Tuesday 9th October, 2012

S-SST

Small Scharmer Solar Telescope at
Vetenskapens Hus / Stockholm House of Science
Master's thesis

Henrik Åkerstedt
Project supervisors: Peter Dettori and Magnus Näslund

VETENSKAPENS HUS
AlbaNova universitetscentrum, 106 91 Stockholm
08 - 553 784 60, vh@vetenskapenshus.se, www.vetenskapenshus.se

KTH VETENSKAP OCH KONST

Stockholms universitet

Abstract

The S-SST is a new solar telescope at *Vetenskapens Hus* / Stockholm House of Science. During the past year it has been assembled and programmed to track The Sun or other celestial objects. The motion control was the most crucial part of the project, that is to implement and tune the servo system. The goal was to have smaller position errors than the maximum resolution of the telescope (estimated to be 2 arcseconds).

This was solved using a cascade PI control loop and a custom manual tuning method which resulted in an rms position error of less than 0.2 arcseconds. So the motors contribution to inaccuracies is negligible compared to the diffraction limit of the lens and the astronomical seeing.

Contents

1 Introduction **6**

I Project description **6**

2 Background **7**

3 Hardware description **9**

3.1 Legacy; the Swedish Solar Telescope (SST) on La Palma 9

3.2 The S-SST at Vetenskapens Hus 10

3.3 Detailed hardware description 10

4 Software description **13**

II Technical description **13**

5 Turret cabling **13**

6 Turret electronics enclosure **15**

7 Control room **16**

7.1 CAN 16

7.2 APCI counter 16

7.3 Watchdog 16

7.4 Handpaddles 16

7.5 Software structure . . . 17
7.6 Service programs . . . 17
7.7 Turret program . . . 18

8 Endstop design **18**

III Servo optimization **19**

9 Project goal **19**

10 Basic control theory **20**
10.1 Open-loop . . . 20
10.2 Feedback loop . . . 20
10.3 Feed forward . . . 21

11 PID **21**
11.1 Proportional . . . 21
11.2 Integral . . . 22
11.3 Derivative . . . 22
11.4 Cascade PID . . . 23

12 S-SST servo system description **23**

13 Optimization process **24**
13.1 General PID tuning techniques . . . 24
13.2 Cascade PID tuning . . . 25
13.3 S-SST servo tuning procedure . . . 25

14 Servo tuning process **26**
14.1 Velocity tuning . . . 26
14.2 Position tuning . . . 28

14.3 Servo evaluation 28

15 Conclusion **30**

16 Possible servo improvements **31**

16.1 Gain scheduling 31

16.2 PDFF 31

16.3 Current control 32

17 Outlook **32**

17.1 Mirror alignment 32

17.2 Motor upgrades 32

17.3 Optics setup 32

17.4 Endstops and watchdog 33

17.5 Lenscap 33

17.6 Manuals 33

17.7 Educational use 33

18 Final words **34**

Appendix A Personal contribution **36**

Appendix B Endstop reset procedure **37**

Appendix C Photos **38**

Acknowledgements

First off, I'd like to thank Peter Dettori for being so patient and for teaching me good coding practice. His eerie ability to sniff out elusive bugs saved me countless hours of debugging.

Secondly, I thank Magnus Näslund for (among other things) getting me involved in this amazing project in the first place. Both Magnus and Peter were also an invaluable asset when writing this report.

I also thank Göran Scharmer whose expertise on telescope design and solar physics came in very handy indeed. His well organized archive was an invaluable asset during the whole project.

Thank you Johan and Bertil Pettersson at Digitalmekanik and Klas Bjelksjö for designing and manufacturing such a beautiful piece of hardware that became the S-SST.

An extended thanks to everyone at Vetenskapens Hus. No matter when or why I could always find someone to help me, if only to hold an umbrella during rainy days on the roof (see figure 19 in appendix C).

To the people at Industrisupport; thanks for going to work on a holiday to fetch us a glass cleaning compound for the mirror gluing that otherwise had to be postponed made my life a lot more stressful.

Finally thanks Sofia Häggvik for letting me be at work for 12 hours a day as the transit of Venus drew closer and closer.

1 Introduction

The solar telescope at *Vetenskapens Hus* / Stockholm House of Science[1] is a 20 cm solar telescope heavily based on the one meter Swedish Solar Telescope (SST) on La Palma, both hardware- and software-wise. It has been given the name S-SST (Small Scharmer Solar Telescope) after its chief designer; prof. Göran Scharmer.

The main goal of the project was to define and implement a servo system for controlling the telescope accurately enough to allow for solar observation. The accuracy and precision highly depends on the servo system (as well as the hardware and optics), so finding optimal servo parameters was a crucial part of the project.

Although tuning of the servo system was the main goal, mounting it on the roof of *Vetenskapens Hus* and calibrating the optics was of course also highly desired. On February 2012 it came to my attention that Venus would transit the Sun on the 6:th of June, so that was set as a deadline to have the S-SST operational[2]. The project then slightly changed from mostly software development to hardware manufacturing, cabling and project management for getting the turret mounted on the roof.

The deadline was met, but due to obscuring foliage and bad weather the transit was only observed for two seconds. After the transit, software work resumed as well as fine tuning the motors and adding safety routines.

The thesis is divided into three main parts.

Part I
: Background information to the S-SST as well as a brief hardware and software description

Part II
: Detailed technical information describing the turret mechanics, the motors and sensors , all the cabling and the control room set-up

Part III
: A brief introduction to control theory, a description of the S-SST servo system and the tuning process as well as an evaluation of its performance

[1] `www.vetenskapenshus.se`
[2] It only had to *work*, it didn't have to work well

Part I
Project description

2 Background

This project has a long history tied with Vetenskapens Hus, the Institute for Solar Physics [1] and the Department of astronomy at AlbaNova, Stockholm University [2]. Below is a brief time-line presenting the main stages of development of the S-SST. Appendix A contains a more detailed account of the progress during the past year.

