms to try to find the best focus condition by minimizing X^2 (Levemberg-Marquardt, simplex, and brute-force minimum search over an interval). It happens, however, that the error function presents several minima around the absolute best focus, which makes the convergence with local search algorithms not very stable. In practice, the most robust solution was obtained generating an entropy vector for a grid of focuser positions inside an interval around a guess position. The algorithm returns the absolute minimum norm of residuals as the absolute best focus, and an interpolation around this position can be used as a local refinement.

We tested the reliability of the entropy-based algorithm compared to an FWHM algorithm and their evolution over a night on the Hat-South telescopes HS 1.3. To exercise the robustness of the algorithm, we obtained the calibration of the behavior model with the telescope 1.4 which has the same optical configuration as 1.3 but has a different tilt. The tilt angle difference it what we typically observe between one telescope to another which is between one and two arcminutes on each axis. The figure 4.5.15 shows the evolution of both algorithms over a night of observation and their correlation to the ambient temperature. We can observe that both algorithms give similar results in these conditions which are favorable to the FWHM algorithm. Under more complicated conditions as shown in figure 4.5.16, the first field is further away from the best focus. We observe that the FWHM algorithm hardly converges while the entropy gives an acceptable focus position on the first iteration.
As a consequence, we could check the robustness of the algorithm under harsh conditions, and it showed to be more robust than the traditional FWHM measurements, even if the calibration was done using a telescope which was affected by a different tilt.

4.5.4 Discussion

We have presented a new technique for one-shot focusing based on an entropy-based merit function. Our results show that this technique can be more robust compared to usual techniques based on the use of FWHM or ellipticity as metric of image quality, and becomes a simple computational approach which enables to implement a sensorless adaptive optics approach for small telescopes.
Such robustness can be explained by the shape-independent nature of the entropy function if

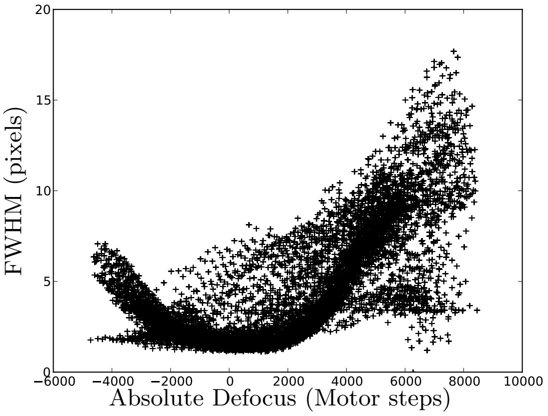


Figure 4.5.7: FWHM metric measured values as a function of absolute defocus (b parameter of the quadratic fit of the evolution of FWHM per zone across focus per zone).

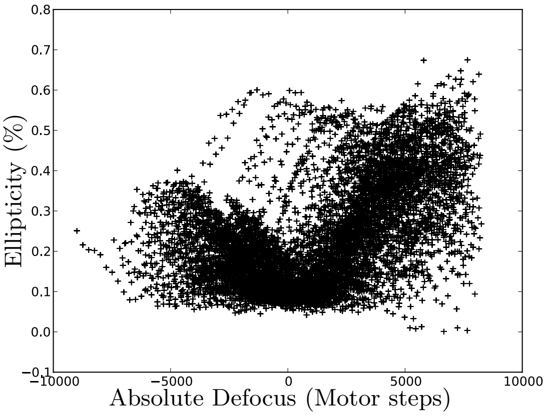


Figure 4.5.8: Ellipticity metric measured values as a function of absolute defocus (b parameter of the quadratic fit of the evolution of Ellipticity per zone across focus per zone).

compared to the other metrics. For both ellipticity and FWHM, we compute the metric from a 2D Gaussian fit of the star. This fit can become unstable as soon as the star is affected by a significant amount of defocus or coma since the shape to be fitted is not approximately Gaussian anymore.
As a result, the focusing technique we have described can become a useful tool when applied to quick f/d ratio or wide-field telescopes. In these configurations, the focusing precision is critical to keep ∼Gaussian-shaped images over the field, as the comma aberration affecting defocused stars near the edge of the field may prevent classical techniques from converging correctly.
As part of the future work of this Thesis, we are currently applying the proposed technique to a control program capable of analyzing in real time images taken in a series and compute the focus correction to apply according to the focus position recorded in the image header. The computed correction can feed either directly to a focuser peripheral or a telescope control scheduler.

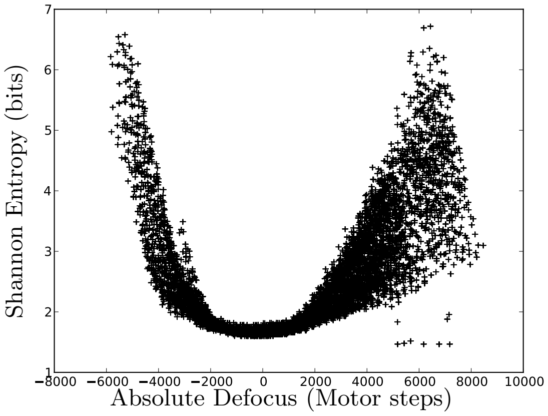


Figure 4.5.9: Entropy metric measured values as a function of absolute defocus (b parameter of the quadratic fit of the evolution of Entropy per zone across focus per zone).

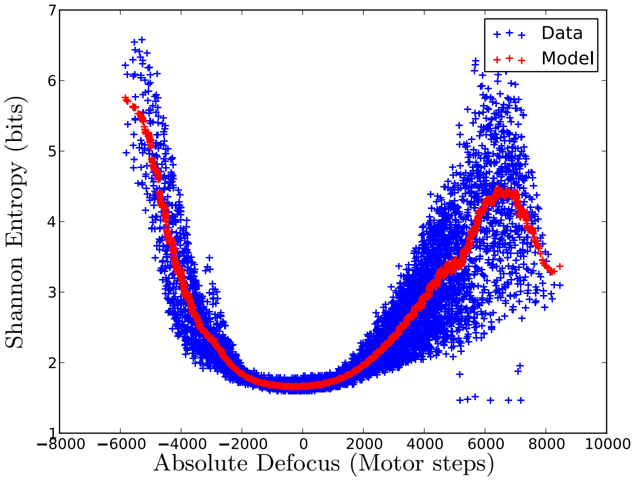


Figure 4.5.10: Entropy value against defocus (blue) and fit to a 1D smoothed spline (red)

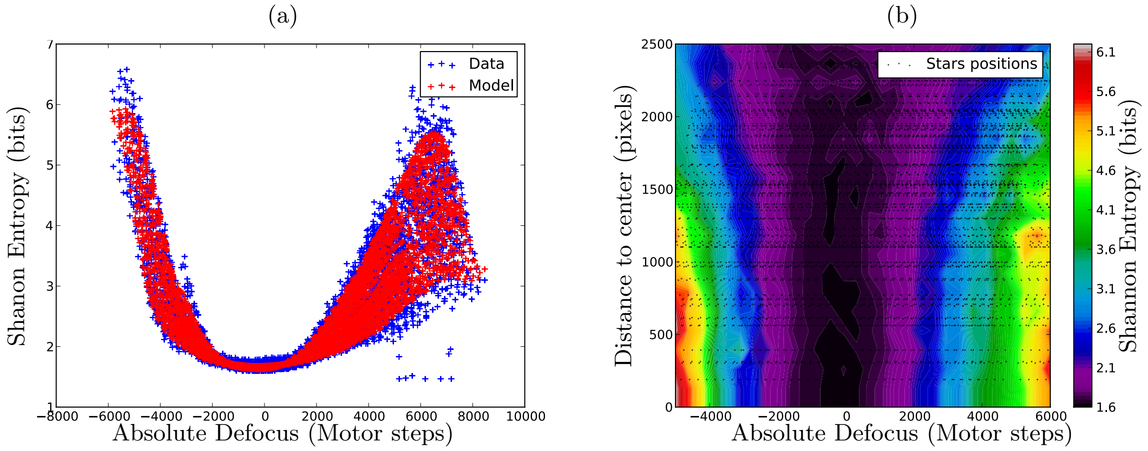


Figure 4.5.11: Entropy as a function of the absolute defocus. The 1-D repressentation in (a) expresses all the measured and modeled values of the entropy vs the absolute defocus. In the 2-D representation in (b) we show the color map of the measured entropy against the distance radius from the optical center. Every black dot represent the position of the entropy measurement in defocus vs the distance from the optical center. The color in between the dots are the interpolated values of the entropy between the measurement we used to build the map.

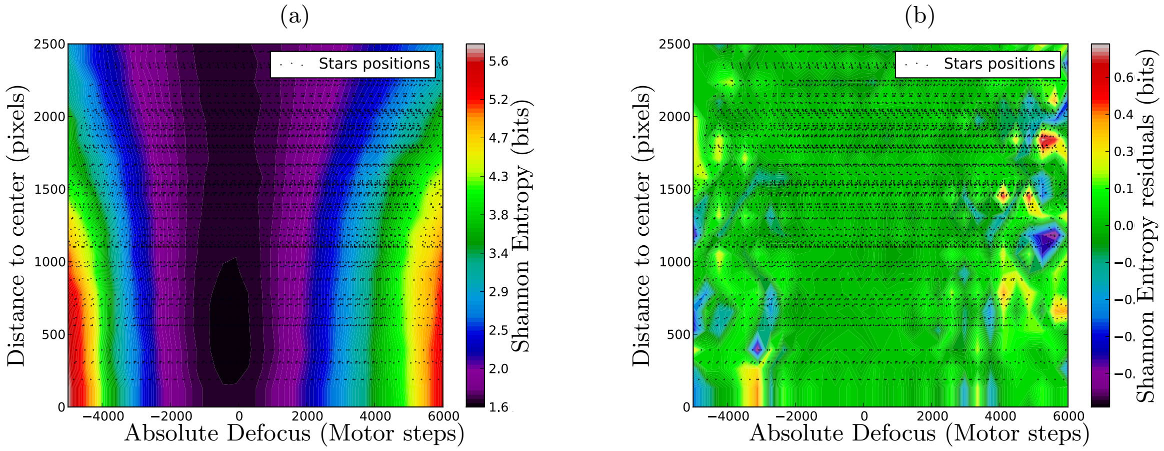


Figure 4.5.12: (a) is a smoothed b-spline version of the map presented in Fig.4.5.11 (b). The graph on the right represents the residuals expressed by the difference of (a) and Fig.4.5.11 (b). Fig.4.5.12 (b) and Fig.4.5.11 (a) show a clearly better convergence than Fig.4.5.10 thanks to the use of a 2D B-Spline model.

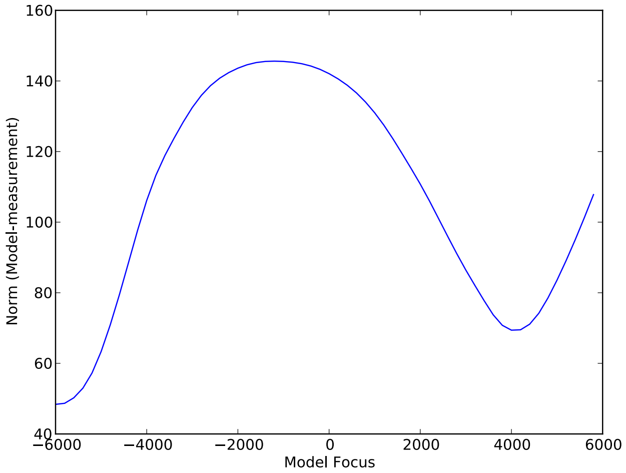


Figure 4.5.13: mean residuals for an image 6000units before focus

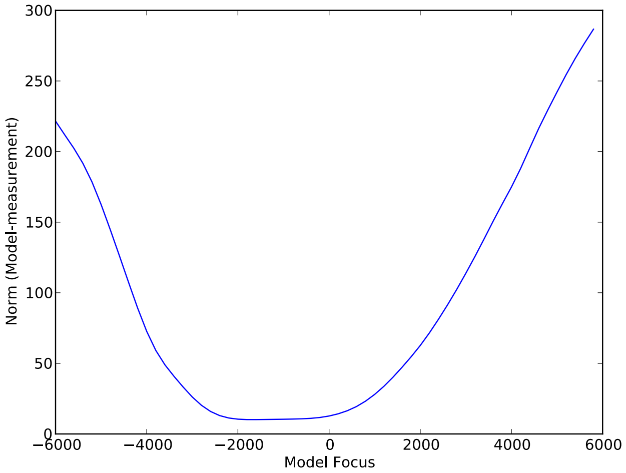


Figure 4.5.14: mean residuals for an image 1000 units before focus

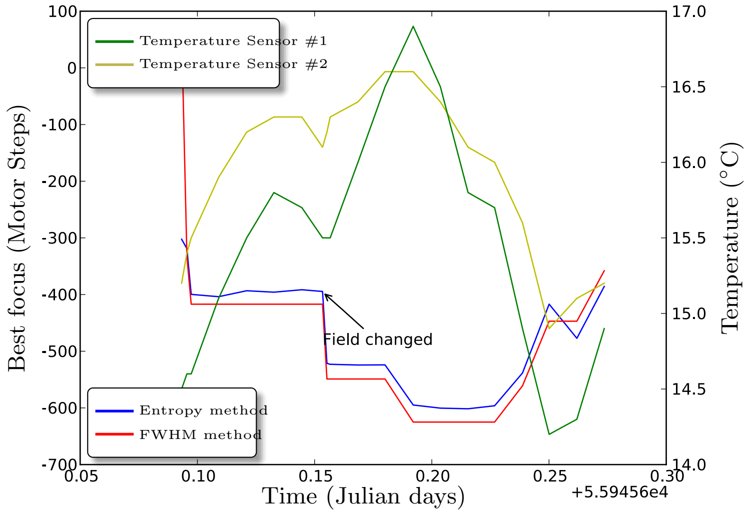


Figure 4.5.15: Comparaison of the results given by the entropy method and the usual FWHM method during an actual observation night. We show the evolution of the resutls given by both algorithms when the telescope is tracking and pointing. (we show with an arrow the moment where the target has been changed) . We also show the evolution of the temperature during the night. During this test the initial best focus guess was only 600 motors steps away from the best focus. It is possible to see that both algorithms converge quiclky. The Entropy method is as efficient as the usual FWHM method in this case.

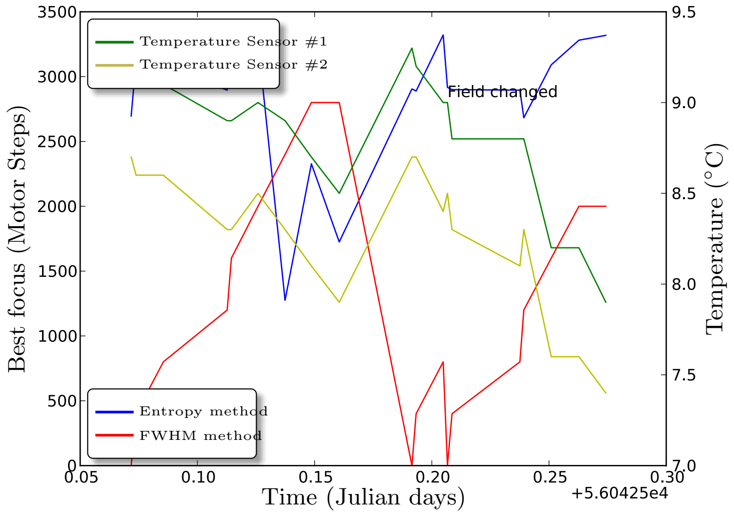


Figure 4.5.16: Comparison of the results given by the entropy method and the usual FWHM method during an actual observation night. We show the evolution of the resutls given by both algorithms when the telescope is tracking and pointing. (we show with an arrow the moment where the target has been changed) . We also show the evolution of the temperature during the night. During this test the initial best focus guess was more than 2000 motors steps away from the best focus. It is possible to see that The entropy algorithm converges quiclky while the the FWHM method can get lost. The Entropy method is much more robust than the usual FWHM method in this case.

5 Construction, Implementation and Results

In this last chapter before conclusions, we will present several applications of the solutions proposed in chapter 4 on actual astronomy projects executed on small telescopes which are the main goal of the developments in this Thesis.

As first implementation, we will present the construction of Virtud50, a 50cm alt-azimuthal telescope built from scratch to test and tune the solutions previously presented. This 50cm telescope is currently installed in Observatoire SIRENE (Silo REhabilité pour Nuits Etoilées) in France in a refurbished 6.5m dome which was unused at the time. This part consisted in modifying and automating the dome, and, specially, in the construction and testing of the 50cm alt-azimuthal telescope equipped with SAPACAN 5-DOF positioning system in its prime focus. We then use this telescope for imaging purposes.

The second application which will be described will refer to the Tololo 40cm and ESO 50cm telescopes. Both stations operate from Santa Martina's Observatory in the surroudings of Santiago, Chile and currently belong to the Pontificia Universidad Catolica de Chile (PUC). The Tololo 40cm telescope is one of the first two 16-inch telescopes deployed by Kitt Peak National Observatory (KPNO), translated to Tololo's Observatory, Chile for the Landolt photometric survey. It was later donated to the PUC in Santiago where it has been equipped with the very first version of the CoolObs system described above in 2014, and it has been used ever since on a daily basis for education purpose at the undergraduate programs of the university. It provides a 36mm front-illuminated Sbig imager. The ESO 50cm telescope is a Cassegrain telescope installed on a fork-equatorial mount. It was designed and manufactured in the decade of the 1960's and installed at La Silla Observatory in 1970. Decommissioned in 2000, it was then donated to PUC and installed at Santa Martina in 2003. After a first telescope control system was retrofitted [131], it became a tool for high-resolution spectroscopy after the installation of the PUCHEROS [132] Echelle fiber-fed spectrograph designed at PUC. The CoolObs TCS system was installed in 2015 and allowed a more precise and stable operation of the spectrograph.

In the third section, we describe the robotization and upgrade of the ESO 1m telescope located at La Silla Observatory. The ESO 1m telescope was the first telescope installed in La Silla in 1966. It now hosts as a primary instrument the FIber Dual EchellE Optical Spectrograph (FIDEOS), a high-resolution spectrograph designed for precise Radial Velocity (RV) measurements on bright stars. To meet the special requirements of this project , it was required to upgrade the Telescope Control System (TCS) and some of its mechanical peripherals . We upgraded the existing TCS into a modern and robust software running on a group of single board computers interacting together as a network, using the CoolObs TCS architecture described before. One of the particularities of the CoolObs TCS is that it allows fusing the input signals of 2 encoders per axis, so that high precision and resolution of the tracking can be achieved with encoders of moderate cost. In the approach taken here, one encoder directly reads the angular position of the telescope's right ascension axis, while the other encoder returns actual position of the motor's axis. The TCS also integrates itself with the FIDEOS instrument in such a way that the operator can interact with all the subsystems through the same remote user interface. Our modern TCS unit allows the user to run observations remotely through a secured Internet web interface, minimizing the need of an on-site observer and thus opening a new age in robotic astronomy for the ESO 1m telescope.

5.1 The Virtud 50cm Telescope with Hexapod

The VIRTUD 50cm Telescope was built to apply the new methods and technical improvements presented in this Thesis. After testing the subsystems and controls in the CD6 laboratory, it was moved to Observatoire Sirene in France in a suitable dome so it could be used and tested under real observing conditions. We implemented on this telescope every technical proposal presented in this thesis. This includes the High precision drive control and new techniques for encoder fusion presented in Section 4.1, the Telescope Control System presented in Section 4.2, the **ZeroC-ICE**-based software architecture presented in 4.3, the 5-DOF parallel patented system SAPACAN presented in 4.4, and the robotic entropy-based control software to position automatically the detector to the best focus and collimation presented in 4.5. It is, thus, the experimental workbench supporting all the algorithmics work presented, adding an innovative design of the tube to take advantage of the SAPACAN system. The full telescope was built from scratch as part of this Thesis. This Section will describe the details of the design and construction of the telescope.

5.1.1 Technical description

5.1.1.1 Optical design

The optical design of the telescope is that of a prime focus telescope with a parabolic primary mirror of 500mm f/4.5 produced by Norman Fullum Optics. Image quality is improved using a commercial Rowe Coma Corrector(RCC) with three lenses produced by Baader Planetarium. Such a corrector, presented in Fig.5.1.1 is designed for optical systems from f/3.5 to f/6 which suits perfectly to the described configuration.

Figure 5.1.1: The Virtud-50cm telescope Rowe Coma Corrector commercially packaged by Baader Planetarium.

5.1.1.2 Mechanical design

The Optical Tube Assembly (OTA) of the Virtud 50cm telescope is an aluminum truss with six branches. The truss itself is made of 40mm aluminum prototyping profiles, so it is an extremely low-cost and lightweight approach, not comparable to typical commercial tube mounts. Even though the performed FEM analysis showed that the tube flexure would remain smaller than 1mm, it would still have significant visible effects and severely affect the image quality. The SAPACAN 5-DOF collimation system described in 4.4 was used to compensate for such flexures and position the detector appropiately. The SAPACAN system, thus, compensates for flexures relaxing significantly the rigidity requirements of the tube. We can see a view of the system installation in Fig.5.1.2

The support of the primary mirror is composed of 6 floating triangles to compensate for the deformation of the mirror itself, which constitute the cell support system. Concept design and FEM analysis were performed using the Plop Freeware software. Results of the optimization and the positioning of the 18 floating points of the primary mirror are presented in Fig.5.1.4.
The lateral support of the mirror was built using two crossed steel cables, ensuring a continuous

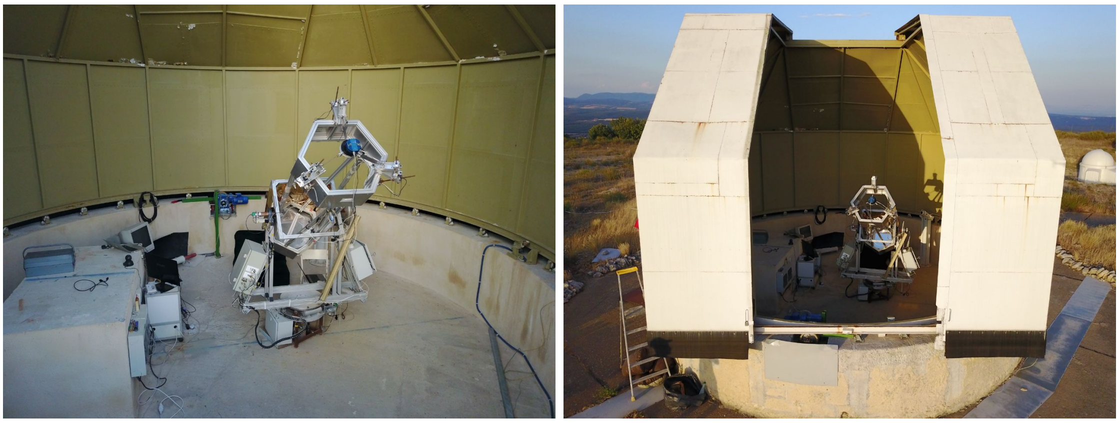


Figure 5.1.2: The Virtud-50cm telescope mounted in its final location. It is possible to apreciate the truss tube design and the SAPACAN 5-DOF remotely controlled collimation system next to the prime focus of the telescope.

lateral force and minimizing the induced astigmatism. The cables go along the width of the mirror in a plane crossing the Center of Gravity (CG). We present the mechanism we used to sustain the mirror in Fig. 5.1.3.

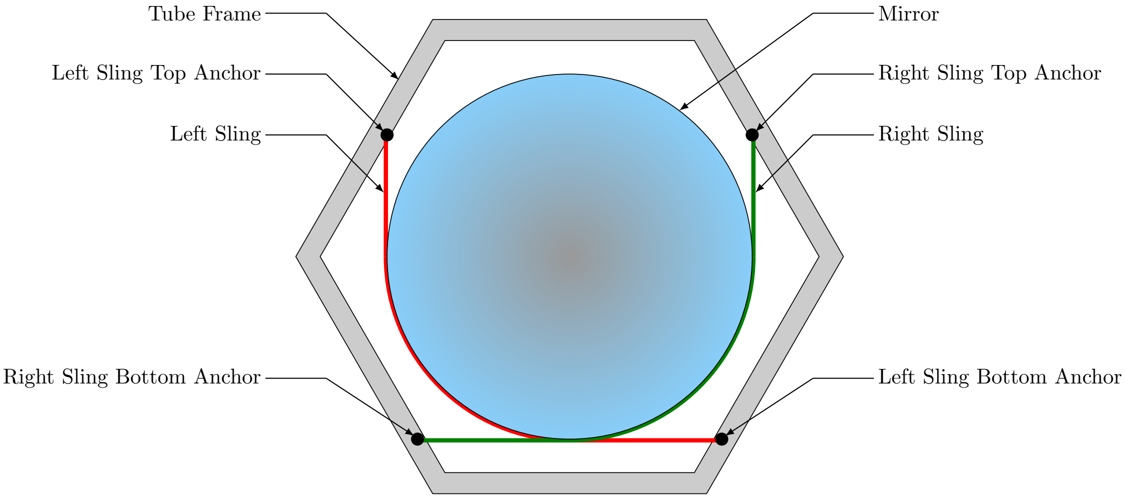


Figure 5.1.3: Attachment method of the primary mirror to the tube. Since the telescope is Alt-Azimuthal it is not necesssary to hold the mirror at the top. The mirror is sustained by two slings which have the advantage of self-centering the mirror inside the tube and provide an homogeneous effort on the periphery of the flange.

Beyond the tube structure made of aluminum profile, the basis and teeth gear wheels of the mount were built on demand by the manufacturer Roberto Castillo, in Chile. A cast aluminum fork mount in an alt-azimuthal configuration was constructed on purpose for the Virtud 50cm. The drive provides a 360 teeth gear and a fitting worm. Encoders of 100.000 steps per revolution equip each of the worm axes and encoders of 1.000.000 steps per revolution equip each of the respective Azimuth and Elevation axes.