1985-2000
The 50 cm Swedish Vacuum Solar Telescope (SVST) is operational

1998
Göran Scharmer, Bertil Pettersson and Klas Bjelksjö began designing the SST

March 2 2002
The 1 m Swedish Solar Telescope (SST) is inagurated

2006-2008
Procurement of mechanical and optical components for the S-SST

2010-2011
Mounting hole drilled in roof and platform assembled at Vetenskapens Hus

2010-2011
Preparations and procurement of the control system hardware

Oct 2011
The author's first involvement in the project

Dec 2011
In-lab assembly of the turret began

Apr 2012
Turret tracking the sun for the first time (in lab)

May 25 2012
Turret lifted and mounted on the roof of Vetenskapens Hus

June 3-6 2012
First light[3] occurred three days ahead of Venus transit

[3] The first time a telescope, or instrument, is opened up and directed at the sky

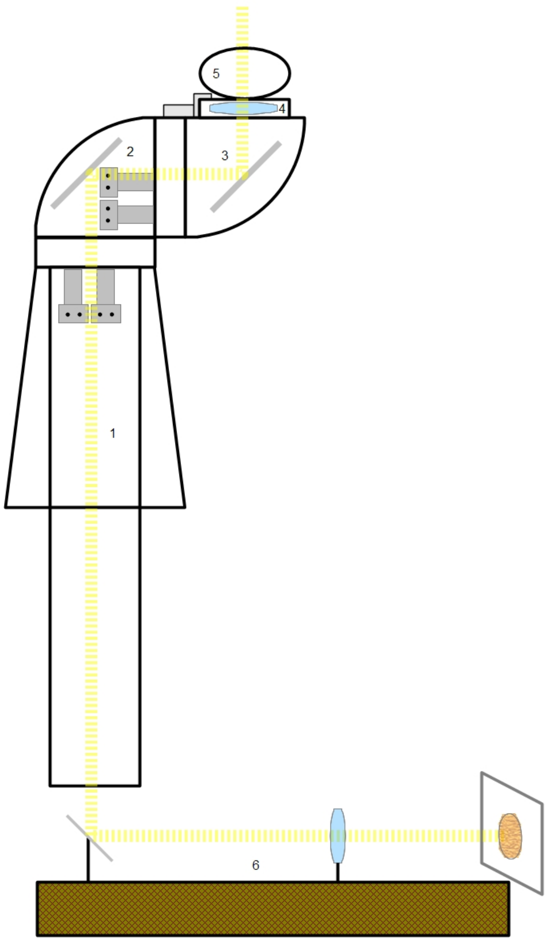


Figure 1: Turret schematic, see table 1 for explanation

The main purpose of the S-SST is observing the sun by means of CCD imaging and spectral analysis for educational purposes. The S-SST can be used both in existing school programs (including sunspot and sun spectrum observation) and more advanced high-school projects. It will also play an important role in teacher education and possibly public outreach.

The design is similar to that of a periscope with a lens mounted on a rotating turret with two degrees of freedom. Light is led through the turret via two slanted mirrors and a tube down to the control room where the light is projected along a granite bench. A schematic drawing of the telescope can be found in figure 1.

As the sun moves across the sky, the telescope needs to precisely follow its apparent motion. This motion, due to the Earth's rotation, can be predicted based on the time of day (moving from east to west), time of year (higher in the summer) and the location of the telescope (larger seasonal variations further away from the equator).

1	Turret base and beampipe
2	Azimuth mirror housing (rotates horizontally)
3	Elevation mirror housing (rotates vertically)
4	Primary lens (⌀= 200 mm, f = 6 m)
5	Protective lenscap (half opened position)
6	Optics bench with 45° mirror, lens and screen

Table 1: Turret schematic legend

The sun is, in most respects, an unexceptional star. But its proximity to Earth (while being ideal in most aspects) poses a problem for most telescopes. It is 10^{10} times brighter than the second brightest star and will destroy most telescopes.

A solar telescope is designed to handle the heat generated by the focused sunlight and to minimize the disturbances caused by the hot air. An advantage is that very short exposure times are needed and, since solar filters[4] are generally used, the optical setup doesn't have to be in a darkroom.

As is common for larger telescopes, the S-SST is alt-azimuth [5] mounted requiring precise motion control to track celestial objects. Smaller "hobby" telescopes and older professional ones (e.g. the Hale telescope) use an equatorial mount where one axis is parallel to the Earth's axis of rotation. Ideally, these only require a fixed speed motor on that axis to follow the apparent motion of the stars. This design is impractical for larger telescopes and with the computing power available, precise motion control is no longer a problem.

Each axis has two motors attached, and are connected to motor controllers. These have built-in sensors for detecting motion but the turret position is measured by a very accurate position sensor ring on each axis giving sub-arcsecond[6] position resolution.

The position is controlled in two layers forming a cascade control-loop (described in detail in section 12): a fast velocity loop governed by the motor controller ensures that the motors are running at a stable velocity. This velocity loop, in turn, gets it's command velocity from a slower software position loop that uses astronomical data to compute the desired position.

3 Hardware description

3.1 Legacy; the Swedish Solar Telescope (SST) on La Palma

The new S-SST at Vetenskapens Hus borrows much of its design from the one meter SST which, in turn is based on the 50 cm SVST (Swedish Vacuum Solar Telescope) inspired by the turret of DST (Dunn Solar Telescope) on Sacramento Peak, New Mexico.

The SST, being the second largest optical refracting telescope in use in the world, is to date the most highly resolving solar telescope [3][4] due to its location (Roque de los Muchachos Observatory, La Palma), its adaptive optics system and the fact that it is a vacuum telescope. [5]

[4] Filters that remove all but (typically) 1/10 000:th of the incoming light
[5] The axes are perpendicular with one horizontal and one vertical
[6] $\frac{1}{360}$ revolution = 1 degree = 60 arcminutes = 3600 arcseconds

It was conceptualised by Göran Scharmer, mechanical design by Johan and Bertil Pettersson and Klas Bjelksjö of Stockholms Digitalmekanik AB and software design by Pete Dettori among others. These people have also been involved in the design and construction of the S-SST.

Figure 2: SST

3.2 The S-SST at Vetenskapens Hus

The S-SST is, in short, a $\frac{1}{5}$ scale version of the SST with a lot of modifications. It was re-drawn from scratch and manufactured by Stockholms Digitalmekanik AB. Magnus Näslund did much of the conceptual work on fitting the telescope to Vetenskapens Hus as well as optics design and hardware purchase.

The S-SST consists of a 4 meter beam pipe with a stiff mounting plate on top. The pipe optically connects the telescope to the control room and optics bench. On top of the pipe sits the azimuth drive housing, azimuth mirror holder, elevation drive housing, elevation mirror holder, lens and lenscap as shown in figure 1.

So, the light enters through the lens, gets reflected on the two flat mirrors that revolve around two perpendicular axes and enters the control room and optics lab one floor below.

3.3 Detailed hardware description

The turret requires precise motion control as explained in section 2. The following section explains the different hardware components that are involved in motion control.

A ring gear attached to the top of each drive housing is driven by two Maxon [6] brand *EC-max 40* 70 W brushless motors. The motors are internally geared down to a ratio of 1:936 which combines with an external gear ratio of 1:9 to a 1:8424 gear ratio.

Motion feedback is provided from an internal Hall sensor as well as a Scancon [7] incremental encoder providing 2000 pulses per revolution. Each motor along with the two sensors are connected to a Maxon [6] *EPOS2 24/5* programmable motor controller.

The encoder acts as a motion sensor, giving 2000 indistinguishable electrical pulses per revolution to the motor controller. Since all pulses are identical, it only senses movement (and direction), not its absolute position. This is called an incremental encoder and needs a counter and a known starting position to determine its position. Since the encoder is used as a velocity sensor, absolute position is not required.

(a) EC-Max motor (b) Scancon encoder (c) EPOS2 24/5 motor controller

Figure 3: Motor components

On top of the azimuth drive housing sits a *SIGNUM* position encoder from Renishaw [8] made up of a *RESM* 300 mm diameter position encoder ring. The ring has a 20 µm scale on the periphery giving 47 200 tics per revolution. A *SR* readhead on the non-moving part of the drive housing optically reads the position and sends it to a *Si* interface that offers a 1000-fold interpolation factor resulting in an impressive 36 pulses per arcsecond position resolution.

The position encoder, like the velocity encoder on the motors, is of the incremental variety so to get the absolute position of the turret each ring is equipped with a index mark which acts as a reference position. The data from the interfaces are sent to an APCIe-1711 counter board from ADDI-DATA [9]. This board keeps track of the pulses and counts how far from the index mark they are. As long as the counter board has power, absolute positioning is achieved.

Attached to the azimuth drive housing is the azimuth mirror housing and 45° mirror holder that deflects the light from the elevation axis into the lab. The mirror is 350 mm in diameter and 60 mm thick, glued onto six pads suspended by three adjustable bolts protruding from the surface of the mirror holder, allowing mirror adjustment. The adjustment is necessary to make sure the mirrors are perfectly aligned as the lightbeam has to pass through a narrow window in the lab, irregardless of where the turret is pointing.

The elevation drive housing sits on the azimuth mirror housing perpendicular to the azimuth axis. It is identical to the azimuth housing except for a differently shaped counterweight holder.

(a) Encoder ring and readhead (b) Encoder interface (c) APCI counter board

Figure 4: Position encoder components

The elevation mirror housing and mirror holder are also very similar to their azimuth counterparts except for a motorised lenscap mounting plate and a socket for a Lichtenknecker Optics [10] 200 mm lens with a focal distance of 6 m.

Currently, a very primitive optics bench is set up with a 45° household mirror, primary focus 60 cm along the bench where a bare SLR can be mounted, optionally a secondary lens projecting the solar disc onto a screen for safe observing.

Apart from the optics bench, the control room is also equipped with a GNU/Linux desktop for turret control. The Signum interfaces for position feedback are mounted clearly visible (as visible as the length of their cables would allow) in the lab to ensure proper operation. The interfaces are connected to the PC via the APCI counter board.

In addition to the desktop, two Arduino-based [11] remotes, or *handpaddles* have been developed. One corded and one wireless for use on the roof. These handpaddles allow for fine tuning offsets if the image drifts during long observations as well as allowing manual control during service and calibration.

For safety reasons several layers of emergency stops have been designed (all are not currently implemented). There is a watchdog that cuts power in case of a software crash, software endstops on the position encoders that disable the EPOS controllers and finally mechanical endstops that cuts power to the motors should the turret move too far.

There is also an emergency stop button and an ammeter showing current consumption of the motors as well as a control panel for the lenscap motor.

4 Software description

A GNU/Linux PC is used to control the turret. A set of programs written in C, some rewritten in C++, deal with hardware control, astronomical calculations and a full featured graphical interface.

Motion control is done by communicating to the four EPOS devices through a CAN-bus [7] using a Linux driver supplied from Maxon.

The servo system was written from scratch utilizing the built-in velocity control loop in each EPOS giving them the task of maintaining a fixed velocity as accurately as possible. Several service programs were written to test the motors and log the feedback performance.

The position encoder is used to ensure that the turret is pointing in the right direction by adjusting the command velocity to the EPOS devices. This is done in a software position loop encapsulating the velocity loop, also called a cascade control loop.