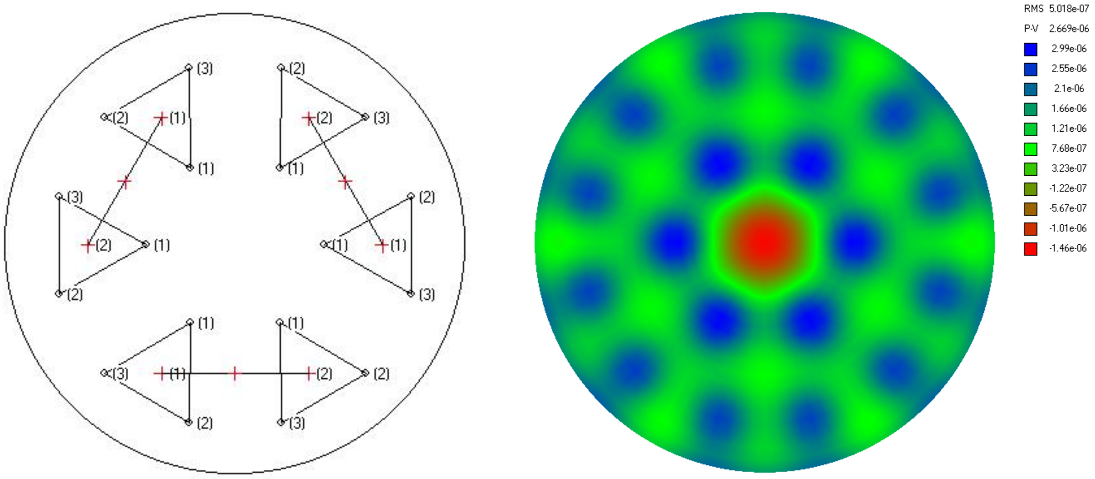


Figure 5.1.4: The Virtud-50cm Cell support design. The mirror is supported by 6 floating triangles (18 contact points). Each pair of triangles is united by a joining lever. The left drawing represents the positioning of the triangles and the position of the supports relative to the mirror, while the right plot shows the resulting mirror deformations at the zenith (in m) according to the FEM model simulated in this configuration.

5.1.1.3 Dome control

Beyond the telescope construction, it was needed to refurbish the former 6.5m unused dome visible in Fig.5.1.5 of the observatory in order to enable remote observations. The associated motorization has been done using two motors of 0.3kW placed 180 degrees apart from the dome center. Traction is performed using precharged wheels placed below the main rail. By measuring the necessary torque force to rotate the dome we obtained a value of 245N/m. As a consequence, considering a dome weight of 98000N and taking into account the friction of the wheel onto the lower part of the dome, each wheel is pressed onto the dome by four springs of 178mm with a compression constant of 67N/mm compressed to 141mm. A variable frequency drive is used as a GYROS LS-SV 008 for each motor, placed in a parallel circuit. We can appreciate the adaptation of the motor to the dome in Fig.5.1.6.
Movement of the dome is obtained using contact with a relay shield from an Arduino Uno connected to the telescope control system. These relays are connected directly to the inputs Clockwise (CW) and Counter ClockWise(CCW) of the variable frequency drives.
We obtain feedback of the position of the dome using a relative quadrature encoder of 1025 counts per revolution (CPR). This encoder measures the angle of the axis of one of the motors. We read the encoder from the same Arduino which controls the position of the dome and receives the target position of the dome continuously through the TCS system.

5.1.1.4 Electronics and control

Motor control is performed using the IPECMOT 48/10 drivers tested in the laboratory. The motor speed is controlled in a closed-loop using the worm encoder which is connected directly to the IPEC encoder input. Each axis encoder is read by a custom board using a PIC 18F26K22 connected to a Raspberry Pi (RPi) computer using a serial/USB as presented in Fig.5.1.7.

The positions of the worm and axis encoders for each axis are measured inside a dedicated RPI. We perform the encoder fusion inside this embedded computer following the procedure described in chapter 4.2.3. Two magnetic limit sensors were installed on each axis to avoid mechanical problems along the telescope motion control, or in case of malfunction. Regarding the elevation control, the upper limit (+90 deg) is active only when the motor moves positively and the negative limit when the target speed is negative. For the azimuth axis, the problem is more complex as the system must be able to point beyond 360 degrees.

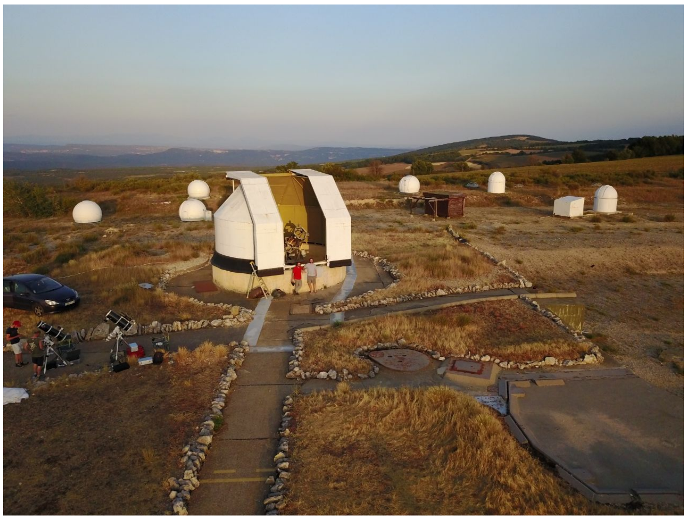


Figure 5.1.5: Aerial view of the refurbished dome at Sirene's Observatory, France.

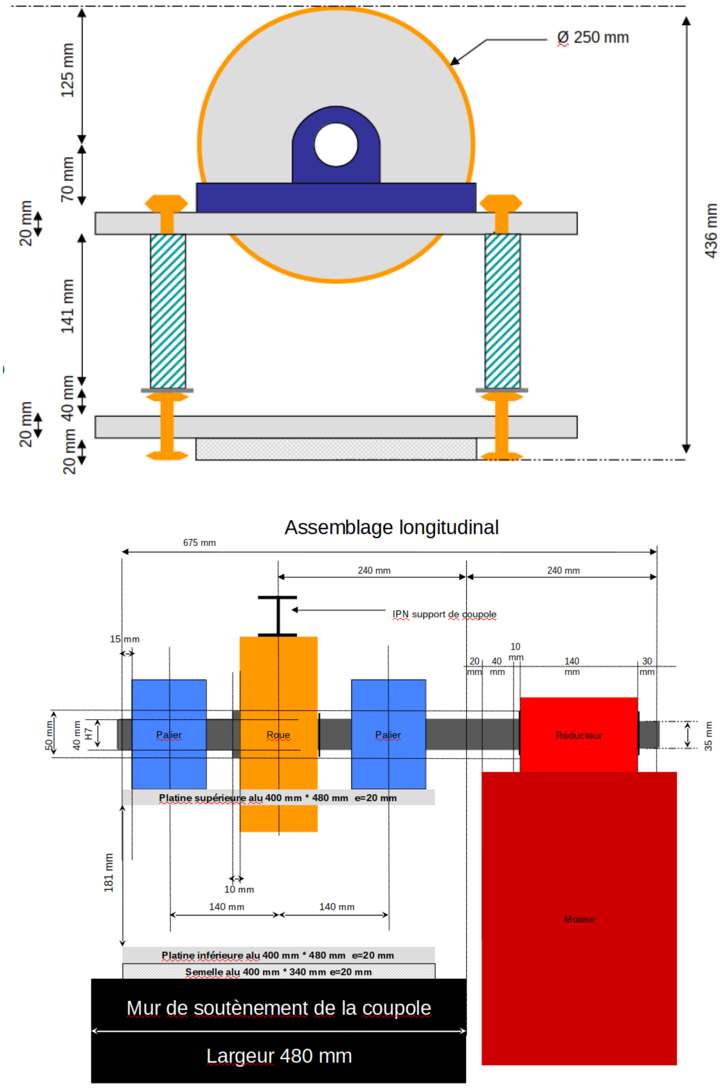


Figure 5.1.6: Installation schematics of the rotation wheel of the dome.

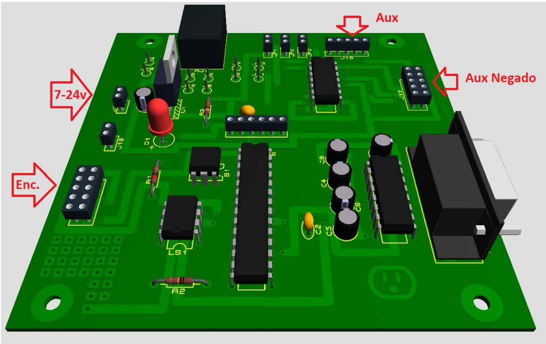


Figure 5.1.7: Interface board for reading the position of the Axis encoder over a serial or serial/usb Port.

As a result, the upper limit (400degrees) is active only when the elevation angle is superior to 180 degrees, and the speed is positive. The negative limit will be active when the latest known position of the telescope azimuth is lower than 180degrees and the target speed negative.

5.1.1.5 Software

After installing CoolObs Software on the computer network, the configuration has been set to use the S20 state of the telescope pointing machine as a final state. As a consequence, the control system performs in alt-azimuthal mode, in a completely transparent way to the user.

5.1.2 Integration and servicing

After testing the system in the laboratory in 2013, it was installed at Sirene in 2014 and used as an exhibition telescope for the annual national outreach event in August.

A July 2015 service mission allowed a significant upgrade to the telescope control system cabling and connections. All cabling was changed, and installed in a new electronic control box. Unfortunately, a major lightning strike during the 2016 winter damaged most of the electronics, including the CCD and the telescope remained unused until the 2016 August service mission. A last service mission of July 2017 allowed to upgrade the control system to the latest version and the control boards from RPI1 to RPI3. The telescope is currently operative and used in observations from Sirene.

5.1.3 Results

The telescope demonstrated being able to track and keep stars over several minutes exposures even in alt-azimuthal mode up to 75 degrees above the horizon. Focusing tilting and collimation were successfully performed using the method defined in section 4.5.

5.2 The PUC 40cm Telescope Imager

This telescope was built in 1963 and initially used as the first telescope ever installed at Kitt Peak National Observatory (KPNO) in Arizona. After being used for years for the generation of the Landolt's Equatorial Standard star catalog, it was then moved to the Cerro Tololo Inter American Observatory (CTIO) in Chile around 1980. CTIO finally donated the telescope to the Pontificia Universidad Catolica University in 2001, so it could be used for teaching and research purposes.It is thus a small telescope of classical design and construction, without any robotic component or system.

Although the university staff attempted at first to upgrade the control system in early 2000 soon after telescope installation in the campus of Santiago, the result obtained was not robust enough to be used easily by students. The astronomy department later moved the telescope to the teaching observatory inside the Santa Martina's Golf club in the outskirts of Santiago. After fitting a commercial telescope control system (Servocat/Argo Navis), it could be observed that the communication protocol (LX200), which is the one used in most of the commercial telescope control systems, had strong inherent limitations which would not allow the telescope to point at field positions with robustness enough to perform good quality images, even after using a reference commercial pointing software (TPOINT). As a consequence, it was decided to implement the first version of the CoolObs TCS in this telescope and use it as a test platform. In this section we will show how we implemented various technical solutions presented in this thesis. This includes the High precision drive control presented in section 4.1, the Telescope Control System presented in Section 4.2, and the **ZeroC-ICE**-based software architecture presented in 4.3.

5.2.1 Technical Specifications

5.2.1.1 Optical Design

In this case, the telescope presents a Ritchey-Chretien Configuration. The primary mirror has a diameter of 16" (406mm), and the resulting focal ratio is 19.5. As a consequence, the resulting field is quite narrow. With a commercial CCD with a square size of 24mm, the resulting field is just 8.1'x 8.1'. Such a narrow field of view is the main reason why the telescope needs a very good precision in pointing; otherwise, objects are easily lost out of the field. As a consequence, the astrometry of the telescope needs to have a good initial guess value to be sure it converges to the correct solution.

5.2.1.2 Mechanical Design

In difference with the Virtud 50cm case, this telescope has a conventional telescope tube. The cell is made of cast aluminium while the tube is a thick aluminium tube. As a consequence, stability and flexures are not an issue. As may be seen in Fig.5.2.1, the mount is a Boller& Chivens German Equatorial (GEM) type with a massive deportation of the Hour Angle axis which allows aiming at the entire sky without the need of flipping over the axis when crossing the meridian, as it typically happens with this kind of mounts.

5.2.1.3 Electronics and Control

At the time of its donation to the university, the right ascension (RA) and declination (DEC) axes were each driven by two asynchronous motors and one stepper motor. This original design used this solution to be able to handle the high dynamic range of possible target speeds for the telescope (tracking, centering and slewing). The setup was initially using this solution because of the hardware available at the time. We replaced it with a more straightforward setup consisting of a total of two DC motors (one for each axis) and their respective modern control and power electronics, which corresponds to the implementation of the high precision drive control described in Section 4.1.
The initial main asynchronous motors available were 1/9 HP Bodine Electric AC motors giving approximately 65 oz-in of torque and 1700 rpm of max speed. Precision movements were performed

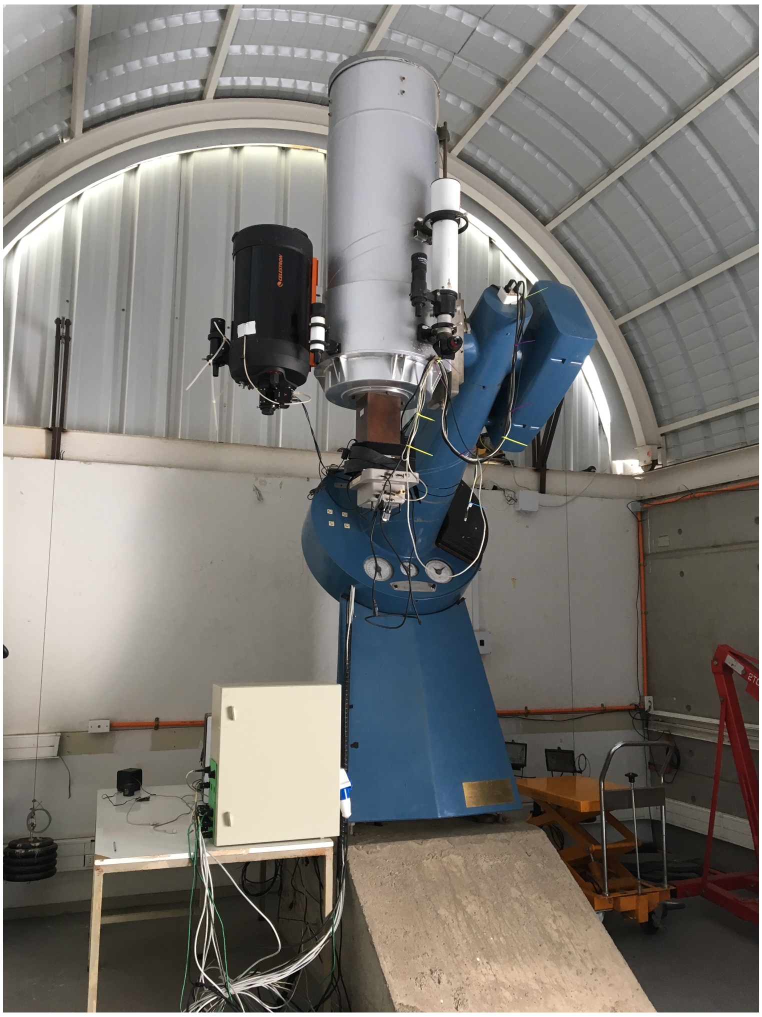


Figure 5.2.1: General view of the PUC40 telescope installed at Santa Martina's Golf Club. It is possible to appreciate the typical Boller & Chivens deported Hour Angle Axis.

with a standard NEMA34 stepper motor. We replaced them by a pair of Pittmann DC054B Series Brush DC 24V motors which have the advantage of allowing a configuration which includes a 5000 ticks/revolution encoder on its axis.

As the encoders embedded within these motors were of high quality, apparently a solution using them could in principle be feasible. However, we considered significantly safer to upgrade to Gurley R158s encoders placed on the worm with 25000 pulses/revolution offering 0.036 arcsec relative resolution when projected on the sky. We can see the mechanical implementation and fitting of the encoder of the mount in Fig.5.2.2. The control loop between the motors and the encoders was closed using the same IPECmot 48/10 industrial DC servo controller which was used in the Virtud50cm alt-azimuthal telescope described in the previous Section.

Position and speed control of both axes were initially driven using a single RaspberryPi 1.0, but after several software upgrades, we upgraded the control devices to three independent RaspberryPi 3.0 units, with the first and second ones used to control the speed of each axis, and the third one for control of the position. All the electronic subsystems described here fit inside the white electronic box visible in the lower left corner of Fig.5.2.1, yielding a quite compact setup.

We present a complete description of the electronical implementation in the documentation of

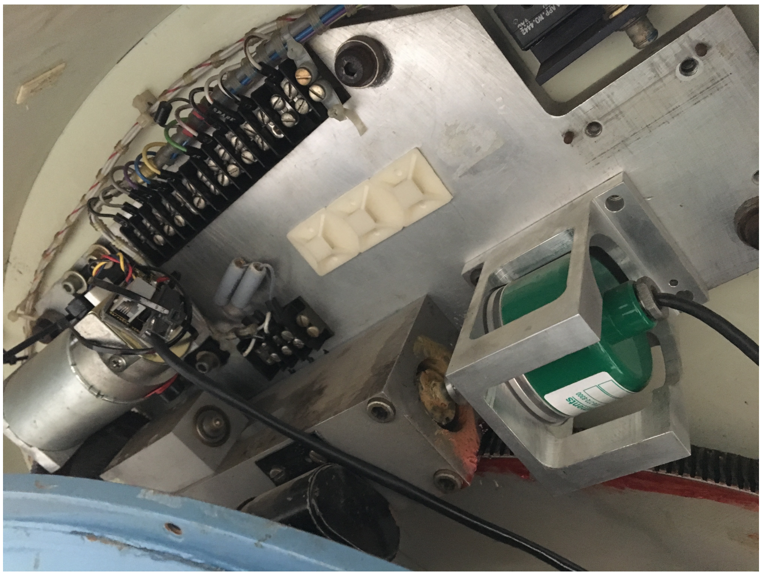


Figure 5.2.2: Motor swap and encoder fitting on the Hour Angle Axis of the telescope.

the electronics box of the telescope in Appendix B.

5.2.2 Implementation and Results

This PUC40 telescope was the first to have the full control system described in Section 4 installed (in 2013), so it was widely used as a testbed platform of the different algorithmic apporaches ever since. As a consequence, most of the latest software improvements and developments were first tested on this telescope. However, it is well known that all installations involve some degree of personalization to adapt to the existent working conditions. In this case, the mechanical configuration of the mount made impossible the installation of an on-axis encoder to dynamically correct for the periodic error as presented in 4.1.3.3. The reason for this was the impossibility to access to the axis since it would involve deep modifications of the complete structure of the mount. However, the high quality of the worm/gear couple and the possibility of precisely tuning their relative alignment have a very positive effect on the quality of the periodic error. The periodic positioning error positioning due to this configuration in this case stays below 2 arcseconds which we could correct by using an external telescope for auto-guiding. A small 80cm refractor is placed in parallel to the main tube. The camera in the focus of this telescope takes an image every 5s and solves the astrometry. Position offsets are sent after each computation to the telescope system as presented in section 4.2.

5.2.2.1 Control Software

Equivalently, the web-based distributed control software to control all the subsystems and peripherals of the telescope described in Section **??** was successfully implemented, becoming the first complete implementation of the telescope algorithmics system presented in Section 4.

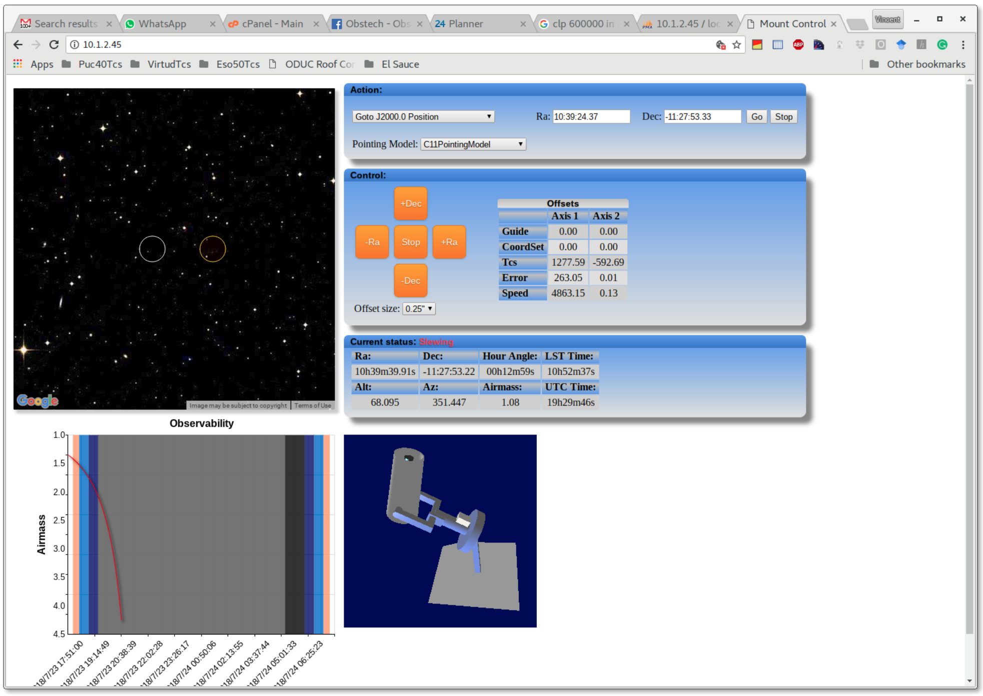
The telescope is controlled from the interface seen in Fig.5.2.3,which is the final layer between all the implemented algorithmics and the astronomer. The operator can be remotely located and so performing a fully robotic observation. The user interface may be seen to be composed of four main parts:

- The top left part is dedicated to the sky map, showing the respective current positions of the telescope and the target.
- The top right part is the main control center and provides three different areas. The first, top one is the action board with the list of possible operations which can be performed by the

mount. The second, below, represents the movement and offsets status of the telescope in the current moment. The last one, at the bottom of the area, displays the current variables defining the position of the telescope and the state of its motion.

- The bottom left part shows the airmass and visibility graph of the current target.
- The bottom right part is a 3D simulated plot representing the current position of the telescope.

On the camera side, we present its interface in Fig.5.2.4. The last image taken from the camera is visible on the left part of the display in a JS9 Applet able to dynamically adjust the contrast of a 16bits image for proper visualization. In complement to the standard observation tools, JS9 also offers some handy observing tools integrated into the applet such as a radial plot of stars to check proper focusing, or catalog overlays. Astrometry is seamlessly integrated into the header so that the current exact mouse position is continuously coherent with the displayed coordinates.



Action:

Goto J2000.0 Position ▾ Ra: 10:39:24.37 Dec: -11:27:53.33

Pointing Model: C11PointingModel ▾

Control:

Offset size: 0.25" ▾

Offsets		
	Axis 1	Axis 2
Guide	0.00	0.00
CoordSet	0.00	0.00
Tcs	1277.59	-592.69
Error	263.05	0.01
Speed	4863.15	0.13

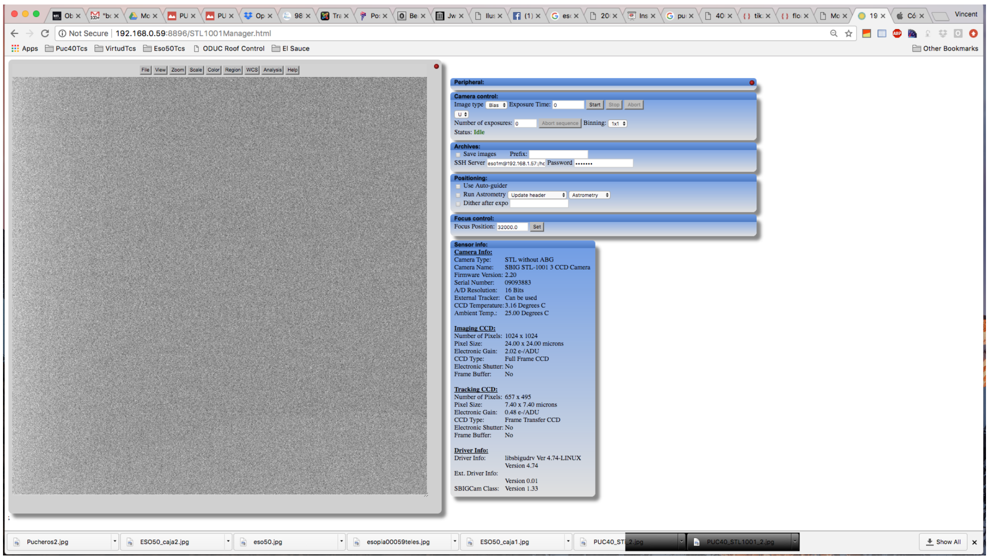
Current status: Slewing

Ra:	Dec:	Hour Angle:	LST Time:
10h39m39.91s	-11:27:53.22	00h12m59s	10h52m37s
Alt:	Az:	Airmass:	UTC Time:
68.095	351.447	1.08	19h29m46s

Figure 5.2.3: Telescope Control System interface Screenshot. The figure shows the TCS interface as it appears to the user during a remote observing session.

5.2.2.2 On-Sky Performance

The telescope has been used in real observing conditions by students and observatory's employees along roughly 500 observation hours for the last thee years. After the application of a 30 position pointing model like the one described in Section 4.2, the system showed to be able to point any object of the sky within 40 arcseconds of the nominal position while the full field of view is 8 minutes. Regarding tracking precision, the system showed no perceptible elongation of stars due to lousy tracking on exposures as long as 180 seconds. Periodic error and drift started to appear when exposure time was increased beyond that value. However, it was possible to arrange the autoguiding system so that it is possible to expose up to 20mn in a single image. It is also possible to accumulate a number of 180s exposure whithout guiding to achieve a longer equivalent exposure. Using this system, equivalent exposures of several hours without star elongation have been obtained. Fig.5.2.5 is an example of this approach taken with the PUC40cm telescope imager. The image is obtained using a stack of 20 exposures of 180s taken with an R filter. It is possible to



Peripheral:

Camera control:
Image type: Bias Exposure Time: 0
U
Number of exposures: 0 Binning: 1x1
Status: Idle

Archives:
□ Save images Prefix:
SSH Server: eso1m@192.168.1.57:/hc Password: •••••••

Positioning:
□ Use Auto-guider
□ Run Astrometry Update header Astrometry
□ Dither after expo

Focus control:
Focus Position: 32000.0

Sensor info:

Camera Info:

Camera Type:	STL without ABG
Camera Name:	SBIG STL-1001 3 CCD Camera
Firmware Version:	2.20
Serial Number:	09093883
A/D Resolution:	16 Bits
External Tracker:	Can be used
CCD Temperature:	3.16 Degrees C
Ambient Temp.:	25.00 Degrees C

Imaging CCD:

Number of Pixels:	1024 x 1024
Pixel Size:	24.00 x 24.00 microns
Electronic Gain:	2.02 e-/ADU
CCD Type:	Full Frame CCD
Electronic Shutter:	No
Frame Buffer:	No

Tracking CCD:

Number of Pixels:	657 x 495
Pixel Size:	7.40 x 7.40 microns
Electronic Gain:	0.48 e-/ADU
CCD Type:	Frame Transfer CCD
Electronic Shutter:	No
Frame Buffer:	No

Driver Info:

Driver Info:	libsbigudrv Ver 4.74-LINUX Version 4.74
Ext. Driver Info:	Version 0.01
SBIGCam Class:	Version 1.33

Figure 5.2.4: Camera Interface Screenshot. The figure shows the real-time interface used for the remote operation of the Main Imaging Camera

see that the shape of the stars has been kept perfectly round during the full exposure time giving rise to a suitable result given the size of the telescope used.

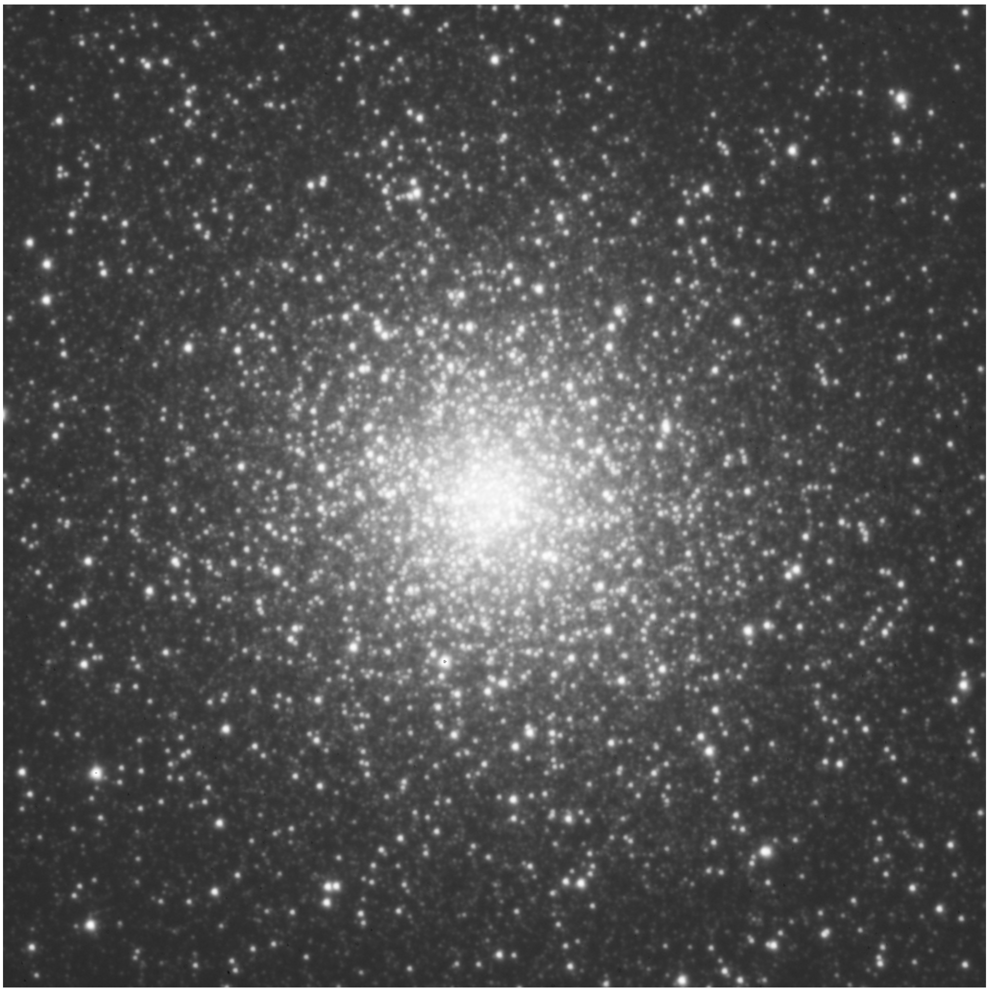


Figure 5.2.5: Integration of a 2h session on the NGC104 Globular cluster using the PUC40cm telescope imager. The result presented is the combined sum of 40 consecutive 180s exposures with a Johnson &Cousins R filter with no auto-guiding. The roundness of the images of stars shows that the system can perfectly handle 3mn exposures without the need of guiding, partially thanks to the effect of using the advanced pointing model described in Section 4.2.

5.3 The ESO 50cm Telescope with PUCHEROS Echelle Spectrograph

As a third example of the implementation of the technologies and algorithms proposed in this Thesis, we present the adaptation of the ESO50cm diameter telescope to a robotic configuration. The ESO50 telescope is a 50cm diameter Cassegrain telescope 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 transported to the Universidad Catolica de Chile's Teaching Observatory (Observatorio Docente UC or ODUC). At first, the control of the telescope was manually driven by an operator but a first automation step 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 for its refurbishing at PUC. It was placed at first in the city campus and then translated to the Santa Martina observatory (ODUC) in 2008. The Observatory is situated in the outskirts of the city of Santiago and is used for student learning sessions.

In this section we will show how we implemented various technical solutions presented in this Thesis in this telescope. This includes the High precision drive control presented in section 4.1, the Telescope Control System presented in Section 4.2, and the **ZeroC-ICE**-based software architecture presented in 4.3.

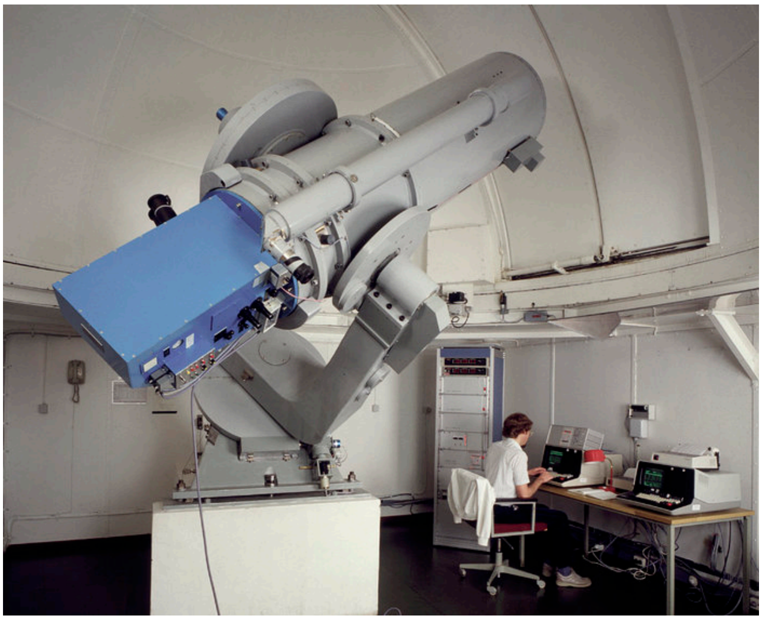


Figure 5.3.1: Initial state of the ESO50 telescope when still installed at La Silla

5.3.1 Previous Upgrades

As mentioned, when the telescope was upgraded within the works of this thesis some previous upgrades had already been implemented. The first upgrade of the telescope was performed by Baffico in 2002 [133]. It kept most of the existing hardware from the 1993 upgrade, and integrated new PIC Micro controllers to control the speed of the motors. Communication with the control computer was handled using a RS232 serial port using the LX200 protocol. The observers used compatible commercial software to control the telescope.

A second software upgrade, performed by Shen in 2012 [134], allowed the system a safer operation by monitoring the drive status and by integrating the whole system into a format compatible with **ZeroC-ICE** (the Internet Communication Engine). Even if this operation could make the interaction with the instrumentation (which was under development at the time) much more flexible, the outdated low level hardware still in use could not allow a robust operation of the complete system. Thus, a new refurbishing was mandatory, and was used in order to turn the telescope into a fully robotic system using the algorithms and methods described in Section 4.

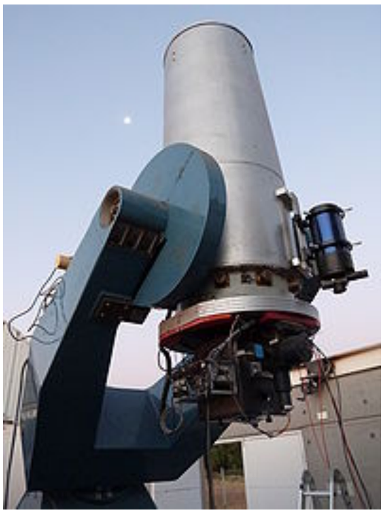


Figure 5.3.2: ESO50 after being moved down to Santa Martina's observatory

5.3.2 CoolObs TCS Adaptation and Results

When our upgrading works started from the point described in the former paragraph the telescope was already using DC motors and relative quadrature encoders on the motor axis, indicating absolutely no mechanical upgrade was performed on the ESO50cm telescope from its start of operations in 1969. The same mechanical upgrade of the control box described for the PUC40cm telescope was adapted to the ESO50cm telescope by adjusting the PID control parameters, the orientation of motors and some resolution factors for adjusting them to the new mechanical characteristics and constraints. The new integrated control system presented in Section 4.1 to 4.3 is visible in the left image of Fig.5.3.3.
We present a complete description of the electronical implementation in the documentation of the electronical box of the telescope in Appendix C. The box itself is indeed very similar to the one used for the PUC40cm telescope presented in the previous section, and in Appendix B with a few variations inherent to the particular cofiguration of the telescope.

The telescope is equipped with PUCHEROS[135] a Fiber-fed Echelle spectrograph, so the field of view of the camera placed in the focal plane is too small to perform any automatic pointing model based on some astrometrical solution. To overcome the problem, the pointing model is generated by centering 30 stars one by one in the pinhole of the instrument. Obviously, this method is significantly more tedious than the one used in the PUC40, although it is not less precise and enables the telescope to achieve a pointing precision as small as 15arcseconds.
In this case, the high precision of the worm/gear made not necessary to install an on-axis encoder to correct for the periodical error in motor displacement. In such conditions, the system can perfectly track a star for more than 20mn without correction within the 2 arcsecond precision required by the diameter of the fiber projected on the sky.
The right part of Fig.5.3.3 shows a second control box in which the computers and electronic boards related to the control of the different peripherals of the instrument have been placed. While the mentioned first box is dedicated in exclusive to the telescope control subsystems, in the second box additional equipment was installed, including the computer controlling the wide field camera in parallel to the telescope, and the motor controller for the flip mirror which allows to select if the light impinging on the fiber comes from the sky or from a calibration lamp.

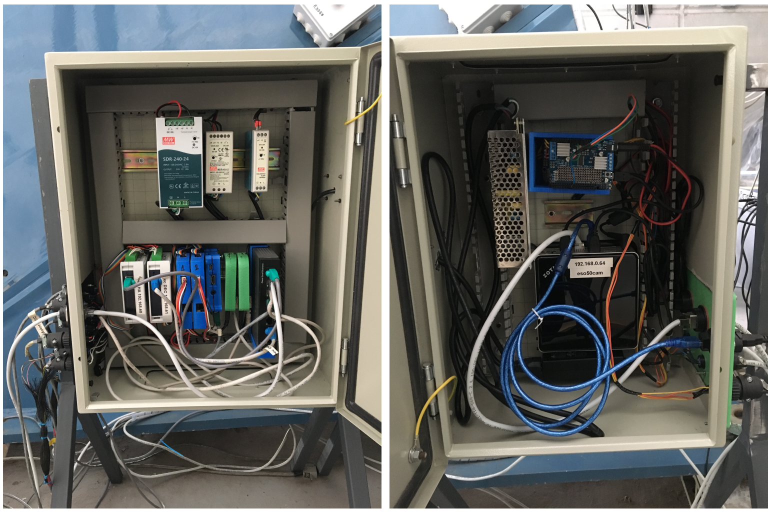


Figure 5.3.3: The new control box and electronics of the 50cm telescope. The TCS and associated susbsystems are embedded in the left-hand box, while imaging-related equipment has been installed in the right-hand box

With the stability and versatility of the implemented Telescope Control System that we presented in Section 4., continuous observations performed with the telescope and the PUCHEROS Spectrograph since the TCS installation in 2012 allowed to obtain the scientific results listed below, which become a reference of the quality of the implementation performed:

- Izzo, L., Mason, E., Vanzi, L., Fernandez, J. M., Espinoza, N., Helminiak, K., & Della Valle, M. (2013). Spectroscopic observations of Nova Cen 2013. The Astronomer's Telegram, 5639.
- Ederoclite, A., 2013, May. T Pyx: towards a new paradigm for nova explosions. In Highlights of Spanish Astrophysics VII (pp. 539-542).
- Berdja, A., Vanzi, L., Jordán, A. and Koshida, S., 2012, September. An Echelle Spectrograph for precise radial velocity measurements in the near IR. In Ground-based and Airborne Instrumentation for Astronomy IV (Vol. 8446, p. 844681). International Society for Optics and Photonics.
- Zapata, A., Vanzi, L., Jones, M. and Brahms, R., 2017, September. Radial Velocity Challenge at the AIUC. In ESO Calibration Workshop: The Second Generation VLT Instruments and Friends.

5.4 The ESO 1m Telescope with FIDEOS Dual Feed Echelle Spectrograph

As a final example of application, we present the works done in the ESO 1-meter telescope, which was the first telescope installed in La Silla Observatory in Chile in year 1966. It was first used as a conventional photometric telescope until 1994, and since then it became fully dedicated to the DENIS [136] project. Due to its obsolete and complicated Control System, the ESO 1-meter telescope was decommissioned in 2005. Since 2015, as part of an agreement between ESO and the Universidad Católica del Norte (UCN), related to the transfer of the Cerro Armazones from the UCN to ESO for the construction of the E-ELT telescope, it was transferred to be on charge of the UCN. As part if this project, the UCN signed an agreement with the Pontificia Universidad Católica de Chile (PUC), for installing on this telescope the FIDEOS spectrograph which was being developed at the Astro-Engineering Center (AIUC) within PUC [137]. The ESO 1-meter telescope main specifications are listed in Table 5.4.1.

The project described here was to restore the ESO 1-meter telescope for its use in modern astronomy missions and to upgrade its Telescope Control System (TCS) according to the specifications described in Section 4. This upgrade was expected to enable precision tracking and remote operation to the telescope which, as explained, was decomissioned in special due to the complexity of its control system.

Table 5.4.1: Characteristics of the ESO-1m telescope.

Name	ESO 1-Meter Telescope
Site	La Silla
Altitude	2375 m
Enclosure	Classical Dome
Optical Design	Cassegrain Reflector
Diameter Primary M1	1.04 m
Mount	Equatorial Fork
First Light date	November 30th 1966

To describe in detail the tasks done, we will first present the initial status of the telescope before starting the upgrade. Next, we will present the hardware which needed to be changed, both electronic and mechanical, and which solutions were chosen for them. In the next section we will present the new telescope control system, showing its functioning principles, a description of its software, the structure of the electronic boards, its Graphic User Interface (GUI), and how the system constructs the telescope's pointing model. Finally, we will present the results obtained up to the time of writing this paper. A final section will describe further work to be done.

5.4.1 Initial Status

The first step for a telescope upgrade is a general diagnosis. All the optical, mechanical, electrical and control elements have to be checked in order to decide whether if they need to be changed or if they meet the new system's requirements, in which case they could be kept. Another relevant criterion besides their current condition is the availability of spare parts of the components, as this type of systems are meant to last for several years so the eventual replacement of components needs to be considered.

5.4.1.1 Mechanics and moving parts

In general, the telescope's mechanical condition was good. Both the structure, including their mobile parts, and the lubrication system were in operative conditions, so a replacement for them was not required. The positioning motors (declination and right ascension), were in good conditions so they could be kept operative in the new system.

The Focusing system, however, was composed of a DC motor using old limit switches, which needed to be replaced by a new, more precise and versatile solution, which was chosen to be a stepper motor. The primary mirror M1 shutter motors could also be kept as they were functioning quite well and their mechanical interface was in perfect conditions. Despite the general good state of the motors, all encoders needed to be replaced by modern models, as the precision of these systems has considerably increased since the construction of the original telescope 50 years ago. There was also an additional problem associated to the availability of spare parts for these encoders, which potentially might have rendered the telescope inoperative in the long term.

5.4.1.2 Optics

Regarding the optics, the primary mirror M1 was pretty dirty and degraded, so it needed to be cleaned and aluminized urgently as its reflectance was only of around 50%. The secondary mirror M2, in difference, seemed to be in good conditions.

5.4.1.3 Dome

The dome needed to be restored and upgraded in order to enable remote operations. The slit didn't close properly due to some mechanical imperfections, although both the rotation and the slit motors were in good condition. The old control system of the dome was also working in perfect conditions, therefore what was required was just an interface between this old system and the new control system which enabled the general remote telescope operation.

5.4.1.4 Control System

The former TCS was mounted in an HP1000 computer, which was at present very unpractical to operate. It didn't allow any kind of remote operation. Also, by now there isn't anybody capable of using it at La Silla. Additionally, as the encoders needed a replacement, all the telescope and its peripherals required a new control system.

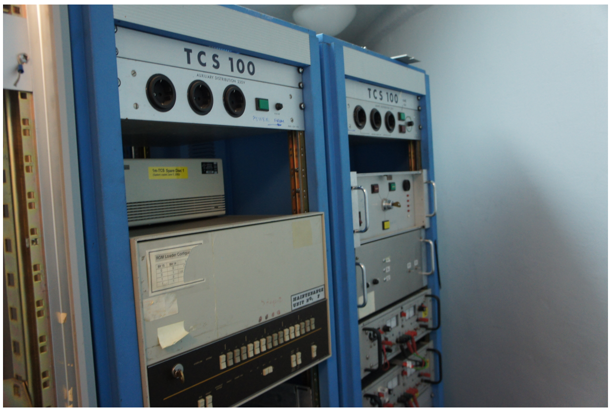


Figure 5.4.1: Old TCS in the ESO1m telescope

This diagnosis led to the conclusion that, beyond some minor upgrades, a full new Telescope Control system was required for the ESO 1m telescope following the algorithmic approaches discussed and detailed in Section 4.

5.4.2 Hardware Upgrade

In terms of hardware, as commented, a series of changes at the level of electronic racks, telescope drives and focuser and dome control needed to be performed. The original power electronics was still using lamp amplifiers so the complete rack had to be removed. The HP1000 computer was also removed. As a result most of the power and control electronic racks and cabling of the telescope control system needed to be entirely changed.

Regarding the main right ascension and the declination drives, the original DC motors of both axes were kept in place, but the encoders connected to the worms measuring the motor axis angle were changed to quadrature encoders delivering 100.000 steps per turn and delivering TTL signal levels.

Right ascension and declination axes were equipped with median resolution encoders installed on-axis with no reduction, giving step size of 1.3 arc-seconds approximately. Since these encoders are placed directly on axis, they are not affected by either gear periodical errors or backlash effects. This dual-encoder per axis arrangement is the same as the one presented in section 4.1 which was later implemented in the Virtud50 telescope presented in section 5.1. In this configuration, the worm encoder allows obtaining a high precision resolution on sky and a smooth movement while the on-axis encoder gives an absolute precision and corrects dynamically fort the periodical error of the worm which cannot be measured from the first encoder. Fig.5.4.2 represents the mechanical setup and encoder placement of the gearing of each axis. The same concept is used for the right ascension and the declination axes.

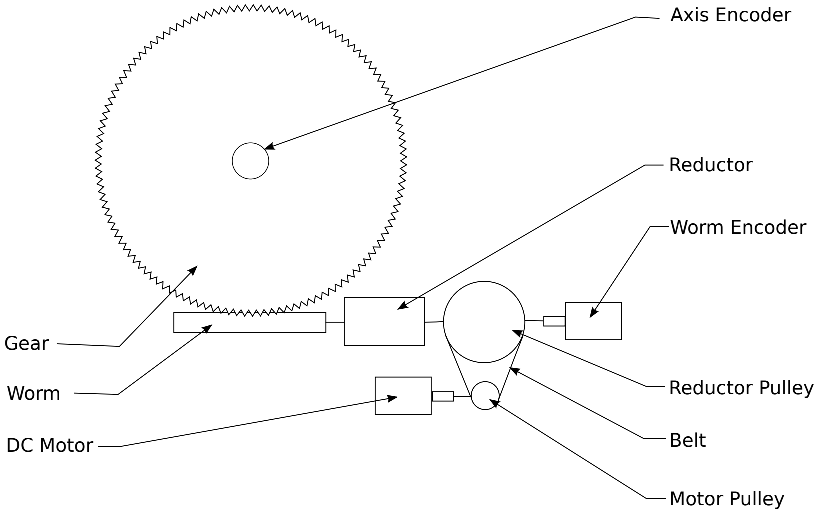


Figure 5.4.2: Mechanical setup of each axis

Focusing of the telescope is in this case provided by the movement of the secondary mirror. This movement is performed using a DC motor coupled to a planetary reductor. Feedback on position was returned by a linear absolute encoder. In the modified system, the reduction gear was kept but the DC motor was replaced by a stepper motor, and the position feedback encoder was removed as it could be provided with precision enough using the position value returned by the stepper motor.

The dome rotation and opening system was also integrated into the control system by modifying the original interface. In this case original motors and relay-based control electronics were kept in their respective original configurations as they were in good condition and delivered adequate functionality. After the changes, commands and interfacing with the TCS are performed using Arduino boards and relay shields which communicate through RS232 commands. The three computer racks in the control room were removed and replaced by two electrical boxes aside the telescope, which may be seen in Fig.4.1.1. The first box (above) hosts the power supplies, the power electronics and the computer for the telescope position control, while the second one below hosts the power supply and control electronics for the mirror covers, focuser and dome control. The physical integration of the TCS is shown in Fig. 5.4.3. Every connector and enclosure was designed in order to respect IP65 standards.

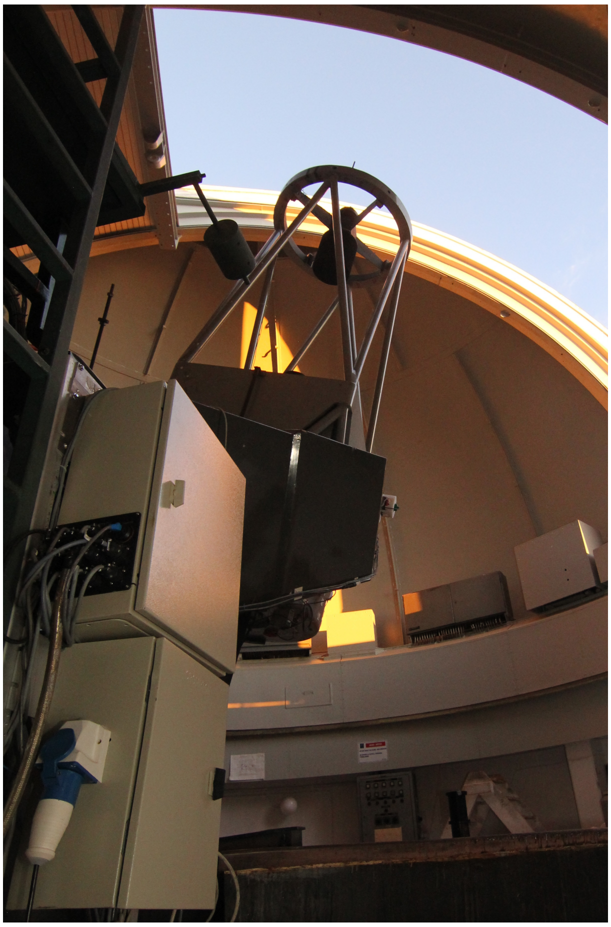


Figure 5.4.3: New TCS installed, showing the position of the two new electronic boxes which implement the TCS

5.4.3 Telescope Control System Upgrade

The TCS software was chosen to run on four independent Raspberry Pi single board computers, interacting together with each otehr so the computing load on each machine is minimized.Right ascension and declination DC motors of the telescope are controlled individually by industrial TCP controllers running an embedded closed loop PID (Proportional-Integral-Derivative) control with a 0.5ms sampling time.

The two first Raspberry Pi are in charge of the speed control of the respective right ascension

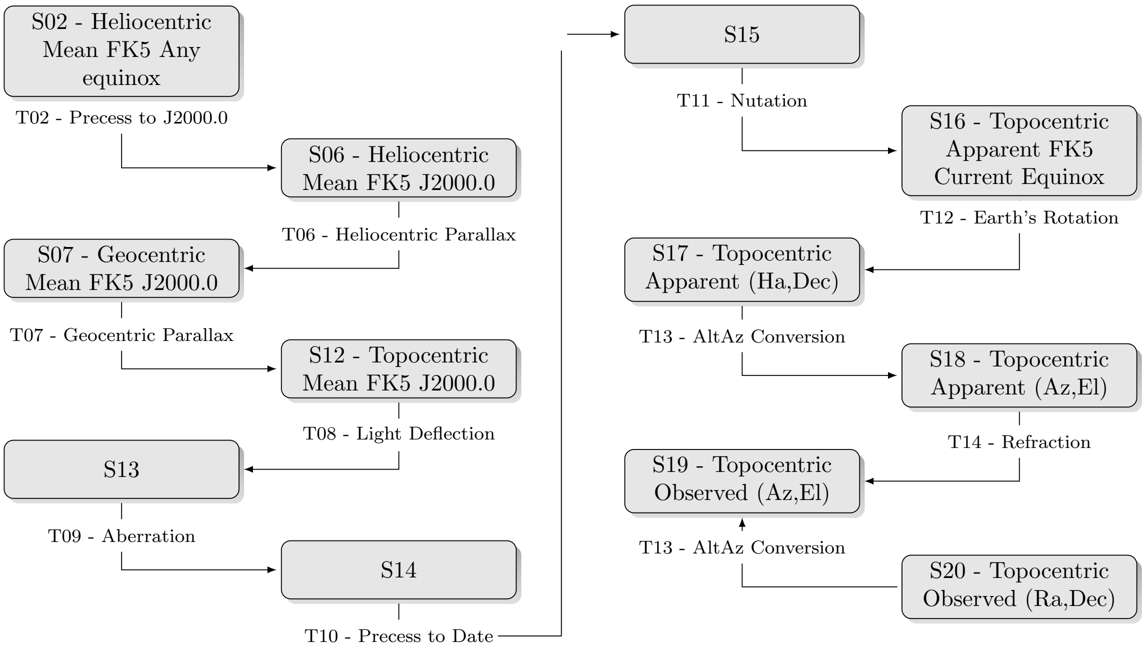

S02 - Heliocentric Mean FK5 Any equinox
T02 - Precess to J2000.0
S06 - Heliocentric Mean FK5 J2000.0
T06 - Heliocentric Parallax
S07 - Geocentric Mean FK5 J2000.0
T07 - Geocentric Parallax
S12 - Topocentric Mean FK5 J2000.0
T08 - Light Deflection
S13
T09 - Aberration
S14
T10 - Precess to Date
S15
T11 - Nutation
S16 - Topocentric Apparent FK5 Current Equinox
T12 - Earth's Rotation
S17 - Topocentric Apparent (Ha,Dec)
T13 - AltAz Conversion
S18 - Topocentric Apparent (Az,El)
T14 - Refraction
S19 - Topocentric Observed (Az,El)
T13 - AltAz Conversion
S20 - Topocentric Observed (Ra,Dec)

Figure 5.4.4: Most useful pointing machine states and transitions according to SLALIB and to the Telescope Pointing Machine definitions.

and declination axes. Since we use two encoders per axis (one placed on the axis of the motor before reduction and the second one on-axis after reduction), both encoders will not have the same resolution and precision on sky. The on-axis encoder gives a 1.3 arc-second per step resolution after interpolation, and will not be affected by mechanical gearing errors or backlashes, while the motor's encoder will give a 0.03 arc-second per step resolution which can be affected by periodical errors or backlashes which amplitudes can be between 5 and 15 arc-seconds. Thus each of these two Raspberry Pi computers will perform the encoder signal fusion using a non linear Kalman Filter (EKF)[106][107] following what was described in Section 4.1. The fusion of both encoders can then give a real precision of 0.25 arc-second on sky. Then, the speed closed loop control of the fused position is also performed for each axis in the corresponding Raspberry Pi.