The S-SST inherits much of its control software from the SST. It was developed by Göran Hosinsky, John Rehn, David Kennedal and Peter Dettori [1].

The *Turret Control System* software contains everything necessary for running the telescope. It tracks the sun and planets in the solar system, other stars as well as other objects manually added to its database. For solar observations it has a separate window enabling specific solar observation features.

Part II

Technical description

5 Turret cabling

The turret has a fair amount of cables from motors, sensors etc going down to the control room (roughly 200 m in total). These are all passed through a watertight enclosure mounted on the turret base for easy access. The cables are either weatherproof themselves (lenscap and position encoder cables) or put in corrugated plastic tubing (motor cables) and passed through cable glands at their endpoints. All cables are also put in braided polyester sleevings to reduce friction and increase UV protection.

[7] A serial protocol used mainly in the automotive industry to connect different components in an effective way. It uses two data lines connecting the PC with the motor controllers daisy chained.

Everything on the roof is rated at IP66 [8] or higher (completely dust tight and protected against powerful water jets), excluding the outlet and the two large tube glands on the bottom of the electronics enclosure (rated IP44).

In figure 5, the wiring of all the cables is shown schematically. The lenscap and elevation axis cables have loops to prevent them from stretching when the turret moves.

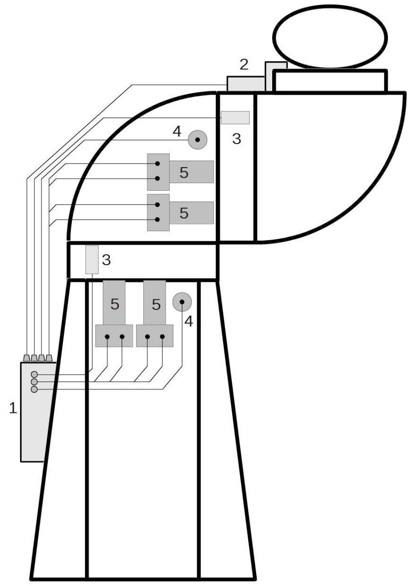


Figure 5: Turret cabling schematic, see table 2 for explanation

1	Electronics enclosure
2	Lenscap motor
3	Physical endstop switch
4	Position encoder
5	Motor enclosure

Table 2: Turret cabling legend

[8]Ingress Protection Rating; a measure of protection from solid objects and water

6 Turret electronics enclosure

Some of the electronics had to be mounted on the roof for signal noise reasons as well as for convenience. The electronics enclosure is powered from a UPS[9] cable that powers the motors through the four EPOS controllers and a 24V regulator. There is also a regular wall outlet on the side of the enclosure for service work.

In addition to power, the lenscap motor and endstops cables goes though the enclosure as well as cables for motor control, position encoders and the handpaddle receiver.

To protect the turret from being moved outside it's allowed region, physical endstops will cut power to the EPOS devices. These can be reset using the circuit breakers. The process of resetting the endstops is described in section 8 and appendix B.

The entire wiring schematic of the electronics enclosure can be found in figure 6

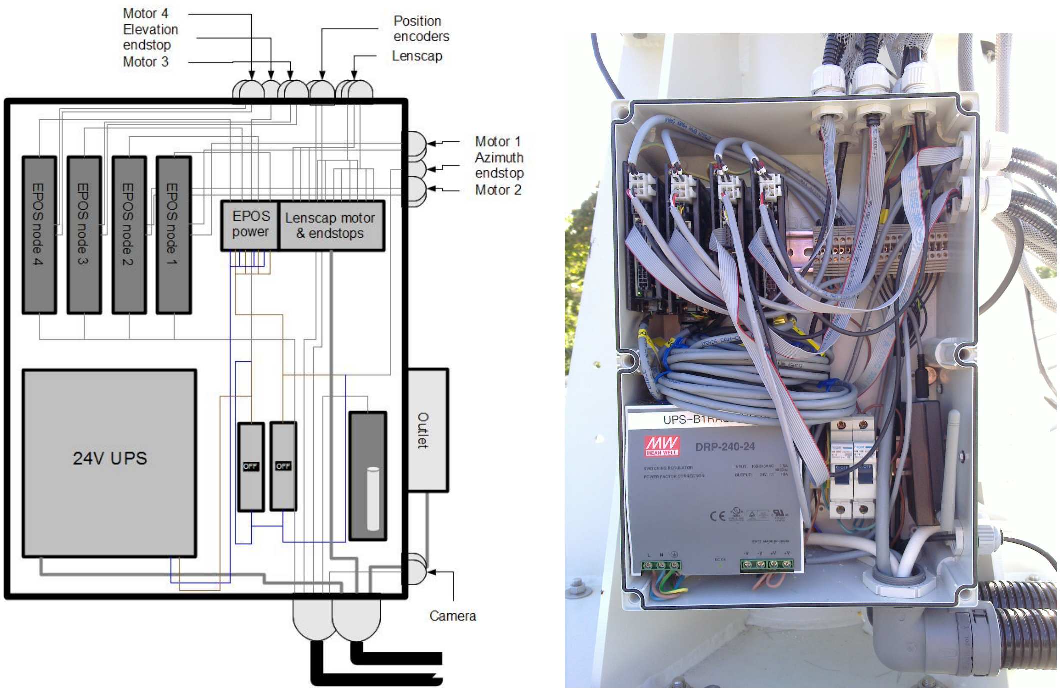


Figure 6: Schematic and photo of electronics enclosure

[9]Uninterruptible Power Supply, giving power to the turret even in the case of a mains outage

7 Control room

7.1 CAN

The four EPOS:es are connected in series, each with a unique *node ID*, with the control PC as master. The protocol, CANopen, transmits data to all nodes via each device and only the node with a matching ID responds. The EPOS:es have built-in object dictionaries that enable reading or writing data (such as setting a command velocity or reading the measured velocity).