The third Raspberry Pi hosts the pointing and tracking software itself and is in charge of controlling the speed and position of the previously defined axes. The pointing and tracking software is Python-based and uses the Telescope Pointing Machine Library (TPM) library proposed by Percival[1] and described in Section 4.2. Mechanical non-perpendicularity, polar alignment flexures and similar behaviors are modelled using the PT Wallace Pointing model equations presented in [138, 17]. The mount control consists in a control loop running at a sampling of 100ms, so at each iteration the target coordinates are expressed from the FK5 state to the local mount coordinates s22. As described in section 4.2, a new state is added in the TPM including the position correction computed by the pointing model. As a result the tracking takes into account in real time the pointing model corrections for the tracking behavior which allows longer exposures in addition to a better pointing. Fig.5.4.5 also shows that this configuration allows to include multiple pointing models as independent states of the system. As a result, this provides an easy way of handling various separate instruments installed on the telescope, each with its own pointing model. Passing from the imager to the spectrograph installed in two separate foci of the telescope can result in different offsets, non perpendicularity and flexures behavior, this technique allows to dynamically change the instrument without affecting the pointing or tracking quality of the telescope.

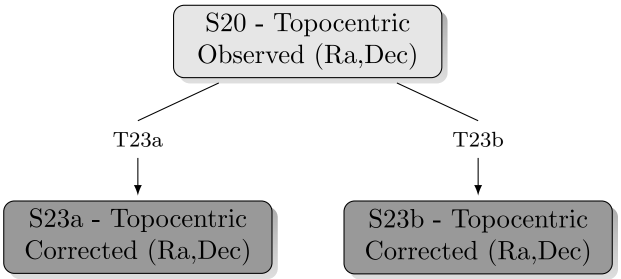

S20 - Topocentric Observed (Ra,Dec)
T23a
T23b
S23a - Topocentric Corrected (Ra,Dec)
S23b - Topocentric Corrected (Ra,Dec)

Figure 5.4.5: New states added in the telescope pointing machine (in dark gray) and their associated transitions.

5.5 Results and Future Work

After the installation at La Silla observatory, we proceeded with the calibration of the encoders and the adjustment of the PID control loop's parameters.
The initial pointing model was obtained with a set of 15 known stars uniformly spread in the sky. The Fig. 5.5.1 shows the pointing residuals which were below 5 arc-seconds RMS. The pointing errors observed during the following nights after the first light of FIDEOS where consistent with the residuals of the pointing model.
The remote control of the telescope has been successfully tested and will remain under commissioning until the end of 2018. The GUI is shown in Fig. 5.5.2. As seen, it is based on the interface model presented in section 4.3 of this Thesis. It allows the user to control the telescope, add new points to the pointing model and controlling all the telescope's peripherals.

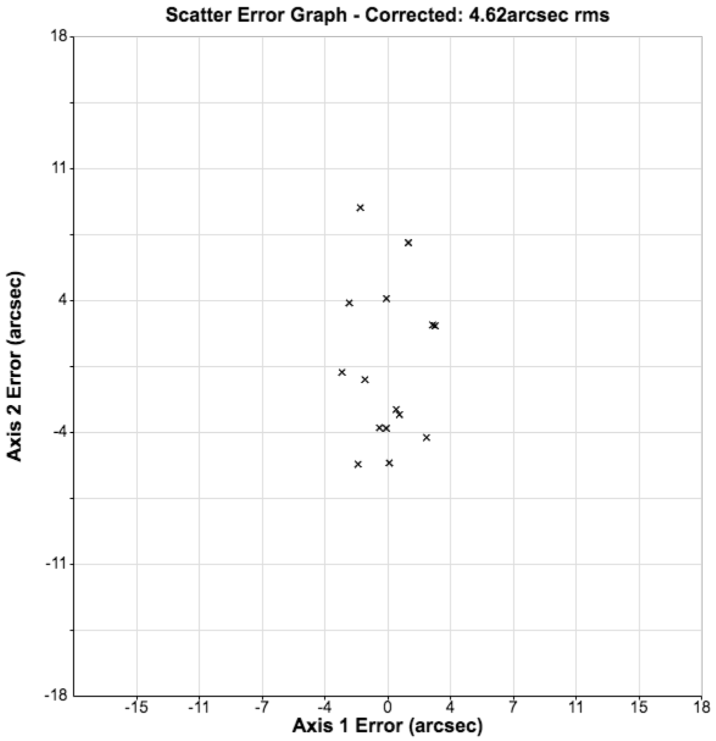
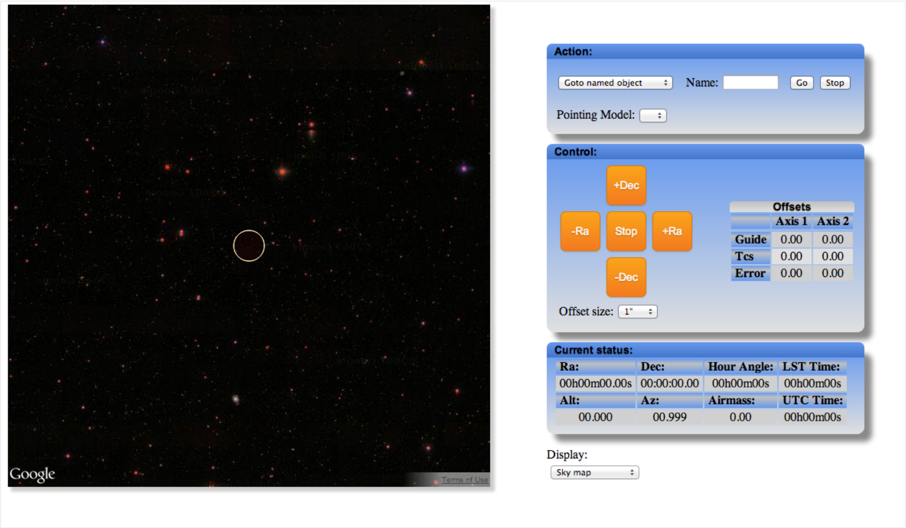


Figure 5.5.1: Pointing residuals of the telescope control system. The graph shows the pointing error in Right Ascension versus the pointing error in Declination express in arc-seconds for 15 bright stars randomly selected in the sky.



Goto named object · Name:

Pointing Model:

Offsets		
	Axis 1	Axis 2
Guide	0.00	0.00
Tcs	0.00	0.00
Error	0.00	0.00

Offset size: 1"

Ra:	Dec:	Hour Angle:	LST Time:
00h00m00.00s	00:00:00.00	00h00m00s	00h00m00s
Alt:	Az:	Airmass:	UTC Time:
00.000	00.999	0.00	00h00m00s

Display: Sky map

Figure 5.5.2: CoolObs TCS GUI

Recent observation and measurement of radial velocities obtained with the ESO1m Telescope using FIDEOS thanks to the stability of the telescope control system implemented and described in section 4 resulted in the following publication:

- Vanzi, L., Zapata, A., Flores, M., Brahm, R., Tala Pinto, M., Rukdee, S., Jones, M., Ropert, S., Shen, T., Ramirez, S. and Suc, V., 2018. Precision stellar radial velocity measurements with FIDEOS at the ESO 1-m telescope of La Silla. Monthly Notices of the Royal Astronomical Society, 477(4), pp.5041-5051.

6 Conclusions

In this Thesis, we presented a complete integrated set of solutions to improve the remote, automatic and unattended operation of a telescope. As detailed in Section 4, we proposed five different innovations to the usual remotely controlled telescope setup, which afterwards were total or partially implemented in up to 4 different upgrades for robotization of telescopes.
We can define an automatic observational station device as an acquisition chain made of a multitude of elements. The work we presented pretends to identify bottlenecks in the robustness of operation of an automatic station at specific points of this data acquisition chain and improve the concept of use of a robotic telescope by improving these bottlenecks, trying tocontribute general-purpose, universal solutions valid for ore than a single system. These innovations we presented take place at spare places of the data acquisition chain, spread from the hardware level to the automatic data processing.
We list here a summary of the presented improvements, their implementation on existing telescopes and the results obtained.

6.1 An Improved Tracking System based in a novel encoder arrangement

In first place, we proposed a low-cost solution to improve the tracking precision of a telescope by using median cost encoders placed on axis. We presented this proposal in chapter 4.1 of this thesis. A novel data fusion algorithm based on Kalman mixing between various measurements of the telescope's axis position allows to precisely control the speed and position of the telescope. We showed how this method allowed to drastically improve the tracking precision by dynamically correcting possible mechanical aberrations of the telescope's mount.
The presented encoder allows a sufficient relative precision to achieve a proper tracking by correcting the typical periodic error of the worm/gear couple of the mount dynamically. However, it may present an internal interpolation periodic error which adds a new tracking aberration. Due to the drastic price difference with an encoder which would permit a direct correction without interpolation correction, we proposed a Kalman fusion algorithm using the data of the motor's position in one input and the encoder in the other one. A proper Kalman model allows to detect and correct for the remaining interpolation error dynamically. By later closing an integrated speed control loop, we could demonstrate that the system can maintain precisely a target speed without deviating from the position it should be at any instant.
We implemented this fusion method in various teaching and scientific instruments such as the Virtud50cm Telescope installed at Sirene Observatory in France, presented in section5.1 and the ESO1m Telescope installed at La Silla Observatory presented in 5.4. The method demonstrated to be extremely efficient since both telescopes were able to track stars within an arcsecond precision for periods above 5 minutes.
Thanks to this method it is now possible to track a star by keeping a precise motor speed on axis for several minutes regardless the precision of the gearing of the mount.

6.2 An Advanced Telescope Control System (TCS) for Improved Guiding

In a second step, we presented a new way to handle the pointing algorithms into a closed loop control to correct in real time the exact position of a telescope according to its misalignments and flexures. We detailed this proposition in Section 4.2. We also showed how our advanced Telescope Control System allows handling external guider systems for performing long exposures by guessing and feed-forwarding drifting errors due to differential flexures. We also detailed in this chapter that another essential advantage of this contribution is its substantially improved robustness at the moment of autoguiding on a field of stars with no prior knowledge of the brightness of stars

present in it. Instead of selecting a single star with a limited range of brightness to obtain a proper guiding, the external guider allows using the complete field of stars and guide on an astrometric solution with absolutely no intervention of the user nor possible mistake at the moment of selecting the guide star. In this case, the telescope control system would use a separate pointing model for the external guider and compute differential drifts between the main imager and the guider continuously. The corrections applied to the mount would then only correspond to the residuals affecting the main imager. As a consequence, the guider guides on a controlled drifting field to suppress any residual drift in the main imager.
We implemented this advanced telescope control system on most of the telescopes presented in section 5 such as the Virtud 50cm telescope in France (Section 5.1), the PUC40 (Section 5.2) and the ESO50cm telescope (Section 5.3) in Santa Martina Observatory in the outskirt of Santiago for teaching and science purposes. It was also successfully implemented on the ESO1m Telescope presented in Section 5.4 in La Silla Observatory with a full dedication to radial velocity measurements. In every telescope, the installation has been a success and showed that the precision and stability of the system allowed to acquire premium quality data used for science measurements, as shown by the different publications obtained using the instruments.

6.3 A Novel Communication and Visualization Protocol Optimized for Network Operation

In addition to that, we proposed a new software protocol dedicated to observatory peripherals intercommunication optimized for network operation and collaborative development in Section 4.3. Instead of grouping all the peripherals of an observing station on a single computer using an imposed programming language, we proposed the CoolObs definitions integrated into the **ZeroC-ICE** communication platform. These definitions group a set of functions and parameters available to each family of peripherals we can find in an observatory. It is possible to integrate the definitions in a driver in a straightforward way, and to develop this driver in an extensive list of programming languages. The communication between drivers and clients is performed over the network and is then independent of the operating system of the drivers or the clients. These configurations, in addition to distributing the charge of computing from one to an unlimited number of machines, also work in favour of large programming communities, where some programmers would prefer a specific language over another depending on their skills or the language Application Programming Interface (API) in which specific hardware distributors propose their documentation.
We also proposed a web-socket based protocol for each driver and clients allowing graphical interfaces to be included inside the same clients or servers and distributed over the web. This interface also remains independent on the operating system of the user, server or client and makes the overall software more versatile than traditional methods centralizing the graphical interfaces in a single client on one user's computer.
We implemented the communication engine and web visualization integrally to the Virtud50cm telescope of Section 5.1, the ESO50cm telescope of Section 5.3, the PUC40cm telescope of Section 5.2 and the ESO1m telescope of Section 5.4. We also implemented this system the dome and focusing control on the CHAT 70cm telescope in Las Campanas. This last telescope is a follow-up telescope for confirmation of exoplanet candidates detected by the HAT-South project. This telescope already participated in the confirmation of more than a dozen exoplanets (and growing) using subsystems derivated from this thesis.

6.4 Low-cost High Precision 5-DOF Positioning System

As a fourth proposal, we presented a new Five Degrees of Freedom (5-DOF) support for prime focus cameras which would allow to dynamically control, at a reduced cost, the positioning in tilt and lateral shift of the camera. We presented this system in Section 4.4. By making a first patented prototype of this device applied on Virtud50cm telescope presented in Section 5.1, we could demonstrate the effectivity of the system which was able to position in a repeatable way and with high precision the camera around the prime focus of the telescope compensating the flexures due to tracking in a low cost, aluminum profile mount. We could demonstrate that the relative positioning precision was better than any possible visible misalignment. The simplicity of the system also showed its advantage of robustness since it is using simple mechanical elements and electronics. We showed that in our case it could replace a hexapod to 1/10th of its cost. Since we are doing movement at low frequencies (1 positioning every 10mn), we do not need in our case the responsivity of the hexapod. The last difference with hexapod is the possibility of rotation around the optical axis which is not a requirement in our case either since the optics are rotationally symmetric. In addition to adequately fulfilling our requirements, this design could ideally fit into other optical application which need precision relative positioning over five axes with limited space in the optical train. This system could be applied since it deports all the mechanics outside the optical plane and only the three slings retaining the primary mirror would cross it.

6.5 Standalone and Robust Focusing Method

Finally, we presented an entropy-based focusing method in section 4.5. Traditionally, focusing is performed by measuring the Full Width at Half Maximum (FWHM) of a single bright star in the middle of the field and moving the focuser inwards and forward until obtaining the best focus position. We proposed a field and shaped independent technique able to find the best focus using a single defocused image by measuring the entropy of boxes around every detected star in the field and comparing it to a behavioral model.

To do so, we first take a series of images by moving the focus position between each of them. In a second step, we then analyze the evolution of entropy of each star while the focus is moving and fit a Gaussian function out of it to find the best focus position as a function of the position of the star in the field. Due to the field curvature, it is then possible to see a radial behavior of best focus around the optical center additionally to tilt effects and model a field behavior model of the telescope. Using this field model we can finally retrofit this model into the best focus model of the stars and make a model of the entropy behavior depending on the distance of best focus of each star and its position in the field. This last step, in conjunction with the previously obtained models, allows us to fit a median optimal distance and direction from the best focus from a single image.

As a consequence, from a single defocused image and an a priori model data of the focus behavior of the instrument, it is possible to know in which direction, and the amount of movement necessary to obtain an optimal focus in a given region.

This method has been developed and implemented on the Virtud 50cm telescope of Section 5.1 but also on the telescopes HS1.1 to 1.4 of the Hat-South Network to demonstrate its efficiency on wide-field telescopes. However, since focus tracking is not a critical point for Hat-South Project (photometry is better with slightly defocused images) and in order to keep back-compatibility with Hat-Net and Hat-South 1.5 to 1.24, it was not applied in the final science detection pipeline.

6.6 Future Work and Evolution

We have proven that it is possible to set a software suite able to handle a complete robotic observatory in a robust and versatile way. We adapted our control system to telescopes in the range of 50cm to 1m of diameter typically used for teaching and university research.

We implemented a large quantity of the peripherals and side software in the robotic observatory Obstech-Observatorio El Sauce which is a spin-off company of Pontificia Universidad Católica de Chile and now hosts more than 25 telescopes on a mountain in the Chilean desert to observe remotely. The material presented in this thesis handles most of the Observatory peripherals such as weather station, automatic roof control and different automation stages.

In a close future, the goal of the presented material is to evolve towards a larger scale of potential users by reducing, even more, its costs and packaging it in a way to adapt itself on small amateur telescopes. We already packaged our Telescope control System electronics and drive-control into one single board computer, and a custom made Arduino shield so that a single reduced size electronic box could control a complete 10-20cm class telescope for a cost within a comparable range of one of these telescopes. The system is currently under testing at Obstech-Observatorio El Sauce as an automatic seeing monitor based on a commercial Celestron 11 telescope in which we retrofitted the CoolObs Software.

After the testing proof, the goal is to package it to most of the commercial mounts and telescope usually purchased by the amateur astronomer's community to facilitate and democratize the automatic observing applied to backyard astronomy. However, we also aim to allow this typical simple observation station to use them as professional devices for remote observing astrophotography and collaborative science.

7 List of Publications

7.1 Patents

- Royo, S.R. and Suc, V., Pontificia Universidad Catolica de Chile and Universitat Politecnica de Catalunya, 2016. Method and system for compensating optical aberrations in a telescope. U.S. Patent 9,300,851.

7.2 Publications derivated from this Thesis

- Suc, V., Royo, S., Jordan, A., Bakos, G. and Penev, K., 2012, September. One-shot focusing using the entropy as a merit function. In Modeling, Systems Engineering, and Project Management for Astronomy V (Vol. 8449, p. 844914). International Society for Optics and Photonics.

- Ropert, S., Suc, V., Jordan, A., Tala, M., Liedtke, P. and Royo, S., 2016, July. TCS and peripheral robotization and upgrade on the ESO 1-meter telescope at La Silla Observatory. In Advances in Optical and Mechanical Technologies for Telescopes and Instrumentation II (Vol. 9912, p. 99124W). International Society for Optics and Photonics.

- Galaz, G., Milovic, C., Suc, V., Busta, L., Lizana, G., Infante, L. and Royo, S., 2015. Deep optical images of Malin 1 reveal new features. The Astrophysical Journal Letters, 815(2), p.L29.

- Shen, T.C., Soto, R., Reveco, J., Vanzi, L., Fernández, J.M., Escarate, P. and Suc, V., 2012, September. Development of telescope control system for the 50cm telescope of UC Observatory Santa Martina. In Software and Cyberinfrastructure for Astronomy II (Vol. 8451, p. 84511T). International Society for Optics and Photonics.

- Royo Royo, S., Arasa Marti, J., Ares Rodríguez, M., Atashkhooei, R., Azcona Guerrero, F.J., Caum Aregay, J., Riu Gras, J., Sergievskaya, I. and Suc, V., 2009. Nuevas líneas de trabajo en metrología óptica en el CD6 de la UPC. In Reunión Nacional de Óptica: Zaragoza, 4-7 de septiembre 2012: libro de abstracts (pp. 461-464).

- Suc, V.,Ropert,S.,Jordan,A, Royo Royo,S. , 2018, Bringing old telescopes to a new robotic life, in Rev. Mex. AA [ACCEPTED]

7.3 Publications using subsystems presented in this Thesis

- Koshida, S., Vanzi, L., Guzman, D., Leiva, R., Bonati, M.A., Avilés, R.L., Baffa, C., Palla, F., Mannucci, F., Shen, T.C. and Suc, V., 2014, July. ARNICA and LonGSp: the refurbishment of two near infrared instruments. In Ground-based and Airborne Instrumentation for Astronomy V (Vol. 9147, p. 91473B). International Society for Optics and Photonics.

- Vanzi, L., Zapata, A., Flores, M., Brahm, R., Tala Pinto, M., Rukdee, S., Jones, M., Ropert, S., Shen, T., Ramirez, S. and Suc, V., 2018. Precision stellar radial velocity measurements with FIDEOS at the ESO 1-m telescope of La Silla. Monthly Notices of the Royal Astronomical Society, 477(4), pp.5041-5051.

- Brahm, R., Hartman, J.D., Jordán, A., Bakos, G.Á., Espinoza, N., Rabus, M., Bhatti, W., Penev, K., Sarkis, P., Suc, V. and Csubry, Z., 2018. HATS-43b, HATS-44b, HATS-45b, and HATS-46b: Four Short-period Transiting Giant Planets in the Neptune–Jupiter Mass Range. The Astronomical Journal, 155(3), p.112.

- Bento, J., Schmidt, B., Hartman, J.D., Bakos, G.Á., Ciceri, S., Brahm, R., Bayliss, D., Espinoza, N., Zhou, G., Rabus, M. and Bhatti, W., 2017. HATS-22b, HATS-23b and HATS-24b: three new transiting super-Jupiters from the HATSouth project. Monthly Notices of the Royal Astronomical Society, 468(1), pp.835-848.

- Bayliss, D., Hartman, J.D., Zhou, G., Bakos, G.Á., Vanderburg, A., Bento, J., Mancini, L., Ciceri, S., Brahm, R., Jordán, A. and Espinoza, N., 2018. HATS-36b and 24 other transiting/eclipsing systems from the HATSouth-K2 Campaign 7 program. The Astronomical Journal, 155(3), p.119.

- Bakos, G.Á., Csubry, Z., Penev, K., Bayliss, D., Jordán, A., Afonso, C., Hartman, J.D., Henning, T., Kovács, G., Noyes, R.W. and Béky, B., 2013. HATSouth: a global network of fully automated identical wide-field telescopes. Publications of the Astronomical Society of the Pacific, 125(924), p.154.

7.4 Other collaborations

- Bakos, G., Csubry, Z., Penev, K., Afonso, C., Bayliss, D., Beky, B., Conroy, P., Csak, B., Henning, T., Jordan, A. and Kovacs, G., 2011, January. Hat-south: A Global Network Of Wide Angle Telescopes Searching For Transiting Extrasolar Planets Around Bright Stars. In Bulletin of the American Astronomical Society (Vol. 43).

- Chen, Y.T., Kavelaars, J.J., Gwyn, S., Ferrarese, L., Côté, P., Jordán, A., Suc, V., Cuillandre, J.C. and Ip, W.H., 2013. Discovery of a new member of the inner Oort cloud from the Next Generation Virgo Cluster Survey. The Astrophysical Journal Letters, 775(1), p.L8.

- Kavelaars, J., Suc, V., Chen, Y.T. and Gwyn, S., 2013. 2010 GB174. Minor Planet Electronic Circulars, 2013.

- Chen, Y.T., Kavelaars, J.J., Gwyn, S., Parker, A., Suc, V., Jordan, A. and Ip, W.H., 2012, May. The Population of Sedna-Like Objects. In Asteroids, Comets, Meteors 2012 (Vol. 1667).

- Szentgyorgyi, A., McLeod, B., Fabricant, D., Fata, R., Norton, T., Ordway, M., Roll, J., Bergner, H., Conroy, M., Curley, D. and Epps, H., 2012, September. The f/5 instrumentation suite for the Clay Telescope. In Ground-based and Airborne Instrumentation for Astronomy IV (Vol. 8446, p. 844628). International Society for Optics and Photonics.

- Penev, K., Bakos, G.Á., Bayliss, D., Jordán, A., Mohler, M., Zhou, G., Suc, V., Rabus, M., Hartman, J.D., Mancini, L. and Béky, B., 2012. HATS-1b: The first transiting planet discovered by the hatsouth survey. The Astronomical Journal, 145(1), p.5.

- Mohler-Fischer, M., Mancini, L., Hartman, J.D., Bakos, G.Á., Penev, K., Bayliss, D., Jordán, A., Csubry, Z., Zhou, G., Rabus, M. and Nikolov, N., 2013. HATS-2b: A transiting extrasolar planet orbiting a K-type star showing starspot activity. Astronomy & Astrophysics, 558, p.A55.

- Bayliss, D., Zhou, G., Penev, K., Bakos, G.Á., Hartman, J.D., Jordán, A., Mancini, L., Mohler-Fischer, M., Suc, V., Rabus, M. and Béky, B., 2013. HATS-3b: An inflated hot Jupiter transiting an F-type star. The Astronomical Journal, 146(5), p.113.

- Jordán, A., Brahm, R., Bakos, G.Á., Bayliss, D., Penev, K., Hartman, J.D., Zhou, G., Mancini, L., Mohler-Fischer, M., Ciceri, S. and Sato, B., 2014. HATS-4b: A dense hot Jupiter transiting a super metal-rich G star. The Astronomical Journal, 148(2), p.29.

- Zhou, G., Bayliss, D., Penev, K., Bakos, G.Á., Hartman, J.D., Jordán, A., Mancini, L., Mohler, M., Csubry, Z., Ciceri, S. and Brahm, R., 2014. HATS-5b: A transiting hot saturn from the HATsouth survey. The Astronomical Journal, 147(6), p.144.

- Hartman, J.D., Bayliss, D., Brahm, R., Bakos, G.Á., Mancini, L., Jordán, A., Penev, K., Rabus, M., Zhou, G., Butler, R.P. and Espinoza, N., 2015. HATS-6b: a warm Saturn transiting an early M dwarf star, and a set of empirical relations for characterizing K and M dwarf planet hosts. The Astronomical Journal, 149(5), p.166.

- Zhou, G., Bayliss, D., Hartman, J.D., Bakos, G.Á., Penev, K., Csubry, Z., Tan, T.G., Jordán, A., Mancini, L., Rabus, M. and Brahm, R., 2013. The mass–radius relationship for very low mass stars: four new discoveries from the HATSouth Survey. Monthly Notices of the Royal Astronomical Society, 437(3), pp.2831-2844.

- Mohler-Fischer, M., Mancini, L., Hartman, J.D., Bakos, G.A., Penev, K., Bayliss, D., Jordan, A., Csubry, Z., Zhou, G., Rabus, M. and Nikolov, N., 2013. VizieR Online Data Catalog: HATS-2b griz light curves (Mohler-Fischer+, 2013). VizieR Online Data Catalog, 355.

- Bakos, G.Á., Penev, K., Bayliss, D., Hartman, J.D., Zhou, G., Brahm, R., Mancini, L., de Val-Borro, M., Bhatti, W., Jordán, A. and Rabus, M., 2015. HATS-7b: A Hot Super Neptune Transiting a Quiet K Dwarf Star. The Astrophysical Journal, 813(2), p.111.