7.2 APCI counter

The counter board is what keeps track of the turret position from the position encoders. It is configured as two 32-bit, 5 MHz counter modules (one for each axis) and two general purpose modules for digital I/O signals. Only one I/O signal is currently used, which is sent to the watchdog.

7.3 Watchdog

This is a safety feature designed to cut power to the motors in the event of a software crash. The watchdog is Arduino-based and gets an input signal from the APCI counter. If a square wave is applied to the input, the watchdog enables power to the motors via a mechanical relay.

Every piece of software controlling the motors sends a square wave to the watchdog, but in the event of a crash the square wave is terminated and power is cut. As soon as the program is restarted, the square wave is applied again and power is restored.

7.4 Handpaddles

Two separate remotes or handpaddles have been made for this project, one being connected to the PC, the other is wireless for use on the roof. They both have a four-way joystick, a centre button and a three-way speed switch (see figure 7).

When used with some of the service programs they rotate the axes manually. When used with the turret program, they can nudge the image if it has started to drift (caused by poor mirror alignment among other things).

7.5 Software structure

The software for controlling the motors has the following structure, arranged from lower to higher level:

- CAN protocol: deals with communication over the CAN bus
- USB protocol: deals with communication over USB
- EPOS interface: uses either the CAN or USB protocol to send and receive data to the EPOS devices
- APCI protocol: retrieves data from the APCI counter board and the position encoders
- Axis class: combines one APCI object and two EPOS objects to form a complete axis

Figure 7: The wireless and wired handpaddles

Each level is "unaware" of the specifics of the underlying objects, so when working with an Axis object, knowledge of the CAN or USB protocol is unnecessary.

The USB protocol was developed mostly as an introductory task. The turret and service programs all use the CAN protocol. However, the two protocols are equivalent and either can be used by the EPOS interface.

7.6 Service programs

To aid in troubleshooting, testing and calibrating several service programs have been written. The most used program is `runMotorsCAN` which is a program that uses the CAN protocol to run the turret manually via keyboard or handpaddle. Each axis can be controlled separately and sensor values printed.

For servo tuning purposes the programs `testMotorsCAN`, `pTuning`, `iTuning` and `positionLoop` can be used. They read what servo parameters to use from configuration files and prints sensor outputs for further analysis.

7.7 Turret program

When using the S-SST for observing, the program `xturret` is used. It is the control software for the SST and has only been modified to work with the S-SST hardware. It has features far beyond the scope of this document but in short it is used for locating and tracking celestial objects (stars, planets, moons, the sun and even features on the solar surface).

8 Endstop design

A (northern hemisphere) solar telescope never needs to look due north, so an early design choice was to prevent elevation axis cables from being pulled off by having an azimuth deadzone $80°$ wide pointing north. This makes the cabling much easier to do. The deadzone is located at $10° \pm 40°$. The elevation axis is also limited and can only move between $0°$ and $120°$. Movement past these endpoints is prevented in three levels:

Software limit

The turret program can limit the axes from moving past their allowed regions.

Encoder endstop

The two encoder rings each have two endstop markers attached. When movement past one of these marking is detected, a signal is sent to the EPOS controllers causing them to stop the motors. The signal then has to be reset and the motors moved back into the allowed region[10].

Physical endstop

There are physical endstop switches mounted on the turret. These should *never* be triggered unless there is a serious problem. When triggered they cut the power to the EPOS:es, thus preventing the motors from being moved back. This has to be reset on the roof following the steps listed in appendix B.

[10]As of now, this is not fully implemented. The signal cable from the encoders is currently not connected to the EPOS:es

Part III
Servo optimization

9 Project goal

The purpose of the servo system is to track the sun (and other celestial objects). The main goal of the project was to implement and tune the servo system to be as good as possible. This rather ambiguous goal can be quantified in the following way:

The optics has a minimum angular resolution of 0.5-0.9 arcseconds for visible light according to the Rayleigh criterion which states that two point sources are distinguishable when the principal diffraction maximum of one source coincides with the first minimum of the second[12]. This is calculated by:

$$\sin\theta = 1.22\frac{\lambda}{D} \quad (1)$$

Where θ is the angular resolution, λ is the wavelength of the light and D is the diameter of the lens.

This means that with a entry lens of 0.2 m, objects separated by less than roughly 1 arcsecond will be impossible to distinguish (on the sun, this equals to 700 km).

In addition to the diffraction limit, the atmosphere also limits the angular resolution. Random movements of the air causes turbulent bubbles that will distort light as it passes through. The magnitude if this disturbance is called *seeing* and it affects all ground-based telescopes, causing point-like objects to look speckled and smeared out.

The apparent size of an object, i.e. the magnitude of the seeing depends on air pressure, temperature, wind speed etc. and is in the order of 0.5-5 arcseconds. On a very good night (daytime seeing is worse due to differential heating) at the S-SST the seeing might be 1-2 arcseconds (compared to 0.5 at the SST)[13].

Combining the diffraction limit and the seeing gives a combined resolution limit of about 2 arcseconds. The goal of the project is to have an rms error of less than 2 arcseconds from the motors, meaning the motors should have only a small effect on the optical performance of the telescope.

To achieve this, a good control system must be chosen.

10 Basic control theory

Any dynamical system that needs to be regulated is dealt with using control theory. It is a branch of engineering and mathematics that, in short, is used to maintain a predefined state of a mechanical system.

The generic type of control loop is presented in figure 8. A reference value is set externally that the controller tries to follow. A sensor measures the output of the system that the controller in turn uses to adjust the system input. This is done repeatedly as the system output approaches the reference.