- Bayliss, D., Hartman, J.D., Bakos, G.Á., Penev, K., Zhou, G., Brahm, R., Rabus, M., Jordán, A., Mancini, L., de Val-Borro, M. and Bhatti, W., 2015. HATS-8b: A low-density transiting super-neptune. The Astronomical Journal, 150(2), p.49.

- Brahm, R., Jordán, A., Hartman, J.D., Bakos, G.Á., Bayliss, D., Penev, K., Zhou, G., Ciceri, S., Rabus, M., Espinoza, N. and Mancini, L., 2015. HATS9-b and HATS10-b: Two Compact Hot Jupiters in Field 7 of the K2 Mission. The Astronomical Journal, 150(1), p.33.

- Rabus, M., Jordán, A., Hartman, J.D., Bakos, G.Á., Espinoza, N., Brahm, R., Penev, K., Ciceri, S., Zhou, G., Bayliss, D. and Mancini, L., 2016. HATS-11b AND HATS-12b: Two Transiting Hot Jupiters Orbiting Subsolar Metallicity Stars Selected for the K2 Campaign 7. The Astronomical Journal, 152(4), p.88.

- Rabus, M., Jordan, A., Hartman, J.D., Bakos, G.A., Espinoza, N., Brahm, R., Penev, K., Ciceri, S., Zhou, G., Bayliss, D. and Mancini, L., 2016. VizieR Online Data Catalog: Spectroscopy and photometry of HATS-11 and HATS-12 (Rabus+, 2016). VizieR Online Data Catalog, 515.

- Mancini, L., Hartman, J.D., Penev, K., Bakos, G.Á., Brahm, R., Ciceri, S., Henning, T., Csubry, Z., Bayliss, D., Zhou, G. and Rabus, M., 2015. HATS-13b and HATS-14b: two transiting hot Jupiters from the HATSouth survey. Astronomy & Astrophysics, 580, p.A63.

- Mancini, L., Hartman, J.D., Penev, K., Bakos, G.A., Brahm, R., Ciceri, S., Henning, T., Csubry, Z., Bayliss, D., Zhou, G. and Rabus, M., 2015. VizieR Online Data Catalog: HATS-13b and HATS-14b light and RV curves (Mancini+, 2015). VizieR Online Data Catalog, 358.

- Zhou, G., Bayliss, D., Hartman, J.D., Bakos, G.A., Penev, K., Csubry, Z., Tan, T.G., Jordan, A., Mancini, L., Rabus, M. and Brahm, R., 2015. VizieR Online Data Catalog: 4 transiting FM binary systems (Zhou+, 2014). VizieR Online Data Catalog, 743.

- Zhou, G., Bayliss, D., Hartman, J.D., Rabus, M., Bakos, G.Á., Jordán, A., Brahm, R., Penev, K., Csubry, Z., Mancini, L. and Espinoza, N., 2015. A 0.24+ 0.18 $M_{\odot}$ double-lined eclipsing binary from the HATSouth survey. Monthly Notices of the Royal Astronomical Society, 451(3), pp.2263-2277.

- Ciceri, S., Mancini, L., Henning, T., Bakos, G., Penev, K., Brahm, R., Zhou, G., Hartman, J.D., Bayliss, D., Jordán, A. and Csubry, Z., 2016. HATS-15b and HATS-16b: Two Massive Planets Transiting Old G Dwarf Stars. Publications of the Astronomical Society of the Pacific, 128(965), p.074401.

- Brahm, R., Jordán, A., Bakos, G.Á., Penev, K., Espinoza, N., Rabus, M., Hartman, J.D., Bayliss, D., Ciceri, S., Zhou, G. and Mancini, L., 2016. HATS-17b: A Transiting Compact Warm Jupiter in a 16.3 Day Circular Orbit. The Astronomical Journal, 151(4), p.89.

- Brahm, R., Jordan, A., Bakos, G.A., Penev, K., Espinoza, N., Rabus, M., Hartman, J.D., Bayliss, D., Ciceri, S., Zhou, G. and Mancini, L., 2016. VizieR Online Data Catalog: Spectroscopy and photometry of HATS-17 (Brahm+, 2016). VizieR Online Data Catalog, 515.

- HATS-18b: An Extreme Short-period Massive Transiting Planet Spinning Up Its Star

- Penev, K., Hartman, J.D., Bakos, G.A., Ciceri, S., Brahm, R., Bayliss, D., Bento, J., Jordan, A., Csubry, Z., Bhatti, W. and de Val-Borro, M., 2017. VizieR Online Data Catalog: Sloan i follow-up light curves of HATS-18 (Penev+, 2016). VizieR Online Data Catalog, 515.

- Bhatti, W., Bakos, G.Á., Hartman, J.D., Zhou, G., Penev, K., Bayliss, D., Jordán, A., Brahm, R., Espinoza, N., Rabus, M. and Mancini, L., 2016. HATS-19b, HATS-20b, HATS-21b: Three Transiting Hot-Saturns Discovered by the HATSouth Survey. arXiv preprint arXiv:1607.00322.

- Bento, J., Schmidt, B., Hartman, J.D., Bakos, G.Á., Ciceri, S., Brahm, R., Bayliss, D., Espinoza, N., Zhou, G., Rabus, M. and Bhatti, W., 2017. HATS-22b, HATS-23b and HATS-24b: three new transiting super-Jupiters from the HATSouth project. Monthly Notices of the Royal Astronomical Society, 468(1), pp.835-848.

- Espinoza, N., Bayliss, D., Hartman, J.D., Bakos, G.Á., Jordán, A., Zhou, G., Mancini, L., Brahm, R., Ciceri, S., Bhatti, W. and Csubry, Z., 2016. HATS-25B THROUGH HATS-30B: A HALF–DOZEN NEW INFLATED TRANSITING HOT JUPITERS FROM THE HATSOUTH SURVEY. The Astronomical Journal, 152(4), p.108.

- Espinoza, N., Bayliss, D., Hartman, J.D., Bakos, G.A., Jordan, A., Zhou, G., Mancini, L., Brahm, R., Ciceri, S., Bhatti, W. and Csubry, Z., 2017. VizieR Online Data Catalog: i filter photometry for HATS-25 through HATS-30 (Espinoza+, 2016). VizieR Online Data Catalog, 515.

- de Val-Borro, M., Bakos, G.Á., Brahm, R., Hartman, J.D., Espinoza, N., Penev, K., Ciceri, S., Jordán, A., Bhatti, W., Csubry, Z. and Bayliss, D., 2016. HATS-31b through HATS-35b: Five Transiting Hot Jupiters Discovered By the HATSouth Survey. The Astronomical Journal, 152(6), p.161.

- de Val-Borro, M., Bakos, G.A., Brahm, R., Hartman, J.D., Espinoza, N., Penev, K., Ciceri, S., Jordan, A., Bhatti, W., Csubry, Z. and Bayliss, D., 2017. VizieR Online Data Catalog: Photometry for HATS-31 through HATS-35 (de Val-Borro+, 2016). VizieR Online Data Catalog, 515.

- Bayliss, D., Hartman, J.D., Zhou, G., Bakos, G.Á., Vanderburg, A., Bento, J., Mancini, L., Ciceri, S., Brahm, R., Jordán, A. and Espinoza, N., 2018. HATS-36b and 24 other transiting/eclipsing systems from the HATSouth-K2 Campaign 7 program. The Astronomical Journal, 155(3), p.119.

- Bento, J., Hartman, J.D., Bakos, G.Á., Bhatti, W., Csubry, Z., Penev, K., Bayliss, D., de Val-Borro, M., Zhou, G., Brahm, R. and Espinoza, N., 2018. HATS-39b, HATS-40b, HATS-41b, and HATS-42b: three inflated hot Jupiters and a super-Jupiter transiting F stars. Monthly Notices of the Royal Astronomical Society, 477(3), pp.3406-3423.

- Brahm, R., Hartman, J.D., Jordán, A., Bakos, G.Á., Espinoza, N., Rabus, M., Bhatti, W., Penev, K., Sarkis, P., Suc, V. and Csubry, Z., 2018. HATS-43b, HATS-44b, HATS-45b, and HATS-46b: Four Short-period Transiting Giant Planets in the Neptune–Jupiter Mass Range. The Astronomical Journal, 155(3), p.112.

- Henning, T., Mancini, L., Sarkis, P., Bakos, G.Á., Hartman, J.D., Bayliss, D., Bento, J., Bhatti, W., Brahm, R., Ciceri, S. and Csubry, Z., 2018. HATS-50b through HATS-53b: four transiting hot Jupiters orbiting G-type stars discovered by the HATSouth survey. The Astronomical Journal, 155(2), p.79.
- Zhou, G., Bayliss, D., Hartman, J.D., Rabus, M., Bakos, G.A., Jordan, A., Brahm, R., Penev, K., Csubry, Z., Mancini, L. and Espinoza, N., 2017. VizieR Online Data Catalog: Differential photometry of the EB* HATS551-027 (Zhou+, 2015). VizieR Online Data Catalog, 745.
- Sarkis, P., Henning, T., Hartman, J.D., Bakos, G.Á., Brahm, R., Jordán, A., Bayliss, D., Mancini, L., Espinoza, N., Rabus, M. and Csubry, Z., 2018. HATS-59b, c: A Transiting Hot Jupiter and a Cold Massive Giant Planet Around a Sun-Like Star. arXiv preprint arXiv:1805.05925.
- Schöller, M., Argomedo, J., Bauvir, B., Blanco-Lopez, L., Bonnet, H., Brillant, S., Cantzler, M., Carstens, J., Caruso, F., Choque-Cortez, C. and Derie, F., 2006, June. Recent progress at the Very Large Telescope Interferometer. In Advances in Stellar Interferometry (Vol. 6268, p. 62680L). International Society for Optics and Photonics.

Bibliography

[1] P. Jeffrey, W, "Telescope Pointing Machine Specification," *Space Astronomy Laboratory*, vol. 1, 2005.

[2] R. Spangenburg and K. Moser, *Observing the Universe*, ser. Out of This World. Scholastic Library Publishing, 2004. [Online]. Available: https://books.google.cl/books?id=ekH9GwAACAAJ

[3] R. Burnham, *Burnham's Celestial Handbook, Volume One: An Observer's Guide to the Universe Beyond the Solar System*, ser. Dover Books on Astronomy. Dover Publications, 2013. [Online]. Available: https://books.google.cl/books?id=z3_CAgAAQBAJ

[4] J. Meeus, *Astronomical Algorithms*. Willmann-Bell, 1998. [Online]. Available: https://books.google.cl/books?id=jYynOwAACAAJ

[5] P. Wallace, "Proposals for keck telescope pointing algorithms," *University of Hawaii*, 1994.

[6] ——, "Slalib, positional astronomy library," *Starlink User Note67*, vol. 61, 1995.

[7] J. Cheng, *The Principles of Astronomical Telescope Design*, ser. Astrophysics and Space Science Library. Springer New York, 2010. [Online]. Available: https://books.google.cl/books?id=z-mQWTu7zFoC

[8] D. Schroeder, *Astronomical Optics*. Elsevier Science, 1999. [Online]. Available: https://books.google.cl/books?id=fu3650dUNPcC

[9] J. O. STRAEDE and P. T. WALLACE, "The anglo-australian 3.9-meter telescope: Software controlled slewing, setting, and tracking," *Publications of the Astronomical Society of the Pacific*, vol. 88, no. 525, pp. pp. 792–802, 1976. [Online]. Available: http://www.jstor.org/stable/40676205

[10] D.Clark, "MMT Mount Control System operation and performance ," SAO, Tech. Rep. #02-2, November 2002.

[11] T.-C. Shen, R. Soto, J. Reveco, L. Vanzi, J. M. Fernández, P. Escarate, and V. Suc, "Development of telescope control system for the 50cm telescope of uc observatory santa martina," in *SPIE Astronomical Telescopes+ Instrumentation*. International Society for Optics and Photonics, 2012, pp. 84 511T–84 511T.

[12] P. T. Wallace and K. P. Tritton, "Alignment, pointing accuracy and field rotation of the uk 1.2-m schmidt telescope," *Monthly Notices of the Royal Astronomical Society*, vol. 189, no. 1, pp. 115–122, 1979. [Online]. Available: http://mnras.oxfordjournals.org/content/189/1/115.abstract

[13] P. SPIE, Ed., *A rigourous algorithm for telescope pointing*, vol. 4848, 2002.

[14] P. T. Wallace, "TPOINT – Telescope Pointing Analysis System," *Starlink User Note*, vol. 100, 1994.

[15] *TheSky6 Getting Started Guide Revision 1.11 Copyright 1984-2004 Software Bisque, Inc.*

[16] D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis, "Astrometry.net: Blind astrometric calibration of arbitrary astronomical images," *The Astronomical Journal*, vol. 139, no. 5, p. 1782, 2010. [Online]. Available: http://stacks.iop.org/1538-3881/139/i=5/a=1782

[17] P. Wallace, "Pointing and tracking algorithms for the keck 10-meter telescope," in *Instrumentation for Ground-Based Optical Astronomy*, ser. Santa Cruz Summer Workshops in Astronomy and Astrophysics, L. Robinson, Ed. Springer New York, 1988, pp. 691–706. [Online]. Available: http://dx.doi.org/10.1007/978-1-4612-3880-5_72

[18] P. T. Wallace, “Pointing and tracking software for the Gemini 8-m Telescopes,” in *Optical Telescopes of Today and Tomorrow*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, A. L. Ardeberg, Ed., vol. 2871, Mar. 1997, pp. 1020–1031.

[19] D. L. Terrett, “A C++ class library for telescope pointing,” in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6274, Jun. 2006, p. 12.

[20] P. A. Strittmatter, “Multiple mirror telescopes,” in *Optical Telescopes of the Future*, F. Pacini, W. Richter, & R. N. Wilson, Ed., 1978, pp. 165–184.

[21] D. R. Blanco, “Specifying and holding collimation tolerance on fast cassegrain telescopes,” SAO, Tech. Rep. #87-3, August 1987.

[22] T. W. Mark A. A. Neil, Martin J. Booth, “New Modal Wave-front Sensor: a theoretical analysis,” *J. Opt. Soc. Am*, vol. 17, no. 6, June 2000.

[23] SAO, Ed., *Megacam: Paving the focal plane of MMT with silicon.* Proc. SPIE, July 1998.

[24] D. Debarre, M. J. Booth, and T. Wilson, “Image based adaptive optics through optimisation of low spatial frequencies,” *Opt. Express*, vol. 15, no. 13, pp. 8176–8190, 2007. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-15-13-8176

[25] E. Grisan, F. Frassetto, V. D. Deppo, G. Naletto, and A. Ruggeri, “No wavefront sensor adaptive optics system for compensation of primary aberrations by software analysis of a point source image. 1. methods,” *Appl. Opt.*, vol. 46, no. 25, pp. 6434–6441, 2007. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-46-25-6434

[26] G. Naletto, F. Frassetto, N. Codogno, E. Grisan, S. Bonora, V. D. Deppo, and A. Ruggeri, “No wavefront sensor adaptive optics system for compensation of primary aberrations by software analysis of a point source image. 2. tests,” *Appl. Opt.*, vol. 46, no. 25, pp. 6427–6433, 2007. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-46-25-6427

[27] C. Roddier and F. Roddier, “Wave-front reconstruction from defocused images and the testing of ground-based optical telescopes,” *J. Opt. Soc. Am. A*, vol. 10, no. 11, pp. 2277–2287, 1993. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-10-11-2277

[28] F. Roddier, “Wavefront sensing and the irradiance transport equation,” *Appl. Opt.*, vol. 29, no. 10, pp. 1402–1403, 1990. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-29-10-1402

[29] A. B. Bhatia and E. Wolf, “The zernike circle polynomials occurring in diffraction theory,” *Proceedings of the Physical Society. Section B*, vol. 65, no. 11, p. 909, 1952. [Online]. Available: http://stacks.iop.org/0370-1301/65/i=11/a=112

[30] R. J. Noll, “Zernike polynomials and atmospheric turbulence,” *J. Opt. Soc. Am.*, vol. 66, no. 3, pp. 207–211, 1976. [Online]. Available: http://www.opticsinfobase.org/abstract.cfm?URI=josa-66-3-207

[31] V. N. Mahajan, “Zernike annular polynomials for imaging systems with annular pupils,” *J. Opt. Soc. Am.*, vol. 71, no. 1, pp. 75–85, 1981. [Online]. Available: http://www.opticsinfobase.org/abstract.cfm?URI=josa-71-1-75

[32] S. R. Restaino, S. W. Teare, M. DiVittorio, G. C. Gilbreath, and D. Mozurkewich, “Analysis of the naval observatory flagstaff station 1-m telescope using annular zernike polynomials,” *Optical Engineering*, vol. 42, no. 9, pp. 2491–2495, 2003. [Online]. Available: http://link.aip.org/link/?JOE/42/2491/1

[33] G. ming Dai and V. N. Mahajan, “Zernike annular polynomials and atmospheric turbulence,” *J. Opt. Soc. Am. A*, vol. 24, no. 1, pp. 139–155, 2007. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-24-1-139

[34] M. J. Booth, "Wavefront sensorless adaptive optics for large aberrations," *Opt. Lett.*, vol. 32, no. 1, pp. 5–7, 2007. [Online]. Available: http://ol.osa.org/abstract.cfm?URI=ol-32-1-5

[35] J. Maeda and K. Murata, "Retrieval of wave aberration from point spread function or optical transfer function data," *Appl. Opt.*, vol. 20, no. 2, pp. 274–279, 1981. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-20-2-274

[36] P. Hawkes, "The end of an era?" *Micron*, vol. 24, no. 2, pp. 159 – 162, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/B6T9N-476DYDN-9/2/7dd8a969a2be368de2a3d0e253079eec

[37] G. zhen Yang, B. zhen Dong, B. yuan Gu, J. yao Zhuang, and O. K. Ersoy, "Gerchberg-saxton and yang-gu algorithms for phase retrieval in a nonunitary transform system: a comparison," *Appl. Opt.*, vol. 33, no. 2, pp. 209–218, 1994. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-33-2-209

[38] L. Noethe and S. Guisard, "Analytical expressions for field astigmatism in decentered two mirror telescopes and application to the collimation of the eso vlt," *Astron. Astrophys. Suppl. Ser.*, vol. 144, no. 1, pp. 157–167, 2000. [Online]. Available: http://dx.doi.org/10.1051/aas:2000201

[39] J. I. KRUGLER and A. N. WITT, "An alignment technique for ritchey-chretien telescopes," *Publications of the Astronomical Society of the Pacific*, vol. 81, no. 480, pp. pp. 254–258, 1969. [Online]. Available: http://www.jstor.org/stable/40674724

[40] B. A. McLeod, "Collimation of fast wide-field telescopes," *Publications of the Astronomical Society of the Pacific*, vol. 108, no. 720, pp. pp. 217–219, 1996. [Online]. Available: http://www.jstor.org/stable/40680706

[41] D. Stewart, "A platform with six degrees of freedom," *Proceedings of the Institution of Mechanical Engineers*, vol. 180, pp. 371–386, Jun 1965.

[42] V. Gough and S. Whitehall, "Universal tyre test machine," in *Proc. FISITA 9th Int. Technical Congress*, 1962, pp. 117–137.

[43] K. H. Hunt, *Kinematic geometry of mechanisms*, ser. Oxford engineering science series. Oxford Clarendon Press, 1978.

[44] P. M. Gray, S. C. West, and W. W. Gallieni, "Support and actuation of six secondaries for the 6.5-m mmt and 8.4-m lbt telescopes," in *Advanced Technology Optical Telescopes V*, vol. 2871, 1997, pp. 374–384. [Online]. Available: http://dx.doi.org/10.1117/12.269060

[45] R. Arsenault, R. Biasi, D. Gallieni, A. Riccardi, P. Lazzarini, N. Hubin, E. Fedrigo, R. Donaldson, S. Oberti, S. Stroebele, R. Conzelmann, and M. Duchateau, "A deformable secondary mirror for the vlt," in *Advances in Adaptive Optics II*, vol. 6272, 2006, pp. 62 720V–62 720V–12. [Online]. Available: http://dx.doi.org/10.1117/12.672879

[46] P. Schipani, S. D'Orsi, D. Fierro, and L. Marty, "Active optics control of vst telescope secondary mirror," *Appl. Opt.*, vol. 49, no. 16, pp. 3199–3207, Jun 2010. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-49-16-3199

[47] E. M. Geijo, J. M. Casalta, M. Canchado, M. San Andres, R. Bre, H. Garcia, A. Tomas, L. Zago, and P. Jeffers, "Vista secondary mirror drive performance and test results," in *Optomechanical Technologies for Astronomy*, vol. 6273, 2006, pp. 627 338–627 338–10. [Online]. Available: http://dx.doi.org/10.1117/12.669949

[48] S. M. Gunnels and D. Carr, "Design of the magellan project 6.5-meter telescope: telescope structure and mechanical systems," in *Advanced Technology Optical Telescopes V*, vol. 2199, 1994, pp. 414–427. [Online]. Available: http://dx.doi.org/10.1117/12.176208

[49] J. M. Casalta, J. Arino, M. Canchado, E. M. Geijo, A. Molins, A. Tomas, and L. Zago, "The performances of the gtc secondary mirror drive unit," in *Astronomical Structures and Mechanisms Technology*, vol. 5495, 2004, pp. 507–517. [Online]. Available: http://dx.doi.org/10.1117/12.551868

[50] R. Biasi, D. Veronese, M. Andrighettoni, G. Angerer, D. Gallieni, M. Mantegazza, M. Tintori, P. Lazzarini, M. Manetti, M. W. Johns, P. M. Hinz, and J. Kern, "Gmt adaptive secondary design," in *Adaptive Optics Systems II*, vol. 7736, 2010, pp. 77 3630–77 3630–10. [Online]. Available: http://dx.doi.org/10.1117/12.857983

[51] J. Hartmann, "Ueber die correction eines periodischen fehlers in der bewegung des potsdamer 80 cm refractors," *Astronomische Nachrichten*, vol. 158, no. 1, pp. 1–14, 1902. [Online]. Available: http://dx.doi.org/10.1002/asna.19021580102

[52] G. E. Kron, "Periodic Error in Worm and Gear Telescope Drives," *Publications of the Astronomical Society of the Pacific*, vol. 72, p. 505, Dec. 1960.

[53] R. H. Hardie and C. M. Ballard, "On reducing the periodic error in a telescope drive," *Publications of the Astronomical Society of the Pacific*, vol. 74, no. 438, pp. pp. 242–243, 1962. [Online]. Available: http://www.jstor.org/stable/40673854

[54] D. G. S. Groeneveld, "Considerations in the design of primary worm-gear drives for astronomical telescopes," *Proceedings of the Astronomical Society of Australia*, vol. 1, p. 245, Mar. 1969.

[55] C. Hannel, "Anti-backlash gear assembly," May 31 1988, uS Patent 4,747,321. [Online]. Available: https://www.google.com/patents/US4747321

[56] J. F. R. van der Ven, "A New System to Eliminate Gear Backlash in Telescopes," *The Messenger*, vol. 29, p. 23, Sep. 1982.

[57] M. Fisher, "High-resolution incremental tape encoder on the william herschel telescope," in *1994 Symposium on Astronomical Telescopes & Instrumentation for the 21st Century.* International Society for Optics and Photonics, 1994, pp. 889–900.

[58] M. Warner, V. Krabbendam, and G. Schumacher, "Adaptive periodic error correction for heidenhain tape encoders," *SPIE Astronomical Telescopes+ Instrumentation*, pp. 70 123N–70 123N, 2008.

[59] "Gemini telescope design," http://www.geminitelescope.com/, 2015.

[60] B. Csák, J. Kovács, G. Szabó, L. Kiss, Á. Dózsa, Á. Sódor, and I. Jankovics, "Affordable spectroscopy for 1m-class telescopes: recent developments and applications," 2014.

[61] "Sidereal technology," http://www.siderealtechnology.com/, 2015.

[62] H. Dahlmann, B. Huber, W. Schröder, L. Schuessele, H. Zech, and M. Ravensbergen, "Optical gyro encoder tested on the ntt." *The Messenger*, vol. 74, pp. 5–6, 1993.

[63] R. McWilliams, "Portable telescope mount with integral locator using magnetic encoders for facilitating location of objects and positioning of a telescope," 2003.

[64] M. Ravensbergen, "Main axes servo systems of the vlt," in *1994 Symposium on Astronomical Telescopes & Instrumentation for the 21st Century.* International Society for Optics and Photonics, 1994, pp. 997–1005.

[65] K. Raybould, P. E. Gillett, P. Hatton, G. Pentland, M. Sheehan, and M. Warner, "Gemini telescope structure design," in *1994 Symposium on Astronomical Telescopes & Instrumentation for the 21st Century.* International Society for Optics and Photonics, 1994, pp. 376–393.

[66] K. Szeto, S. Roberts, M. Gedig, G. Austin, C. Lagally, S. Patrick, D. Tsang, D. MacMynowski, M. Sirota, L. Stepp *et al.*, "Tmt telescope structure system: design and development progress report," *SPIE Astronomical Telescopes+ Instrumentation*, pp. 70 122G–70 122G, 2008.

[67] C. Ren, J. Xu, Y. Ye, G. Wang, and X. Jiang, "China song telescope tracking system based on direct drive technology," *SPIE Astronomical Telescopes+ Instrumentation*, pp. 84 491N–84 491N, 2012.

[68] F. Leonard, M. Venturini, and A. Vismara, "Pm motors for direct driving optical telescope," *Industry Applications Magazine, IEEE*, vol. 2, no. 4, pp. 10–16, 1996.

[69] W. Guo-min, "Review of drive style for astronomical optical telescope," *Progress in Astronomy*, vol. 4, p. 008, 2007.

[70] P. Gutierrez, "Standardization of direct drive servos in telescope applications," *Astronomical Telescopes and Instrumentation*, pp. 325–335, 2003.

[71] R. M. Genet, D. Rowe, D. Gray, B. Albertson, W. Chiu, M. Kirkup, D. Murphy, J. Ridgely, J. Schmitt, M. Swanson *et al.*, "An 18-inch direct drive alt-az telescope," in *Society for Astronomical Sciences Annual Symposium*, vol. 27, 2008, p. 123.

[72] L. Chen, Z. Zhang, and H. Wang, "The improvement of ccd auto-guiding system for 2.5m telescope," *SPIE Astronomical Telescopes+ Instrumentation*, vol. 8451, p. 84512K, Sep. 2012.

[73] R. Suszynski and K. Wawryn, "Stars' centroid determination using psf-fitting method," *Metrology and Measurement Systems*, vol. 22, no. 4, pp. 547–558, 2015.

[74] J. S. Lopez, R. J. Tobar, T. Staig, D. A. Bustamante, C. Menay, H. von Brand, and M. Araya, "A reference architecture specification of a generic telescope control system," *Astronomical Data Analysis Software and Systems XIX*, vol. 434, p. 317, 2010.