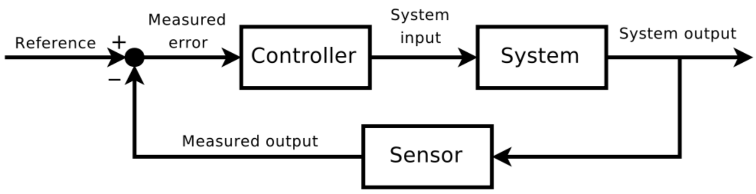


Figure 8: Generic feedback loop

The purpose of the control loop is usually two-fold: first to reach the reference value, second to maintain that value regardless of perturbations. An electrical stove will serve as an example of a mechanical system governed by different control loops.

10.1 Open-loop

A open loop has no feedback. In the case of an electrical stove, the reference value is what is set on the dial. The controller in this case translates the dial setting into a current sent to the hot plate.

There is no active temperature regulation in this set-up, but the plate will reach a stable temperature as heat radiation and heat production comes to an equilibrium. Any perturbation (like a pot of water being put onto the hot plate) will cause it to reach a different temperature.

10.2 Feedback loop

If the system output is fed back to the controller it is a closed loop system. This feedback is used to modify the reference value according to a set of rules to regulate the system performance. Depending on the purpose of the system, different rules can be applied (optimal for accuracy, stability, speed etc.).

The simplest example is to use a thermometer on the hot plate to compare the reference temperature and the actual temperature. The difference is added to the reference and sent to the hot plate. The larger the error is, the larger is the correction term. This is a simple type of a PID loop discussed below.

10.3 Feed forward

The very best way to achieve good control is to model the physical system into the controller. If the system and its environment is well known, this can be used as a basis for the control loop.

In the case of the hot plate the heating power can be calculated along with heat transfer onto the plate and its environment. Any foreseen disturbances (like a pot of water being put on the plate) are aptly accounted for, but unforeseen ones (e.g. a leaking pot) need to be compensated for by using regular feedback or similar.

11 PID

The most commonly used feedback loop is the PID-controller. It is easy to implement, easy to understand and does not require any knowledge of the particular system to be controlled.

A PID controller uses three different modes (Proportional, Integral and Derivative) to control the system looking at present-, past- and future errors respectively. All modes need not be present and can be adjusted independently. These modes are described below and explained through figure 9.

11.1 Proportional

The P-term accounts for the current error, $e(t)$ in the system by subtracting the measured output from the reference value. A multiplicative factor K_P (also called gain) controls the strength of the P-term. Its mathematical definition is:

$$P = K_P \cdot e(t)$$

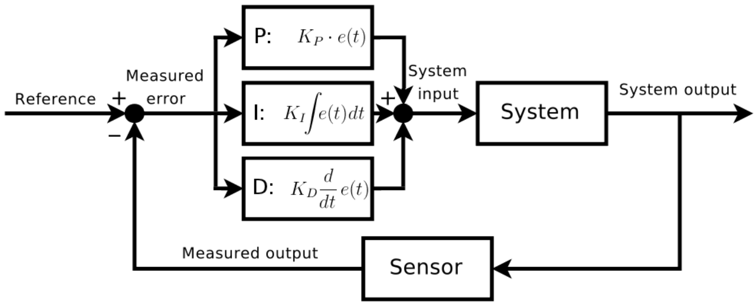


Figure 9: Generic PID loop

Basically, the larger the error, the larger the correction term from the P-term (thus called proportional term). A higher K_P gives a faster response but might produce overshoot and oscillations. Also, a P-only controller will actually never reach its reference value, it will only approach it. To deal with these steady-state errors another term is necessary.

11.2 Integral

By integrating the proportional error both its magnitude and duration will affect the controller. Any steady-state error will accumulate into a larger I-term that will bring the controller to its reference value. As with the P-term, the strength of the integral term is governed by a multiplicative factor; K_I according to the definition below.

$$I = K_I \int_0^t e(\tau) d\tau$$

Even more so than the P-gain, a too high I-gain will cause a heavy overshoot since errors accumulate over time (also called wind-up). This can be very problematic if a large sudden error is introduced (or the reference value is changed). A common way to fix this problem is to "clamp" the integral term to ensure it is within a predefined range.

11.3 Derivative

To deal with sudden disturbances and to reduce the settling time of the controller, a derivative term can be included. Its definition is:

$$D = K_D \frac{d}{dt} e(t)$$

Since the derivative is calculated only by looking at adjacent sensor values, it is very noise sensitive. Tuning the D-term can therefore be difficult and a low-pass filter is often used.

11.4 Cascade PID

In many applications (e.g. motion control) a simple PID controller is not sufficient. Usually the physical reference parameter is a position, but motor controllers are easiest to control by velocity or acceleration. Also, the position reference value might only change a few times a second whereas a motor needs very frequent updates to work accurately.

To solve these problems a cascade PID controller can be used. One outer PID loop takes the physical reference parameter as input and outputs a reference for a inner, faster PID loop that controls the actual device. In the case of motion control an outer position loop outputs a reference velocity or acceleration for the inner loop that controls the motor. The two loops can run at quite different frequencies as long as the outer is slower.

12 S-SST servo system description

The current design of the servo system for motion control in the S-SST is two cascaded PI-loops, the outer being a slow PI position loop and the inner a fast PI velocity loop. The derivative term is not used in either loop because the EPOS:es are PI-only by design and high frequency corrections are not needed in the outer loop.

The motors are controlled individually by EPOS motor controllers that include a PI implementation (no derivative term available) for velocity control. It also includes a feed forward term that is not currently used. The P and I tuning parameters (also called gains) are set as constants when configuring the EPOS:es. These loops run independently at 1 kHz updating motor current and reading the actual velocity from sensors on each motor.

The reference values for the velocity loops are provided by two position PI loops (one for each axis). These loops get their reference values from the turret control software (based on the suns position, time of day, etc). Based on the reference position and the current position of the turret, a reference velocity is calculated and sent to each nested velocity loop at 5 Hz.

The performance of this design is discussed in section 14.3

13 Optimization process

13.1 General PID tuning techniques

There are many ways to tune a PID loop, some can be considered quite general purpose. A common method that is fast and provides reasonable gains is[14]:

1. Zero all gains
2. Apply square waves to the system
3. Raise P until the system overshoots
4. Raise I until overshoot is approximately 15%
5. Raise D to remove some overshoot

A more thorough and systematic albeit empirical method is the Ziegler–Nichols method:

1. Zero all gains
2. Apply square waves to the system
3. Raise P until the system begins to oscillate
4. Note the P gain used (denoted ultimate gain, K_U), and the period of oscillation (T_U)
5. Set gains according to table 3

	K_P	K_I	K_D
P	$0.5K_U$	-	-
PI	$0.45K_U$	$1.2K_P/T_U$	-
PID	$0.6K_U$	$2K_P/T_U$	$K_PT_U/8$

Table 3: Ziegler–Nichols method of PID tuning

These methods provide a good starting point for further fine-tuning as each parameter affects the system in a predictable way according to table 4[15].

Response	Rise-time	Overshoot	Settling time	Steady state error	Stability
Increase K_P	Decrease	Increase	Small Increase	Decrease	Degrade
Increase K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increase K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

Table 4: Effects of independently changing P, I or D gains

Many manufacturers provide proprietary auto-tuning software along with their products that runs the system through series of tests to automatically determine the best parameters.

13.2 Cascade PID tuning

Even though a cascade PID system has twice the degrees of freedom of a regular PID system, tuning is not that complicated. It is done in two steps; first the inner loop is tuned separately using one of the techniques above. Then the outer loop is tuned separately, treating the inner loop as a closed system. Since the loops work at difference frequencies and in different domains, they do not interfere with each other[16].

13.3 S-SST servo tuning procedure

Usually a PI loop is a lot simpler to tune than a PID loop due to fewer degrees of freedom, so the tuning process was fairly straight forward. The motor controllers included servo tuning software that unfortunately was not ideal since the nominal motor speed is much lower than what the tuning software is using.

Instead manual tuning had to be performed. The motors were assumed to be equal, so both motors on each axis runs with identical parameters. Normally, manual tuning is performed by first tuning a P-only loop, then increase I-gain until the steady state error vanishes.

Unfortunately, the high amount of friction made that impossible. Any P-gain (within reasonable limits) was unable to overcome the static friction of the turret, so the tuning process had to be modified. Measuring overshoot also proved to be difficult due to the limited resolution of the motor's encoders.

The chosen tuning method as seen below for the velocity loop was therefore modified taking into account the challenges stated above and the fact than a low steady-state error is the top priority for the velocity controller.

1. Set P-gain to a stable value that enables movement
2. Apply square waves to system (0 to 5 arsec/s), changing I-gain in small increments
3. Choose the lowest I-gain with a minimum steady-state error
4. Fix the I-gain and alter P-gain until the system has a reasonable rise-time (<3s)
5. Optional: Adjust P and I around their centre points to further reduce rise-time

The outer position loop does not have to deal with friction, it only sees the velocity controller as a “black box” meaning that the tuning of the position loop was more straight forward. It was done according to the Ziegler–Nichols method in section 13.1

14 Servo tuning process

The following section describes the tuning process of the elevation axis only. During the process of calibration, one of the motors broke and has been disabled for the time being. Until the motor is replaced, one axis must run on only one motor. The load on the azimuth axis is constant, whereas the elevation axis must “lift” the lens when rotating upwards and therefore requires more power.

As a temporary solution, the azimuth axis runs on only one motor and elevation axis runs on two motors. Until the replacement motor arrives, the azimuth servo is left roughly calibrated, but it will be tuned in a similar way as the elevation axis described below.

14.1 Velocity tuning

For the first tuning cycle, P-gain was set to 2000 (anything between 500 and 3000 will work, as determined through trial-and-error) and I-gain was tested between 100 and 3000 for 60 seconds each. For the latter half of each run (after a steady state has been achieved) the rms[11] error was calculated and plotted against the corresponding gain value. These plots are shown in figure 10.

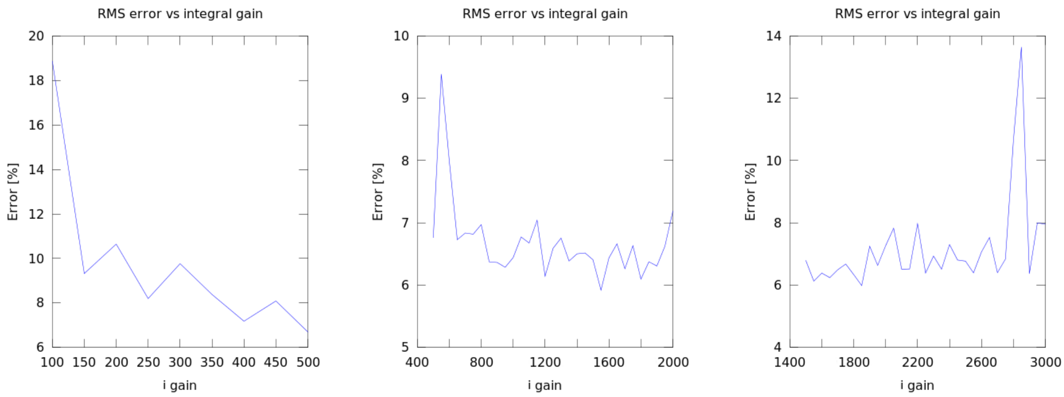


Figure 10: I-gain tuning (P=2000)

[11] The root mean square, or quadratic mean is calculated by: $x_{\text{rms}} = \sqrt{\frac{1}{n}\left(x_1^2 + x_2^2 + \cdots + x_n^2\right)}$

According to the tuning plots, the lowest rms error is achieved at an I-gain of around 1500, but this is centred in a rather wide minimum. Too high an I-gain will make the system unstable as stated in table 4, so an I-gain of 1000 was chosen. This makes the system a lot more stable than a gain of 1500, with only a marginal increase in rms error.