[75] R. J. Tobar, H. H. von Brand, M. A. Araya, and J. S. López, "An amateur telescope control system: toward a generic telescope control model," *SPIE Astronomical Telescopes+ Instrumentation*, pp. 70 192I–70 192I, 2008.

[76] S. M. Bisque, T. M. Bisque, D. R. Bisque, and M. L. Bisque, "System for operating an astronomical observatory in real time using http," 2007.

[77] J. Irwin, Z. K. Berta, C. J. Burke, D. Charbonneau, P. Nutzman, A. A. West, and E. E. Falco, "On the angular momentum evolution of fully convective stars: Rotation periods for field m-dwarfs from the mearth transit survey," *The Astrophysical Journal*, vol. 727, no. 1, p. 56, 2011.

[78] J. Irwin, D. Charbonneau, P. Nutzman, and E. Falco, "The mearth project: searching for transiting habitable super-earths around nearby m dwarfs," in *IAU Symp*, vol. 253. Cambridge Univ Press, 2009, pp. 37–43.

[79] R. J. Siverd, J. Pepper, K. Stanek, R. W. Pogge, B. S. Gaudi, and D. L. DePoy, "Kelt: A wide-field survey of bright stars for transiting planets," *Proceedings of the International Astronomical Union*, vol. 4, no. S253, pp. 350–353, 2008.

[80] P. F. Sydney, J. L. Africano, A. Fredericks, K. M. Hamada, V. S. Hoo, D. L. Nishimoto, P. W. Kervin, S. Bisque, and M. Bisque, "Raven automated small telescope systems," in *International Symposium on Optical Science and Technology*. International Society for Optics and Photonics, 2000, pp. 237–247.

[81] F. Melsheimer and R. Genet, "A computerized low-cost 0.4-meter research telescope," *International Amateur-Professional Photoelectric Photometry Communications*, vol. 15, p. 33, 1984.

[82] B. Mossop, "Telescope tracking system," 2013.

[83] D. di Cicco, “S&t test report: The telescope drive master,” *Sky and telescope*, vol. 122, no. 4, pp. 60–63, 2011.

[84] A. Pál and G. A. Bakos, “Astrometry in wide-field surveys,” *Publications of the Astronomical Society of the Pacific*, vol. 118, no. 848, pp. 1474–1483, 2006. [Online]. Available: http://www.journals.uchicago.edu/doi/abs/10.1086/508573

[85] G. Bakos, C. Afonso, T. Henning, A. Jordan, M. Holman, R. W. Noyes, P. D. Sackett, D. Sasselov, G. Kovacs, Z. Csubry, and A. Pal, “Hat-south: A global network of southern hemisphere automated telescopes to detect transiting exoplanets,” in *Transiting Planets*, ser. Proceedings of the International Astronomical Union, vol. 4, 5 2008, pp. 354–357. [Online]. Available: http://journals.cambridge.org/article_S174392130802663X

[86] D. Reichart, M. Nysewander, J. Moran, J. Bartelme, M. Bayliss, A. Foster, J. Clemens, P. Price, C. Evans, J. Salmonson *et al.*, “Prompt: panchromatic robotic optical monitoring and polarimetry telescopes,” *arXiv preprint astro-ph/0502429*, 2005.

[87] A. Shporer, T. Brown, T. Lister, R. Street, Y. Tsapras, F. Bianco, B. Fulton, and A. Howell, “The lcogt network,” in *The Astrophysics of Planetary Systems: Formation, Structure, and Dynamical Evolution*, ser. Proceedings of the International Astronomical Union, vol. 6, 10 2010, pp. 553–555. [Online]. Available: http://journals.cambridge.org/article_S1743921311021193

[88] G. Christie, “Detecting exoplanets by gravitational microlensing using a small telescope,” *arXiv preprint astro-ph/0609599*, 2006.

[89] L. Books, “Free astronomy software: Kstars, celestia, stellarium, starlink project, digital universe atlas, instrument neutral distributed interface,” 2010.

[90] O. Fors, F. Montojo, J. Nunez, J. Muiños, J. Boloix, R. Baena, R. Morcillo, and M. Merino, “Status of telescope fabra roa at montsec: Optical observations for space surveillance & tracking,” *arXiv preprint arXiv:1109.5903*, 2011.

[91] F. Montojo, O. Fors, J. Muinos, J. Núñez, R. López-Morcillo, R. Baena, J. Boloix, T. López-Moratalla, and M. Merino, “The fabra-roa telescope at montsec (tfrm): A fully robotic wide-field telescope for space surveillance and tracking,” *arXiv preprint arXiv:1109.5918*, 2011.

[92] M. Jelínek, P. Kubánek, M. Nekola, and R. Hudec, “Bart: an intelligent grb and sky monitoring telescope (2000–2004),” *Astronomische Nachrichten*, vol. 325, no. 6-8, pp. 678–678, 2004.

[93] P. Kubánek, M. Jelínek, M. Nekola, M. Topinka, J. Strobl, R. Hudec, T. d. J. M. Sanguino, A. d. U. Postigo, and A. J. Castro-Tirado, “Rts 2- remote telescope system, 2 nd version,” *Gamma-Ray Bursts: 30 Years of Discovery*, vol. 727, pp. 753–756, 2004.

[94] P. Kubánek, M. Jelínek, S. Vítek, A. de Ugarte Postigo, M. Nekola, and J. French, “Rts2: a powerful robotic observatory manager,” *Astronomical Telescopes and Instrumentation*, pp. 62 741V–62 741V, 2006.

[95] P. Kubánek, “Rts2,Äîthe remote telescope system,” *Advances in Astronomy*, vol. 2010, 2010.

[96] P. KLIBANEK, M. Jelínek, S. Vítek, A. de Ugarte Postigo, M. Nekola, J. French, and M. PROUIZA, “Status of robotics telescopes driven by rts2 (bart, bootes, fram and watcher),” *Il Nuovo cimento della Società italiana di fisica. B*, vol. 121, no. 12, pp. 1501–1502, 2006.

[97] P. Kubánek, M. Jelínek, J. French, M. Prouza, S. Vítek, A. J. Castro-Tirado, and V. Reglero, “The rts2 protocol,” *SPIE Astronomical Telescopes+ Instrumentation*, pp. 70 192S–70 192S, 2008.

[98] S. Fraser, “Scheduling for robonet-1 homogenous telescope network,” *Astronomische Nachrichten*, vol. 327, no. 8, pp. 779–782, 2006.

[99] Y. Tsapras, R. Street, K. Horne, C. Snodgrass, M. Dominik, A. Allan, I. Steele, D. Bramich, E. Saunders, N. Rattenbury *et al.*, "Robonet-ii: Follow-up observations of microlensing events with a robotic network of telescopes," *arXiv preprint arXiv:0808.0813*, 2008.

[100] A. Chavan, G. Giannone, D. Silva, T. Krueger, and G. Miller, "Nightly scheduling of eso's very large telescope," in *Astronomical Data Analysis Software and Systems VII*, vol. 145, 1998, p. 255.

[101] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[102] F. Förster, N. López, J. Maza, P. Kubánek, and G. Pignata, "Scheduling in targeted transient surveys and a new telescope for chase," *Advances in Astronomy*, vol. 2010, 2009.

[103] P. Kubanek, "Genetic algorithm for robotic telescope scheduling," *arXiv preprint arXiv:1002.0108*, 2010.

[104] M. Warner, V. Krabbendam, and G. Schumacher, "Adaptive periodic error correction for heidenhain tape encoders," pp. 70 123N–70 123N–8, 2008. [Online]. Available: http://dx.doi.org/10.1117/12.787462

[105] Y. Zimmerman, Y. Oshman, and A. Brandes, "Improving the accuracy of analog encoders via kalman filtering," *Control Engineering Practice*, vol. 14, no. 4, pp. 337 – 350, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0967066105000407

[106] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[107] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of basic engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[108] P. Nachtwey. (2015) Feed forwards augment pid control. [Online]. Available: http://www.controleng.com/single-article/feed-forwards-augment-pid-control/e6dcd10be91de5c64e075ac84749e77d.html

[109] J. McCormac, D. Pollacco, I. Skillen, F. Faedi, I. Todd, and C. A. Watson, "Donuts: A science frame autoguiding algorithm with sub-pixel precision, capable of guiding on defocused stars," *Publications of the Astronomical Society of the Pacific*, vol. 125, no. 927, pp. 548–556, 2013. [Online]. Available: http://www.jstor.org/stable/10.1086/670940

[110] D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis, "Astrometry.net: Blind astrometric calibration of arbitrary astronomical images," *The Astronomical Journal*, vol. 139, no. 5, p. 1782, 2010. [Online]. Available: http://stacks.iop.org/1538-3881/139/i=5/a=1782

[111] J. Medkeff, "The ascom revolution," *Sky and Telescopes*, vol. 99, no. 5, May 2000.

[112] J. Mutlaq. Instrument neutral distributed interface white paper. [Online]. Available: http://www.clearskyinstitute.com/INDI/INDI.pdf

[113] ——. (2017) Indi open astronomy instrumentation. [Online]. Available: http://www.indilib.org

[114] J. Rumbaugh, I. Jacobson, and G. Booch, *Unified Modeling Language Reference Manual, The (2nd Edition)*. Pearson Higher Education, 2004.

[115] C. Pernechele, F. Bortoletto, and K. Reif, "Hexapod control for an active secondary mirror: general concept and test results," *Appl. Opt.*, vol. 37, no. 28, pp. 6816–6821, Oct 1998. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-37-28-6816

[116] P. Schipani, L. Noethe, C. Arcidiacono, J. Argomedo, M. Dall'Ora, S. D'Orsi, J. Farinato, D. Magrin, L. Marty, R. Ragazzoni, and G. Umbriaco, "Removing static aberrations from the active optics system of a wide-field telescope," *J. Opt. Soc. Am. A*, vol. 29, no. 7, pp. 1359–1366, Jul 2012. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-29-7-1359

[117] P. Nanua, K. Waldron, and V. Murthy, "Direct kinematic solution of a stewart platform," *Robotics and Automation, IEEE Transactions on*, vol. 6, no. 4, pp. 438–444, Aug 1990.

[118] P. Schipani, "Hexapod kinematics for secondary mirror aberration control," *Mem. SAIt Suppl*, vol. 9, pp. 472–474, 2006.

[119] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators.* John Wiley & Sons, 1999.

[120] B. Dasgupta and T. Mruthyunjaya, "The stewart platform manipulator: a review," *Mechanism and Machine Theory*, vol. 35, no. 1, pp. 15 – 40, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094114X99000063

[121] S. Royo and V. Suc, "Método y sistema para compensar aberraciones ópticas en un telescopio," Mar. 1 2012, wO Patent App. PCT/ES2011/070,541. [Online]. Available: https://www.google.cl/patents/WO2012025648A1?cl=es

[122] F. Gao, W. Li, X. Zhao, Z. Jin, and H. Zhao, "New kinematic structures for 2-, 3-, 4-, and 5-dof parallel manipulator designs," *Mechanism and Machine Theory*, vol. 37, no. 11, pp. 1395 – 1411, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094114X02000447

[123] O. Piccin, B. Bayle, B. Maurin, and M. de Mathelin, "Kinematic modeling of a 5-dof parallel mechanism for semi-spherical workspace," *Mechanism and Machine Theory*, vol. 44, no. 8, pp. 1485 – 1496, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0094114X0800236X

[124] G. Meunier, B. Boulet, and M. Nahon, "Control of an overactuated cable-driven parallel mechanism for a radio telescope application," *Control Systems Technology, IEEE Transactions on*, vol. 17, no. 5, pp. 1043–1054, Sept 2009.

[125] H. Li, O. Tutunea-Fatan, and H.-Y. Feng, "An improved tool path discretization method for five-axis sculptured surface machining," *The International Journal of Advanced Manufacturing Technology*, vol. 33, no. 9-10, pp. 994–1000, 2007. [Online]. Available: http://dx.doi.org/10.1007/s00170-006-0529-z

[126] V. Suc, S. Royo, A. Jordán, G. Bakos, and K. Penev, "One-shot focusing using the entropy as a merit function," pp. 844 914–844 914–10, 2012. [Online]. Available: http://dx.doi.org/10.1117/12.927021

[127] K. Kuehn and R. Hupe, "Real-Time Analysis of Large Astronomical Images," *ArXiv e-prints*, Mar. 2012.

[128] M. Booth, "Wave front sensor-less adaptive optics: a model-based approach using sphere packings," *Opt. Express*, vol. 14, no. 4, pp. 1339–1352, 2006. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-14-4-1339

[129] G. Bakos, C. Afonso, T. Henning, A. Jordán, M. Holman, R. W. Noyes, P. D. Sackett, D. Sasselov, G. Kovács, Z. Csubry, and A. Pál, "Hat-south: a global network of southern hemisphere automated telescopes to detect transiting exoplanets," *Cambridge Univ. Press*, 2009. [Online]. Available: http://edoc.mpg.de/447965

[130] E. Bertin and S. Arnouts, "Sextractor: Software for source extraction," *Astron. Astrophys. Suppl. Ser.*, vol. 117, no. 2, pp. 393–404, 1996. [Online]. Available: http://dx.doi.org/10.1051/aas:1996164

[131] T.-C. Shen, R. Soto, J. Reveco, L. Vanzi, J. M. Fernandez, P. Escarate, and V. Suc, "Development of telescope control system for the 50cm telescope of uc observatory santa martina," pp. 8451 – 8451 – 8, 2012. [Online]. Available: http://dx.doi.org/10.1117/12.925567

[132] L. Vanzi, J. A. Chacon, M. Baffico, G. Avila, C. Guirao, T. Rivinus, S. Stefl, and D. Baade, "Pucheros: a low-cost fiber-fed echelle spectrograph for the visible spectral range," pp. 7735 – 7735 – 7, 2010. [Online]. Available: http://dx.doi.org/10.1117/12.857020

[133] M. Baffico, G. Avila, D. Baade, E. Bendek, C. Guirao, O. Gonzalez, P. Marchant, V. Salas, I. Toledo, S. Vasquez *et al.*, "Observatorio uc at santa martina: A small observing facility operated by puc," in *Ground-based and Airborne Telescopes II*, vol. 7012. International Society for Optics and Photonics, 2008, p. 70122O.

[134] T.-C. Shen, R. Soto, J. Reveco, L. Vanzi, J. M. Fernández, P. Escarate, and V. Suc, "Development of telescope control system for the 50cm telescope of uc observatory santa martina," in *Software and Cyberinfrastructure for Astronomy II*, vol. 8451. International Society for Optics and Photonics, 2012, p. 84511T.

[135] L. Vanzi, J. Chacon, K. G. Helminiak, M. Baffico, T. Rivinius, S. Štefl, D. Baade, G. Avila, and C. Guirao, "Pucheros: a cost-effective solution for high-resolution spectroscopy with small telescopes," *Monthly Notices of the Royal Astronomical Society*, vol. 424, no. 4, pp. 2770–2777, 2012. [Online]. Available: + http://dx.doi.org/10.1111/j.1365-2966.2012.21382.x

[136] N. Epchtein, B. De Batz, E. Copet, P. Fouqué, F. Lacombe, T. Le Bertre, G. Mamon, D. Rouan, D. Tiphène, W. B. Burton, E. Deul, H. Habing, J. Borsenberger, M. Dennefeld, A. Omont, J. C. Renault, B. R. Volmerange, S. Kimeswenger, I. Appenzeller, R. Bender, T. Forveille, F. Garzon, J. Hron, P. Persi, M. Ferrari-Toniolo, and I. Vauglin, "Denis: A deep near-infrared survey of the southern sky," *Astrophysics and Space Science*, vol. 217, no. 1, pp. 3–9, 1994. [Online]. Available: http://dx.doi.org/10.1007/BF00990013

[137] M. Tala, A. Berdja, M. Jones, L. Vanzi, S. Ropert, M. Flores, and C. Viscasillas, "Fideos: a high resolution echelle spectrograph for the eso 1m telescope at la silla," pp. 914 789–914 789–7, 2014. [Online]. Available: http://dx.doi.org/10.1117/12.2056551

[138] P. T. Wallace, "A rigorous algorithm for telescope pointing," pp. 125–136, 2002. [Online]. Available: http://dx.doi.org/10.1117/12.460914

Appendices

A Slice Definitions

```
#pragma once
module CoolObs
{

  //Define vector types
  sequence<string> StringVector;
  sequence<double> DoubleVector;
  sequence<int> IntVector;
  sequence<short> ShortVector;


  //Define the main master ice class from which every ice obect derivates from
  interface CoolObsPeripheral
  {
    void GetStatus(out string Status);
    void Shutdown();
  };

  //Define the mount class
  interface GenericMount extends CoolObsPeripheral
  {
    void Home();
    void GotoPosition(double Ra2000, double Dec2000, double RaNonSiderealSpeed, double DecNonSiderealSpeed);
    void GotoAltAz(double Az,double Alt,out bool result);
    void GetObject(string Name,out double ra,out double dec);
    void GetObjectEphem(string Name, out double az, out double alt, out double airmass,out string ha,out int Limits);
    void GetCoordEphem(double ra,double dec, out double az, out double alt, out double airmass,out string ha, out int Limits);
    void Dither(double deltaRa,double DeltaDec);
    void GotoObject(string Name);
    void TrackPosition(double Ra2000, double Dec2000, double RaNonSiderealSpeed, double DecNonSiderealSpeed);
    void Sync(double Ra2000,double Dec2000);
    void SyncObject(string Name);
    void MapOnTarget(string PointingModelId);
    void MapOnPosition(double Ra1000, double Dec2000,string PointingModelId);
    void GetRealPosition(out double Axis1,out double Axis2);
    void GetRealTarget(out double Axis1,out double Axis2);
    void GetRealSpeed(out double SpeedAxis1, out double SpeedAxis2);
    void GetAltAz(out double Alt, out double Az);
    void GetAirmass(out double Airmass);
    void GetJd(out double JulianDate);
    void GetHourAngle(out string HourAngle);
    void GetLST(out string LocalSiderealTime);
    void SetPark();
    void GotoPark();
    void Stop();
    void GetObservatory(out double Lat,out double Lon, out double Alt);
    void GetMountType(out string MountType);
    void MeasurePosition(string PointingModelId,out double Ra2000, out double Dec2000);
    void MeasureTarget(string PointingModelId,out double Ra2000, out double Dec2000);
    void GetCurrentPointingModel(out string PointingModelIdentifier);
    void GetPointingModels(out StringVector PointingModels);
    void SetPointingModel(string PointingModelIdentifier);
    void GetTrackingOffset(out double Axis1,out double Axis2);
    void GetTrackingError(out double Axis1,out double Axis2);
    void GetTcsOffset(out double Axis1,out double Axis2);
    void Guide(double RaGuide,double DecGuide);
    void MountGuide(double Axis1,double Axis2);
    void SetControl(double KpA1,double KiA1,double KdA1,double Kp2A1,double Ki2A1, double Kd2A1,double KpA2,double KiA2,double KdA2,double Kp2A2,double Ki2A2, double Kd2A2,double SamplingTime,double MaxSpeed,double Control12Limit);
  };
```

```

//Define the pointing model class
interface PointingModel extends CoolObsPeripheral
{
  void SetModel(string PointingModelFile);
  void StartNewModel();
  void SaveModel();
  void GetObservations(out DoubleVector SkyPositionAxis1,out DoubleVector MountPositionAxis1,out DoubleVector SkyPositionAxis2,out DoubleVector MountPositionAxis2);
  void SetObservations( DoubleVector SkyPositionAxis1, DoubleVector MountPositionAxis1, DoubleVector SkyPositionAxis2, DoubleVector MountPositionAxis2);
  void SetRansacOptimization(int Active);
  void IsRansacOptimizationActive(out int Active);
  void GetParamNames(out StringVector ParamNames);
  void GetParamActive(out IntVector ActiveParams);
  void GetParamValues(out DoubleVector ParamValues);
  void SetParamActive(IntVector ActiveParams);
  void ComputeModel();
  void GetIdentifiersList(out StringVector IdentifiersList);
  void GetPlot(string Axis1Identifier,string Axis2Identifier,out DoubleVector Axis1Values,out DoubleVector Axis2Values);
  void CorrectPosition(double Axis1SetPoint, double Axis2SetPoint, out double CorrectedAxis1, out double CorrectedAxis2);
  void AddObservation(double SkyAxis1, double SkyAxis2, double RealAxis1, double RealAxis2);
};

//Define the rotator
interface Rotator extends CoolObsPeripheral
{
  void SetPositionAngle(double Position);
  void HomeRotator();
  void GetPositionAngle(out double Position);
  void GotoPositionAngle(double Position);
  void Track(double speed);
};


//Define Filter wheel objects
interface CoolObsFilterWheel extends CoolObsPeripheral
{
  void GetFilters(out StringVector Filters);
  void GetCurrentFilter(out string filter);
  void SetCurrentFilter( string filter);
};

//The callback line reciver is a callback which is sent for each read line from a camera object
interface CallbackLineReceiver
{
    void callback(ShortVector Line);
    void Abort();
};

//Define the generic camera object
interface CoolObsSimpleCamera extends CoolObsPeripheral
{
  void GetCameras(out StringVector Cameras);
  void GetCamerasSerials(out StringVector Serials);
  void Connect(string Camera, out short result);
  void GetCameraModel(out string CameraModel);
  void GetCameraSensor(out StringVector SensorsNames);
  void GetCameraSerialNumber(out string SerialNumber);
  void GetSensorSize(out short XSize, out short YSize);
  void GetPixelSize(out double PixelSizeX,out double PixelSizeY);
  void GetFullWellCapacity(out double FullWell);
  void GetMaxAdu(out long MaxAdu);
  void GetBinningModes(out short BinXMax,out short BinYMax);
  void GetBinning(out short BinX,out short BinY)  ;
  void GetGainModes(out StringVector GainModes);
  void GetCurrentGain(out double Gain);
  void GetImagingModes(out StringVector ImagingModes);
  void GetCurrentImagingMode(out string ImagingMode);
  void GetSubraster(out short XMin, out short YMin,out short Width,out short Height,out bool result);

  void GetTemperaturesNames(out StringVector TemperaturesSensors);
  void GetTemperatures(out DoubleVector Temperatures);
  void GetTemperatureSetPoint(out double SetPoint);
  void GetCameraProcessInfo(out double Percent);
  void GetCameraInfo(out string CameraInfo);
  void CanSetFilter(out bool value);
  void CanControlTemp(out bool value);
  void CanSetGain(out bool value);
  void CanSetSubraster(out bool value);
  void LockCameraSensor(bool value);
  void SetCameraSensor(string SensorName,out bool result );
  void SetBinning(short BinX,short BinY,out bool result ) ;
  void SetCurrentGain(string Gain);
  void SetCurrentImagingMode( string ImagingMode);
  void SetSubraster(short XMin, short YMin, short Width, short Height,out bool result);

  void SetTemperatureSetPoint(double SetPoint);
      void SetExpTime(double ExposureTime,out bool result);
  void StartExposureAsync(CallbackLineReceiver* proxy);
  void StopExposure( );
  void AbortExposure(string SensorName);
  void StartCooling(double Setpoint);
  void StopCooling();
  void StartWarmUp();
};

//Generate a coolobs camera which is a group of camera and filterwheel in the same server ( like a sbig camera)
interface CoolObsCamera extends CoolObsFilterWheel,CoolObsSimpleCamera
{
};

//Define the camera manager object
interface CoolObsCameraManager extends CoolObsCamera
{
  void CanFocus(out bool value);
  void CanCollim(out bool value);
  void TakeImage(string ImageName);
  void SetImageSave(bool SaveImageActive);
  void SetImagePath(string SaveMode, string ImagePath);
  void SetFlatTargetADU(double TargetADU);
  void SetAutoAstrometry(bool Active,string AstrometryAction);
  void SetDithering(double DitherValue);
  void SetExposureTime(double ExposureTime);
  void GetExposureTime(out double ExposureTime);
  void GetReadoutTime(out double ReadoutTime);
  void SetGuideDelay(double GuideDelay);
  void GetFocus(out double value);
  void SetFocus(double value);
  void AutoFocus();
  void AutoCollim();
  void StartGuiding();
  void StopGuiding();
  void PauseGuiding();
  void ResumeGuiding();
  void GetGuidingDataX(out DoubleVector GuidingDataX);
  void GetGuidingDataY(out DoubleVector GuidingDataY);
  void GetGuidingOffsetsX(out DoubleVector GuidingOffsetsX);
  void GetGuidingOffsetsY(out DoubleVector GuidingOffsetsY);
  void GetGuidingDataTime(out DoubleVector GuidingDataTime);
  void GetLastFlatParams(out double CorrectionFactor);
  void GetPointingModel(out string PointingModel);
  void GetDefaultMount(out string MountId);
  void GetAstrometryStatus(out string AstrometryStatus);

};

//The guider manager derivates from the camera manager and does not have extended functions yet
interface CoolObsGuiderManager extends CoolObsCameraManager
{


};

//Define the focuser object
interface Focuser extends CoolObsPeripheral
{
  void Home();
  void GetFocus(out double CurrentPosition);
  void GotoFocus(double FocusAbsolutePosition);
  void SetFocus(double FocusAbsolutePosition);
  void RelativeMoveFocus(double dFocus);
};

//define the rotofocuser, which has a rotator and a focuser in the same driver (like officina stellare rotofoc)
interface RotoFocuser extends Focuser,Rotator
{
};

//Define a roll off roof
interface Shed extends CoolObsPeripheral
{
  void Home();
              void IsHomed(out int HasHomed);
  void Open();
  void Stop();
  void Close();
};

//define a dome
interface Dome extends Shed
{
  void Sync(double Position);
  void GetPosition(out double Position);
  void Move(double Position);
};

//Define the hexapod object
interface Hexapod extends Focuser
{
  void Get3DPosition(out double PositionTip,out double PositionTilt,out double ShiftX,out double ShiftY,out double PositionFocus);
  void Goto3DPosition(double PositionTip,double PositionTilt,double ShiftX,double ShiftY,double PositionFocus);
};

//Define the micropap interface which connects up to 6 motors
interface Micropap extends CoolObsPeripheral
{
  int Home(int MotorIndex,int MaxTravel,string Direction);
  int MoveMotor(int MotorIndex, int numSteps, string Direction);
};

//Define the ipec server
interface Ipec extends CoolObsPeripheral
{
  void OpenLoop(int pwm);
  void Home();
  void IsHomed(out int Homed);
  void Sync(double Position);
  void ClearControl();
  void CloseLoopSpeed(float speed);
```

```
    void CloseLoopPosition(float Position);
    void SetP(double P);
    void SetI(double I);
    void SetD(double D);
    void SetPID(double P,double I ,double D);
    void Stop();
    void GetPosition(out double Position);
    void GetSpeed(out double Speed);
    void GetTrackingData(out DoubleVector Speed, out DoubleVector SetPoint, out
    IntVector Output, out DoubleVector Time, out DoubleVector Position,out
    DoubleVector AbsPosition,out DoubleVector SerialPosition,out DoubleVector
    PosSetPoint,out double Resolution,out double AbsResolution,out double Sampling)
    ;
    void StartRecord();
    void SetDoublePwmSpeedLimit(int limit)  ;

  };

  //Define the cloud sensor
  interface AAGCloudSensor extends CoolObsPeripheral
  {
    void GetAmbientTemp(out double temp);
    void GetRainSensorTemp(out double temp);
    void GetSkyTemp(out double temp);
    void GetRainFrequency(out double Freq);
    void GetIllumination();
  };

};
```

B Puc40 Electrical documentation

Documentación TCS PUC40.

Yerko Luco Gaudino.
Ingeniero en Electrónica.
Instituto de Astrofísica.
Facultad de Física.
Pontificia Universidad Católica de Chile.

October 18, 2017

1 Introducción

El siguiente documento tiene por objeto: describir el diagrama de las conexiones eléctricas del loop HA y DEC, el diagrama de red de los elementos, el sistema de potencia, las entradas-salidas del sistema y el mapping de los cables que vienen desde la montura.

2 Hardware

2.1 Loop HA y DEC

El sistema de control del telescopio o TCS está soportado por dos controladores IPECMOT 48/10 T1 del fabricante suizo FiveCo; uno para el loop de Ascención Recta (HA) y el otro para el loop de Declinación (DEC).

El loop HA está formado por un motor DC (HA MOT), un encoder de 8 líneas (HA ENC) y 2 limit switches (LS1A/LS1B). El loop DEC está formado por un motor DC (DEC MOT), un encoder de 4 líneas (DEC ENC), un conversor PC104 de 4 a 10 líneas para conectar el encoder al IPEC DEC (DEC PC104) y 2 limit switches (LS2A/LS2B).

Asimismo, los controladores están conectados en paralelo a una fuente de 24V y a un par de botones E-STOP que se conectan al pin ENABLE.

La figura 1 muestra los diagramas del loop HA y DEC. La figura 2 muestra el controlador IPEC 48/10 T1.

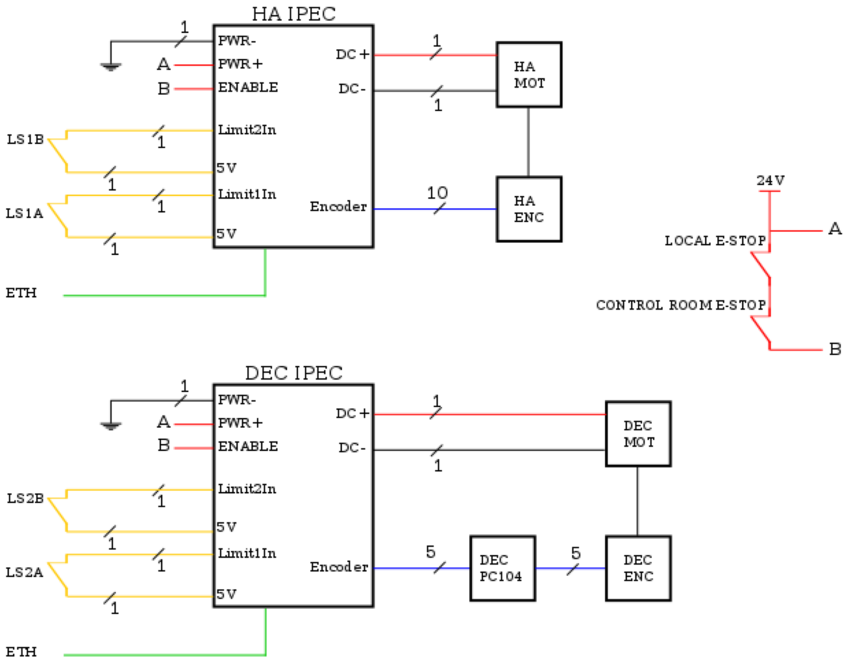


Figure 1: Diagrama TCS.

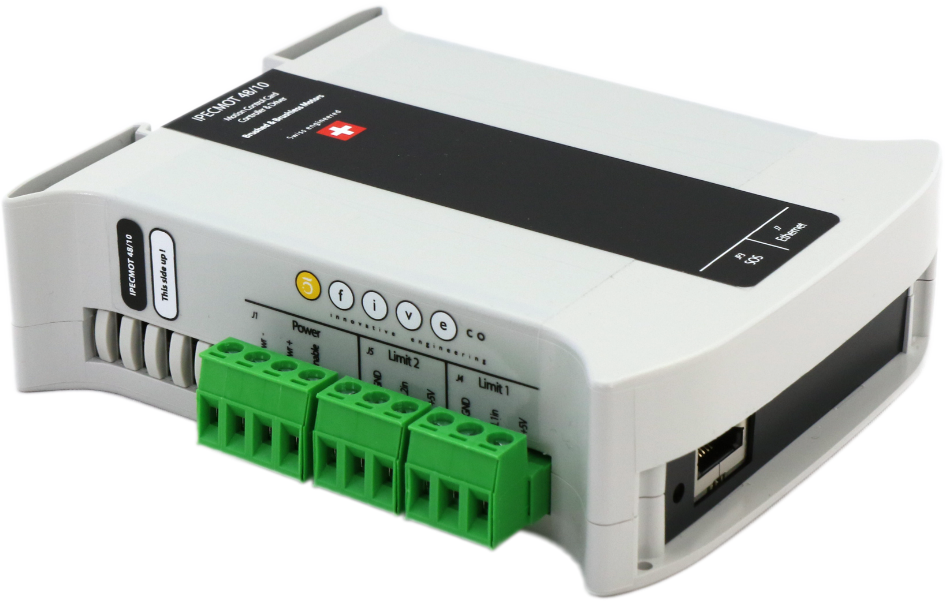


Figure 2: IPEC 48/10 T1.

2.2 TCS e Instrumentos

Cada loop de control está gobernado por un servidor Raspberry Pi (RPI HA y RPI DEC) que aloja el software del TCS donde se procesan los datos.

Estos datos son sincronizados por el servidor principal RPI TCS donde se aloja la interfaz de control que permite operar el telescopio.

Además del TCS, existen otros dispositivos dentro del sistema de control. Por un lado el PC NUC i3 (figura 5), aloja la cámara SBIG STL1001 y el controlador ROBOFOCUS en sus puertos USB. Por el otro, el RPI GUIDER aloja la cámara guía del telescopio, también conectada a un puerto USB. El NUC i3 y el RPI GUIDER junto con los servidores RPI TCS, RPI HA y RPI DEC y los controladores IPEC HA y IPEC DEC forman una red sobre ethernet donde cada uno tiene asignado un número ip fijo. Cada dispositivo se conecta a un switch ENCORE 10/100 de 8 puertos (figura 6) que permite la arquitectura de la red. La figura 3 muestra el diagrama de red del TCS, los números ip que fueron asignados y los dispositivos USB. La figura 4 muestra la tarjeta Raspberry Pi 3 utilizada en los servidores HA, DEC, TCS y GUIDER.

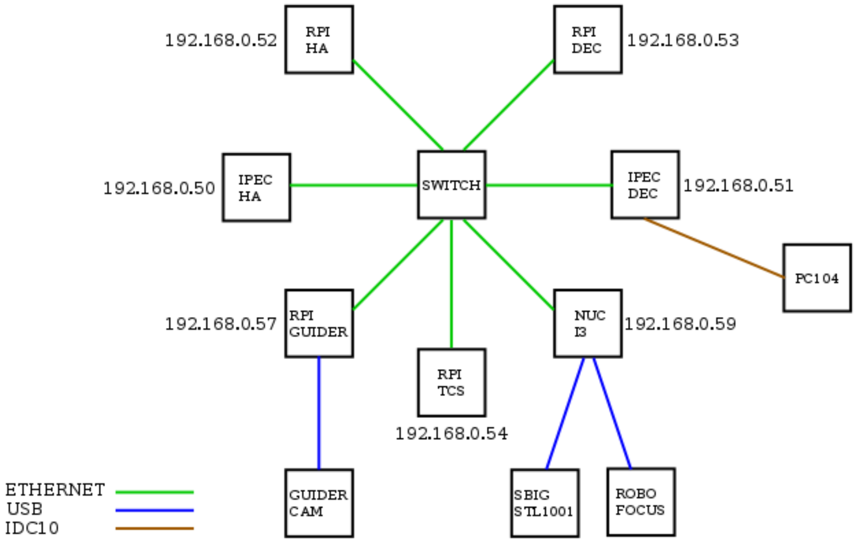


Figure 3: Diagrama de Red.

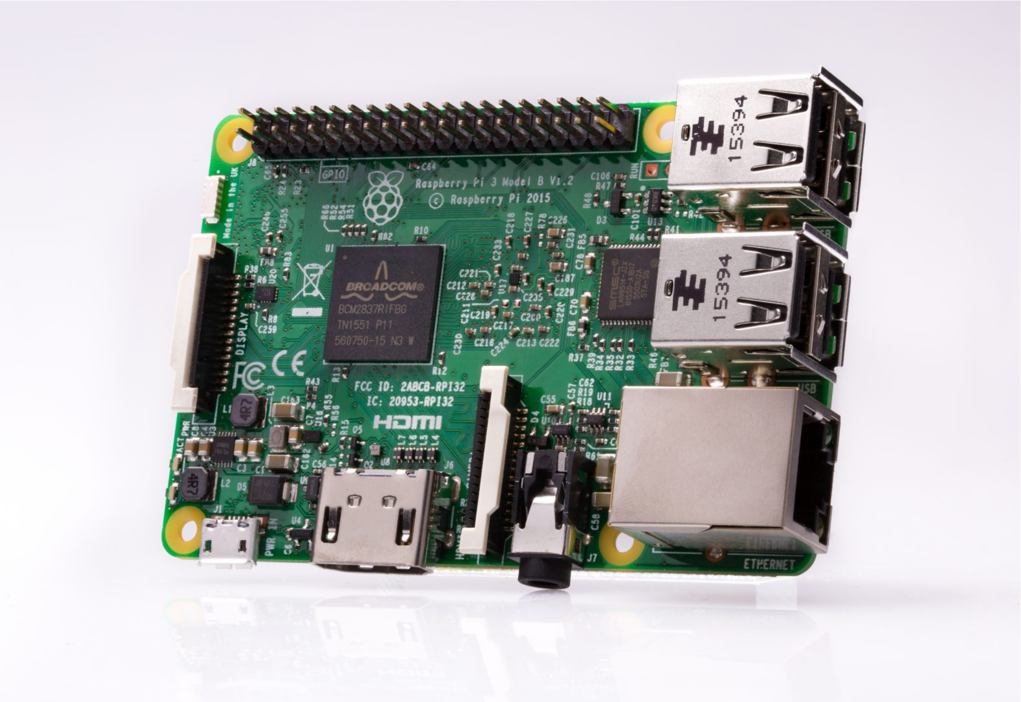


Figure 4: Raspberry Pi 3.

Figure 5: NUC i3.

Figure 6: Switch Encore ENH908NWY.

2.3 Potencias

La figura 7 muestra el diagrama de las potencias. Los controladores IPEC están alimentados por la fuente V1 (figura 8) de 24V/14.6A que proporciona potencia a los motores. El controlador ROBOFOCUS y el conversor PC104 están alimentados por la fuente V2 (figura 9) de 12V/10A. Los RPI TCS, HA, DEC, GUIDER y el switch ENCORE 8-Ports están alimentados por la fuente V3 (figura 10) de 5V/6A. Finalmente el computador NUC i3 está alimentado por la fuente VNUC de 19.5V.

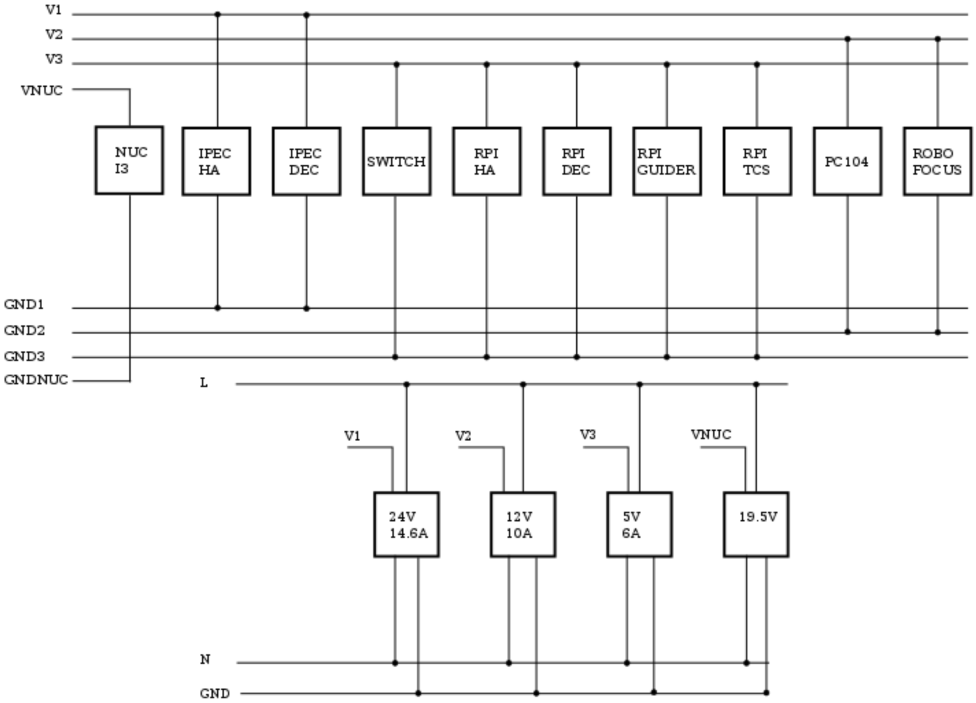


Figure 7: Distribución de las potencias.

Figure 8: Switching 24V/14.6A.

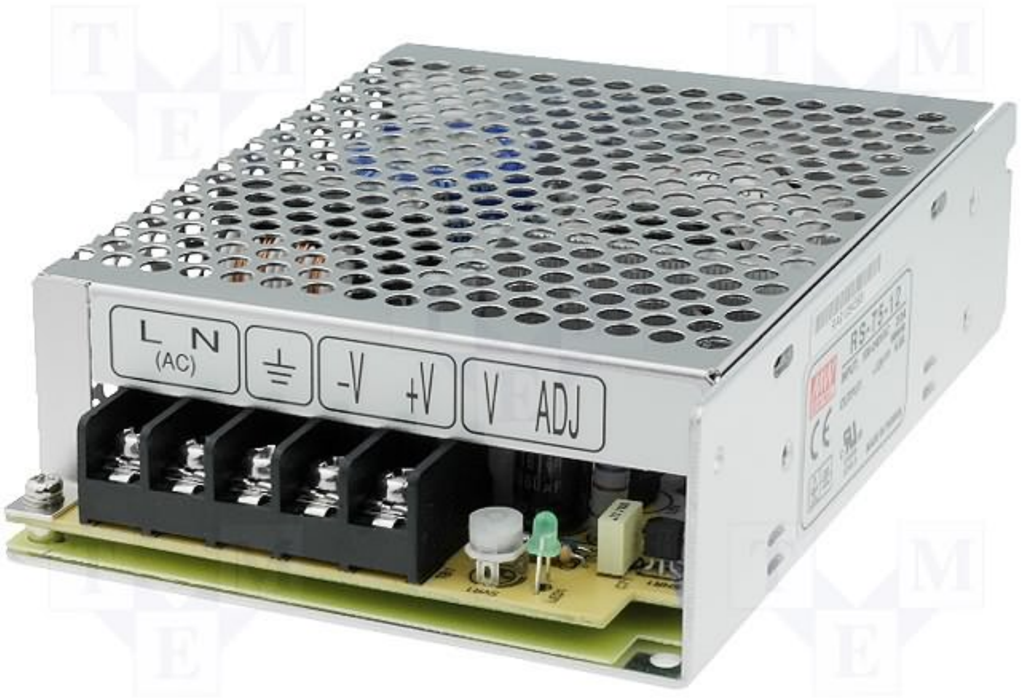


Figure 9: Switching 12V/10A.

Figure 10: Switching 5V/6A.

3 Entradas y Salidas.

3.1 Encoder DEC

El conector que soporta el encoder del lazo de declinación o DEC ENCODER es un conector circular de 10 pins o 10PCONN en la entrada del enclosure. El DEC ENCODER trabaja con 4 líneas: 5V-GND-A-B. Estas líneas llegan hasta un conector housing de 5 pins o IO5TCONN que permite la conexión a la tarjeta conversora PC104. La figura 4 muestra las líneas y los conectores que soportan las señales del DEC ENCODER.

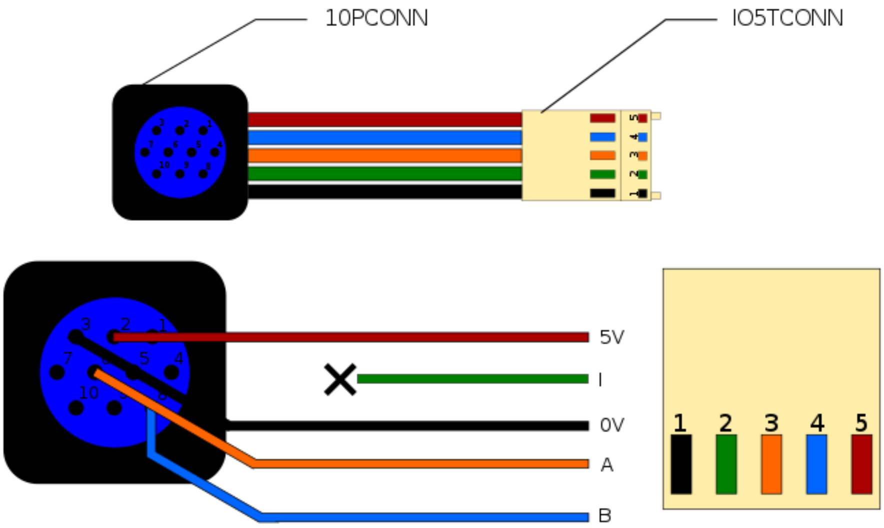


Figure 11: Entrada Encoder DEC.

3.2 Encoder HA

El conector que soporta el encoder del lazo de ascención recta o HA ENCODER es un conector circular de 10 pins o 10PCONN en la entrada del enclosure. El HA ENCODER trabaja con 8 líneas: 5V, GND, /A, /B, /I, A, B, I. Estas líneas llegan a un conector IDC10P que permite la conexión al controlador IPEC HA. La figura 5 muestra las líneas y los conectores que soportan las señales del HA ENCODER.

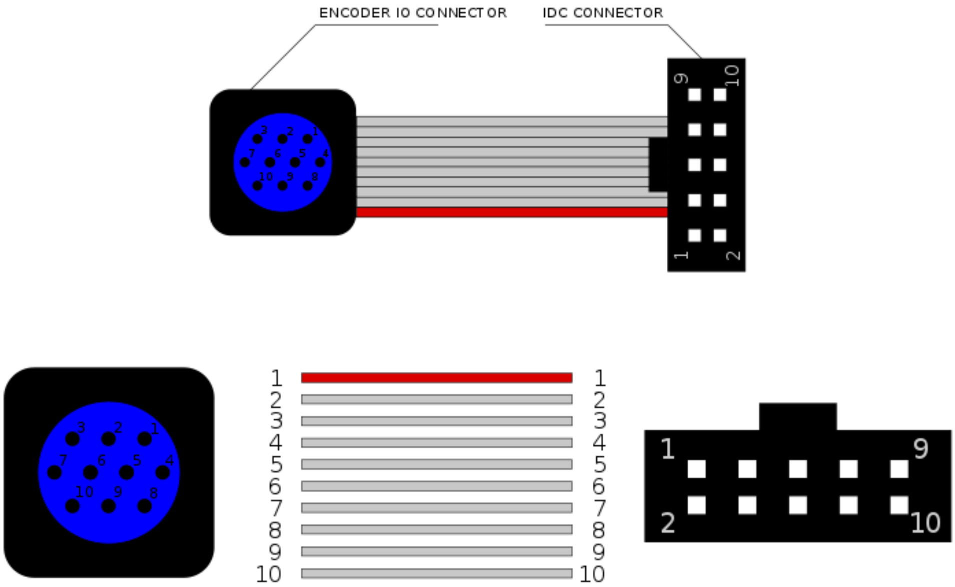


Figure 12: Entrada Encoder HA.

3.3 E-STOP

La parada de emergencia o E-STOP es un limit switch N.C. que hace corto circuito entre el POWER+ pin y el ENABLE pin. Cuando este se activa, el limit switch abre el circuito y el ENABLE pin queda en 0V e inmediatamente el motor deja de funcionar. El sistema utiliza una rama de dos E-STOP en serie, uno local y otro en la sala de control. Esta rama junto con los pins ENABLE de ambos IPEC forman un nodo B, haciendo posible que ambos motores se detengan al activar cualquiera de los dos E-STOP (figura 1). En la figura 6 se muestran las líneas y los conectores que soportan las señales entre los E-STOPs y los IPEC.

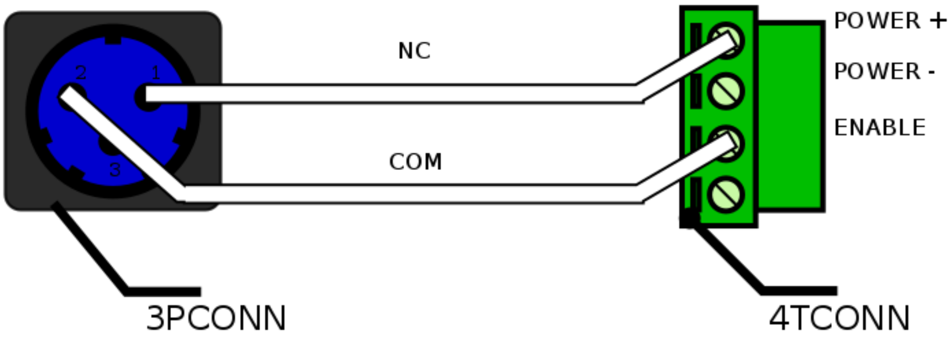


Figure 13: Entrada E-STOP.

3.4 Limit Switch

El conector que soporta las señales de los limit switch es un conector circular de 5 pins o 5PCONN. Las señales llegan hasta una regleta de 3 terminales o 3TCONN que permite la conexión al controlador IPEC. Cada IPEC soporta hasta 2 limit switch de entrada, lo que hace un total de 4 entradas de limit switch. La figura 7 muestra las líneas y los conectores que soportan las señales de cada limit switch que llegan a los IPEC.

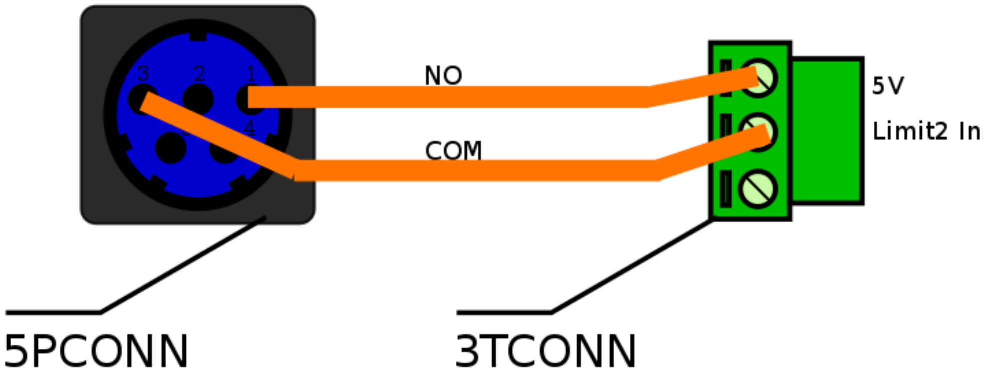


Figure 14: Entrada LS.

3.5 Motor DC

El conector que soporta las señales de salida de los motores en la montura es un conector circular de 7 pins o 7PCONN. Las señales se transmiten desde una regleta de 8 terminales o 8TERCONN en los controladores IPEC. Luego viajan por un par de conductores AWG22 y llegan finalmente al conector 7PCONN. La figura 8 muestra las líneas y los conectores que soportan las señales de salida de los motores dc.

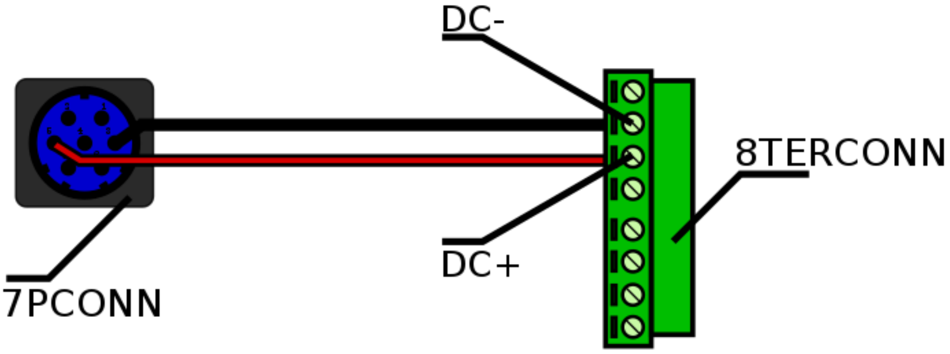


Figure 15: Salida MOTOR.

3.6 Robofocus

El conector que soporta las señales de salida del motor ROBOFOCUS es un conector circular de 7 pins o 7PCONN. Las señales se transmiten desde un conector DB9 en el ROBOFOCUS hasta el conector de salida 7PCONN. La figura 9 muestra las líneas y los conectores que soportan las señales de salida del ROBOFOCUS.

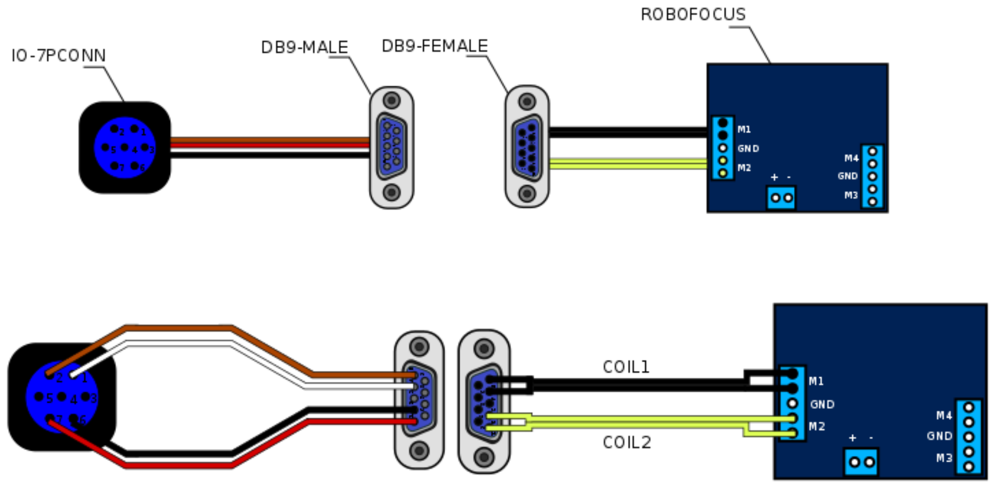


Figure 16: Salida ROBOFOCUS.

4 Montura.

4.1 Encoder DEC Input

El encoder DEC sólo trabaja con 4 líneas: 5V-0V-A-B. Estas líneas viajan sobre un cable de teléfono hasta la parte trasera de la montura. Luego se extienden con un cable CAT5 hasta el conector 10PCONN que se conecta en el enclosure. La figura 10 muestra las líneas y el conector que soportan las señales del encoder DEC.

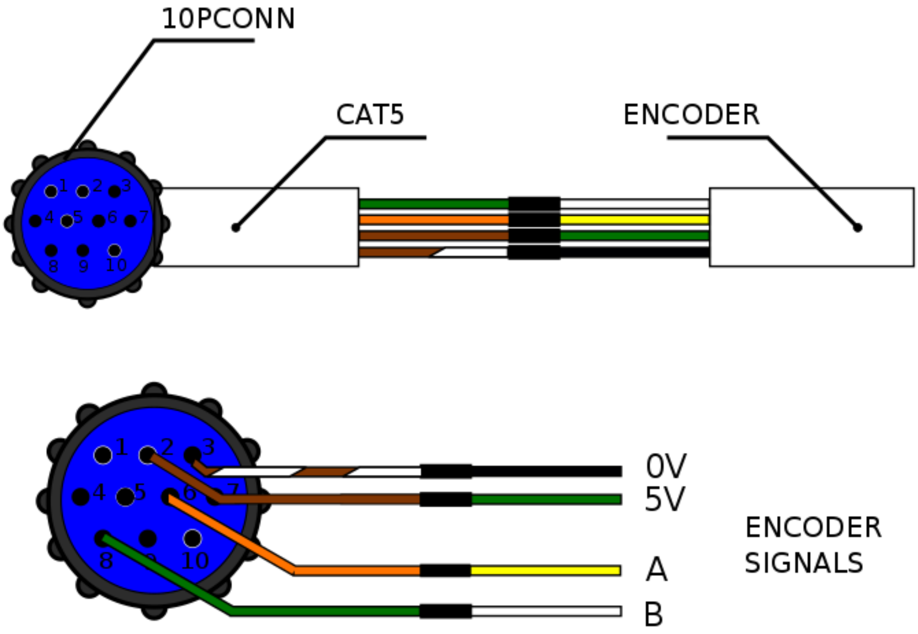


Figure 17: Cable Encoder DEC.

4.2 Encoder HA Input

El Encoder HA trabaja con 8 líneas: 5V-0V-A-B-I-/A-/B-/I. Estas líneas viajan sobre una extensión de cable CAT5 que llega a un conector 10PCONN que se conecta al enclosure. La figura 11 muestra las líneas y el conector que soportan las señales del encoder HA.

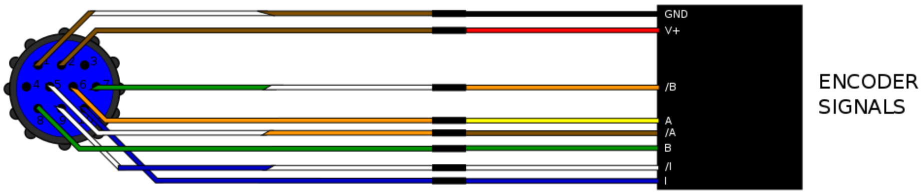


Figure 18: Cable Encoder HA.

4.3 E-STOP Input

El telescopio tiene dos botones de pánico o E-STOP: uno local y otro en la sala de control. Estos están conectados en serie, de manera que cualquiera que se active detenga los motores. La figura 12 muestra el conector que llega desde los E-STOP. Se puede notar que estos son pulsadores N.C.

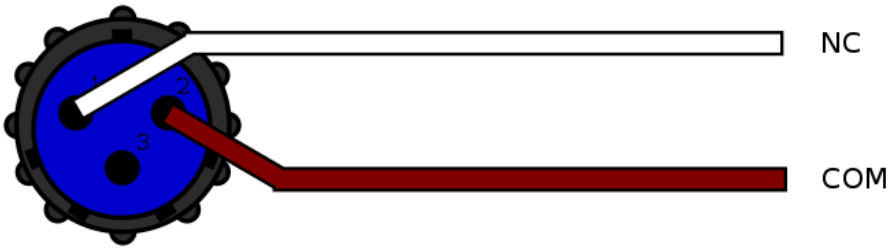


Figure 19: Entrada E-STOP.

4.4 Limit Switch Input

El telescopio tiene 4 limit switches, 2 por cada eje. Esto permite limitar el rango del desplazamiento en ellos. Cada limit switch se conecta al TCS mediante un conector circular de 5 pins o 5PCONN. La figura 13 muestra las líneas y el conector que soportan las señales de los limit switches.

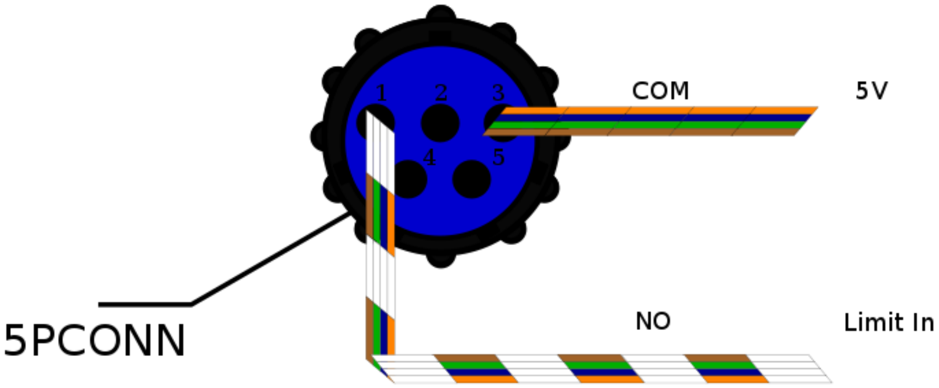


Figure 20: Entrada LS.

4.5 Motor Output

La montura trabaja con 2 motores: uno para Ascensión Recta o HA y otro para Declinación o DEC. Cada motor se conecta al TCS mediante un conector circular de 7 pins o 7PCONN. La figura 14 muestra las líneas y el conector que soportan las señales de los motores.

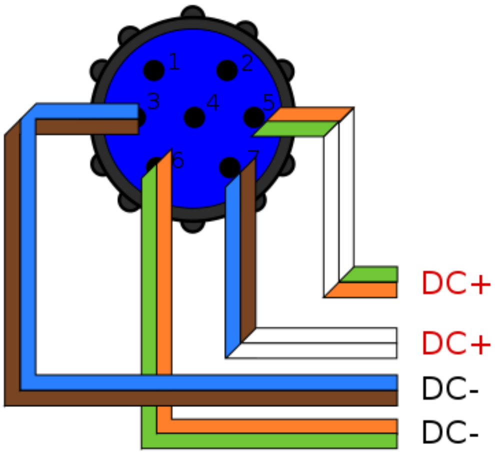


Figure 21: Entrada MOTOR.

4.6 Robofocus Output

El sistema ROBOFOCUS controla un motor stepper que permite modificar la posición del foco en el telescopio. Este motor se conecta al TCS mediante un conector circular de 7 pins o 7PCONN. La figura 15 muestra las líneas y el conector que soportan las señales del motor steper del focuser.

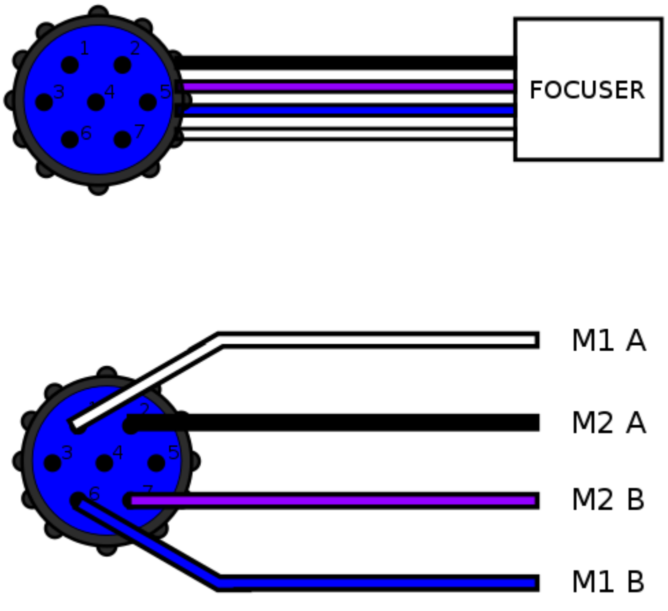


Figure 22: Entrada ROBOFOCUS.

C Eso50 Electrical documentation

CONEXIONES ELECTRICAS ESO50TCS

Yerko Luco
Ingeniero en Electrónica
Observatorio Docente
Pontificia Universidad Católica de Chile
yluco@astro.puc.cl

Descripción

El siguiente documento tiene por objeto describir las conexiones eléctricas internas dentro del sistema de control TCS del telescopio ESO50.
La Figura 1 muestra la electrónica utilizada por el TCS, la conexión de los cables y los conectores de las entradas y salidas.

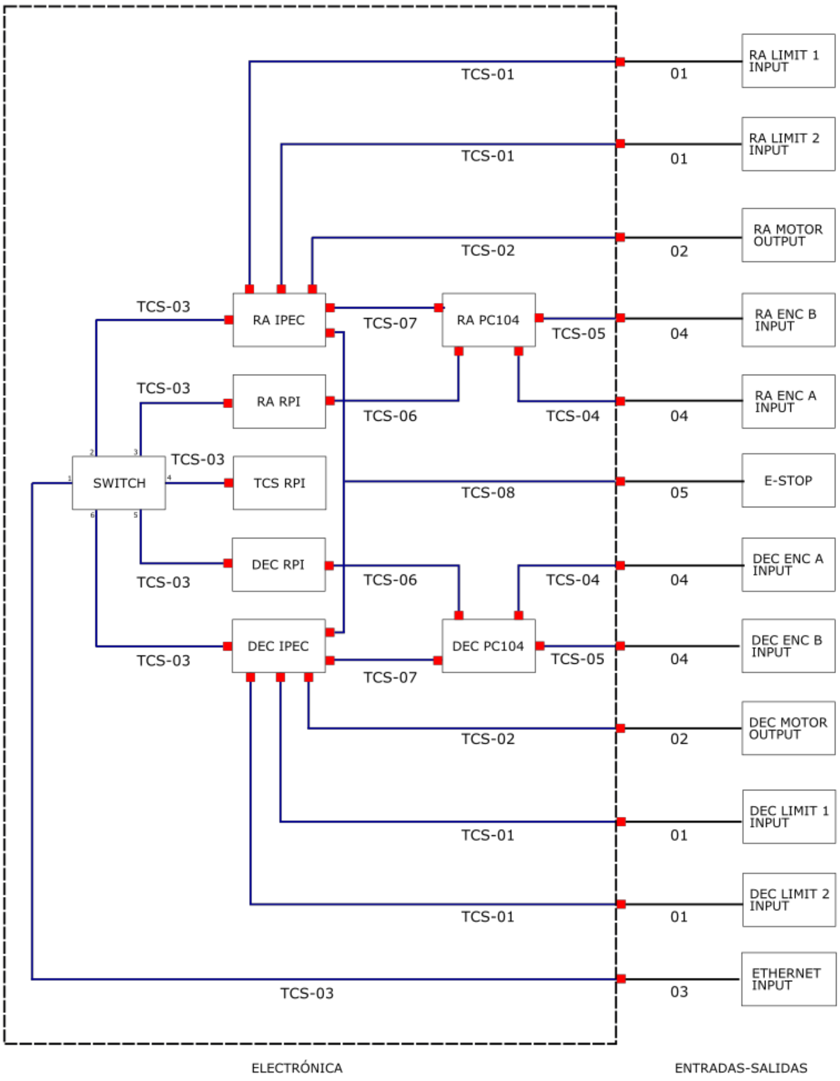


Figura 1. Diagrama en bloques.

Cables TCS

Los cables internos de la caja están etiquetados de la siguiente manera:

TCS-01	Estándar Limit Switch – IPEC.
TCS-02	Estándar Motor – IPEC.
TCS-03	Estandar TIA-T568B para redes LAN.
TCS-04	Estándar Encoder A.
TCS-05	Estándar Encoder B.
TCS-06	Conversor USB - RS232.
TCS-07	Estándar IDC10P para la conexión IPEC-PC104.
TCS-08	Estándar E-STOP – IPEC.

Tabla 1. Cables internos del TCS.

TCS-01

El cable TCS-01 conecta los Limit Switches del Telescopio al controlador IPEC. La tabla 2 muestra los materiales para la construcción del cable. La figura 3 muestra el detalle del cable TCS-01.

Detalle	**Proveedor**
Conector Circular 5 pines.	Victronics.
Cable hook-up Blanco.	Victronics.
Regleta 3T IPEC MOT.	Victronics.

Tabla 2. Materiales TCS-01.

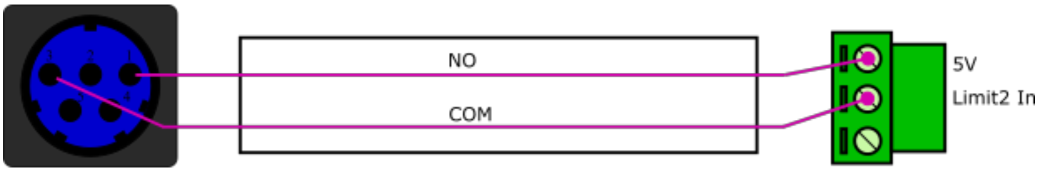


Figura 3. Estándar TCS-01.

TCS-02

El cable TCS-02 conecta los motores DC de RA y DEC del Telescopio a los controladores IPEC RA y DEC. La tabla 3 muestra los materiales para la construcción del cable. La figura 4 muestra el detalle del cable TCS-02.

Detalle	**Proveedor**
Conector Circular 10 pines.	Victronics.
Cable hook-up negro.	Victronics.
Cable hook-up rojo.	Victronics.
Regleta 3T IPEC MOT.	Victronics.

Tabla 3. Materiales TCS-02.

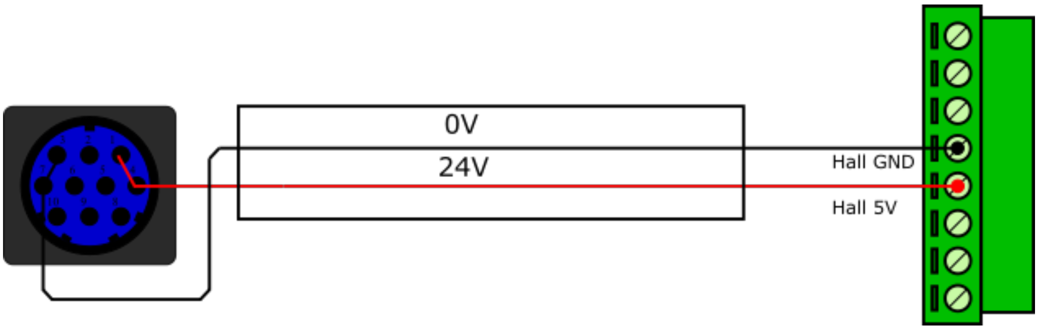


Figura 4. Estándar TCS-02.

TCS-03

El cable TCS-03 conecta la entrada de la red del tablero al switch del TCS . La tabla 4 muestra los materiales para la construcción del cable. La figura 5 muestra el detalle del cable TCS-03.

Detalle	**Proveedor**
Cable de Red.	PC-Factory.
Conector RJ45.	PC-Factory.

Tabla 4. Materiales TCS-03.

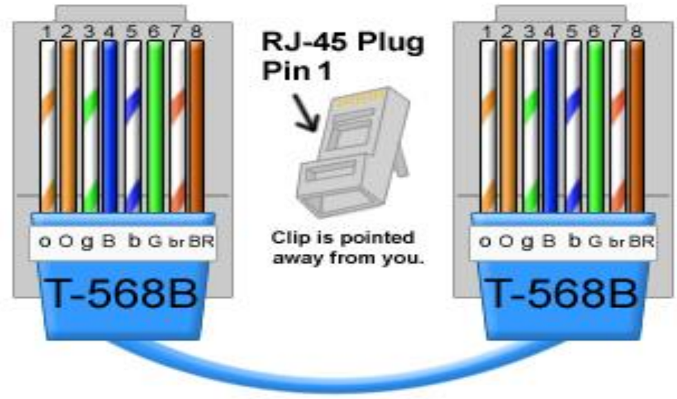


Figura 5. Estándar TCS-03.

TCS-04

El cable TCS-04 conecta los encoders principales desde la entrada del tablero hasta la tarjeta PC-104 . La tabla 5 muestra los materiales para la construcción del cable. La figura 6 muestra el detalle del cable TCS-04.

Detalle	**Proveedor**
Conector Circular 7P.	Victronics.
Cable hook-up negro.	Victronics.
Cable hook-up verde.	Victronics.
Cable hook-up amarilllo.	Victronics.
Cable hook-up azul.	Victronics.
Cable hook-up rojo.	Victronics.
Conector PCB 5T.	Victronics.

Tabla 5. Materiales TCS-04.

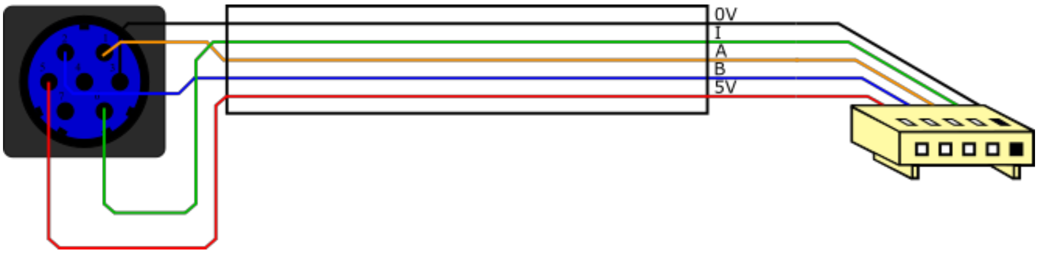


Figura 6. Estámdar TCS-04.

TCS-05

El cable TCS-05 conecta los encoders secundarios desde la entrada del tablero hasta la tarjeta PC-104 . La tabla 6 muestra los materiales para la construcción del cable. La figura 7 muestra el detalle del cable TCS-05.

Detalle	**Proveedor**
Conector Circular 7P.	Victronics.
Cable plano 10P.	Victronics.
Conector IDC10T.	Victronics.

Tabla 6. Materiales TCS-05.

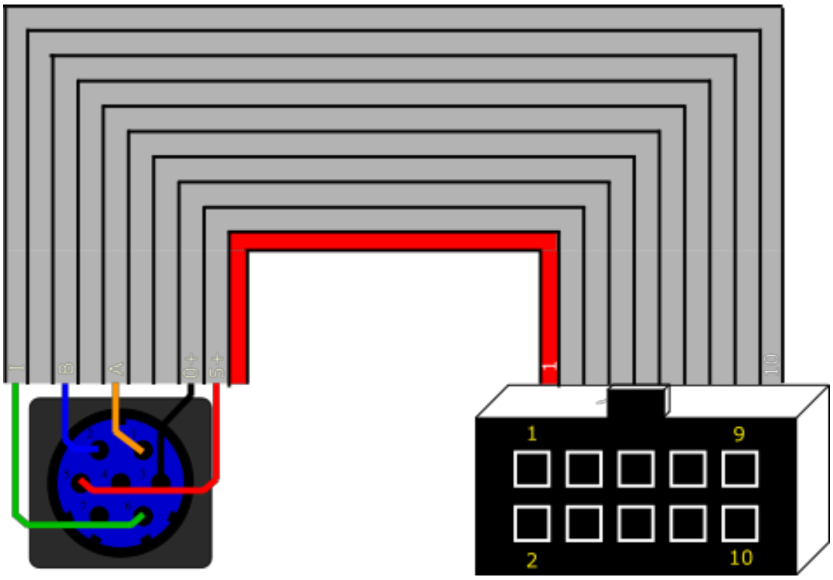


Figura 7. Estándar TCS-05.

TCS-06

El cable TCS-06 conecta los encoders secundarios desde la tarjeta PC-104 RA y PC-104 DEC hasta los Raspberry PI RA y Raspberry Pi DEC que los leen. La tabla 7 muestra la descripción y el proveedor del cable. La figura 8 muestra el detalle del cable TCS-06.

Detalle	Proveedor
Conversor USB-RS232.	PC-Factory.

Tabla 7. Materiales TCS-06.

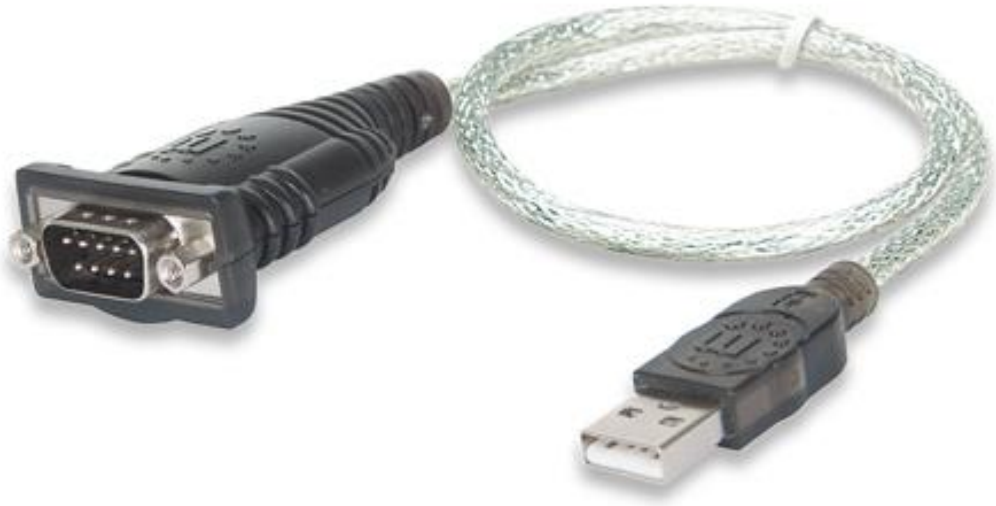


Figura 8. Estámdar TCS-06.

TCS-07

El cable TCS-07 conecta los encoders primarios desde las tarjetas PC-104 RA y PC-104 DEC hasta los IPEC RA e IPEC DEC . La tabla 8 muestra los materiales para la construcción del cable. La figura 9 muestra el detalle del cable TCS-07.

Detalle	**Proveedor**
Cable plano 10P.	Victronics.
Conector IDC10T.	Victronics.

Tabla 8. Materiales TCS-07.

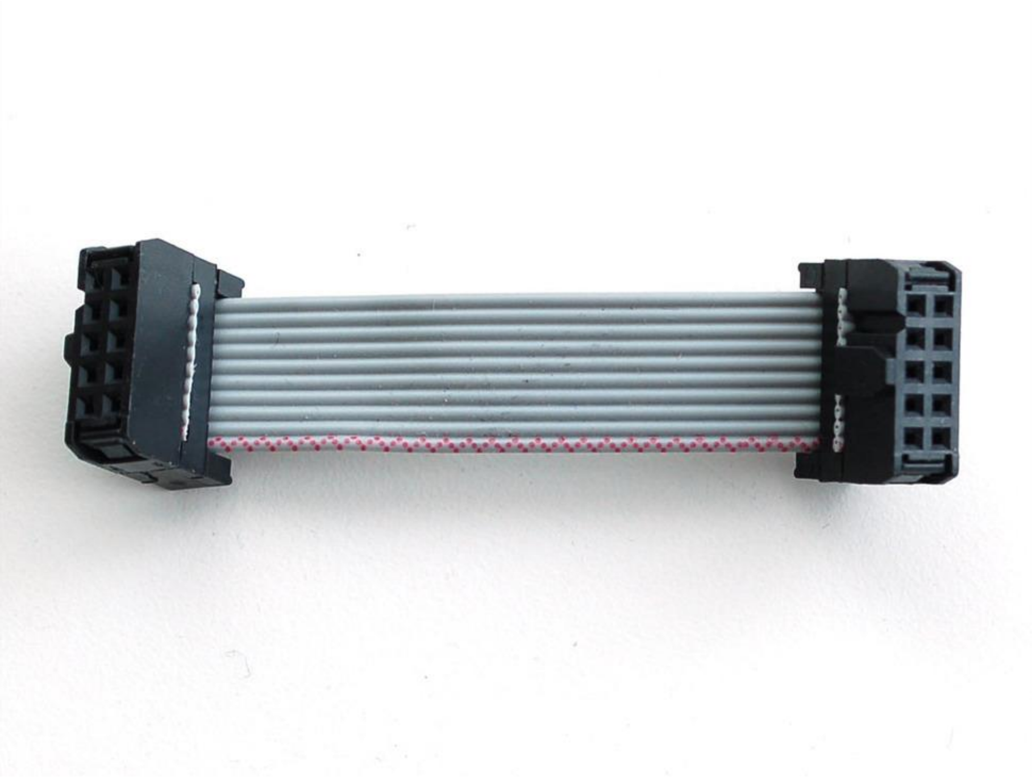


Figura 9. Estámdar TCS-07.

TCS-08

El cable TCS-08 conecta los E-STOP desde la entrada del tablero hasta los IPEC RA e IPEC DEC. La tabla 9 muestra los materiales para la construcción del cable. La figura 10 muestra el detalle del cable TCS-08.

Detalle	**Proveedor**
Conector Circular 3P.	Victronics.
Cable hook-up Naranjo.	Victronics.
Regleta 3T.	Victronics.

Tabla 9. Materiales TCS-08.

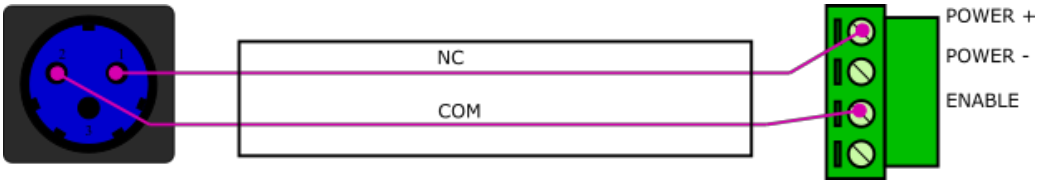


Figura 10. Estándar TCS-08.