With an I-gain of 1000 fixed, the P-gain was cycled between 500 and 3000 in a similar manner. The main purpose of the P-gain is to decrease rise-time[12], so figure 11 shows a plot of rise-time against P-gain.

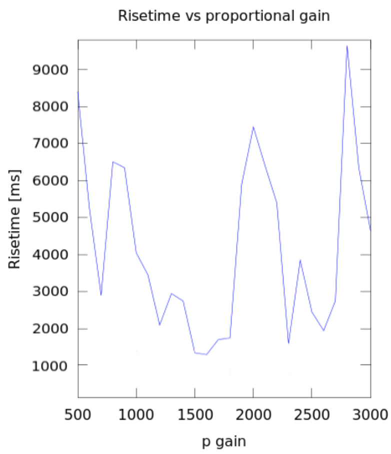


Figure 11: P-gain tuning (I=1000)

The lowest rise-time occurs for a P-gain of 1600, which was chosen for the final parameter. A note on the rise-time calculation is needed here since the sampling time only allowed for a few position encoder pulses per data point. This caused very coarse velocity data calculated from the position encoder. To circumvent this problem a smoothing algorithm was applied to the velocity data and the time taken for the smoothed velocity to rise from 10 % to 90 % of the command velocity was used instead.

The difference between raw and smoothed data can be seen in the plots in figures 12 and 13. The red lines represents raw and smoothed position encoder data. Green and blue lines represent motor velocity data and the black line represents the command velocity. These plots also demonstrates good and bad servo parameters. Figure 12 exhibits “ringing” or damped oscillations due to a high P-gain and a negative steady-state error due to a low I-gain.

Figure 13 shows the motor performance with a well tuned servo system. The command velocity is reached quickly (within 3 seconds), is stable and with no noticeable steady-state error.

[12] Usually defined as the time required for a system to change from 10% to 90% of an applied step height

14.2 Position tuning

The tuning of the position loop can be done without any knowledge of the velocity loop. Also, since the velocity loop takes care of the high friction in the system, the position loop proved to be a lot simpler to tune. The procedure simply involved setting the I-gain to zero and increasing the P-gain until sustained oscillations occurred.

According to figures 14, sustained oscillations occurred at a P-gain of 3.0 with an oscillation period of 1.1 seconds. According to the Ziegler–Nichols method of table 3, suitable control parameters are: K_P = 1.35 and K_I = 0.29.

14.3 Servo evaluation

The chosen servo parameters are summarised in table 5.

Velocity K_P	1600
Velocity K_I	1000
Position K_P	1.35
Position K_I	0.29

Table 5: Optimal servo parameters for elevation servo

To test the performance of the servo system, normal use was simulated by running the axis at 5 arcseconds/s, increasing the speed by 0.001 arcseconds/s every second for a total duration of 60 seconds. The choise in speed is based on the fact that the suns elevation speed varies between 0 and 10 arcseconds/s at present latitude. The change in speed is supposed to simulate a small portion of the arc the sun makes across the sky.

The position error was recorded at 5 Hz and used to calculate an rms error for the entire run (excluding the first 5 seconds needed for the turret to settle). Using the parameters in table 5, an rms error of 0.13 arcseconds was recorded. The whole run is shown in figure 15.

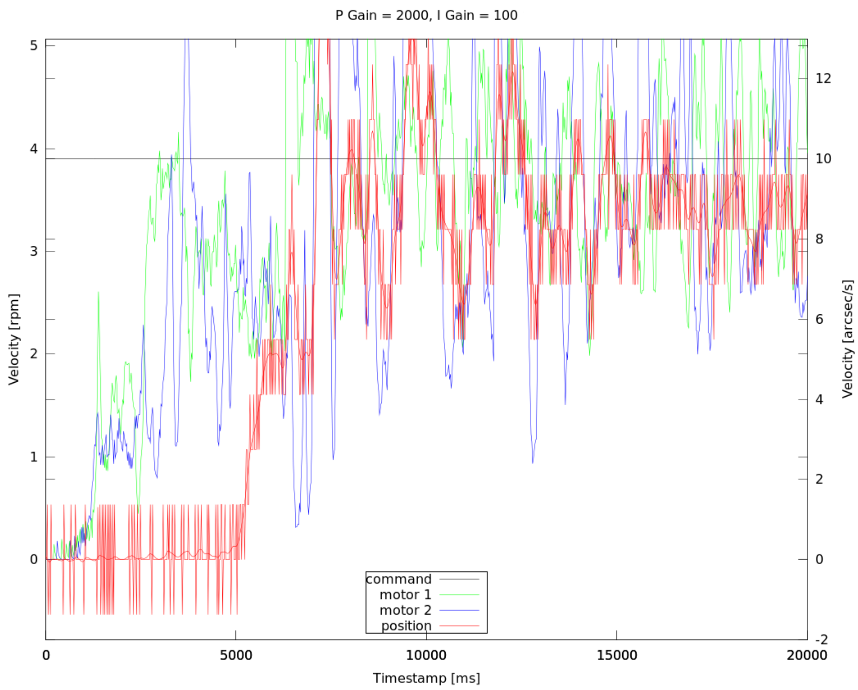


Figure 12: Poorly chosen parameters (P=2000, I=100)

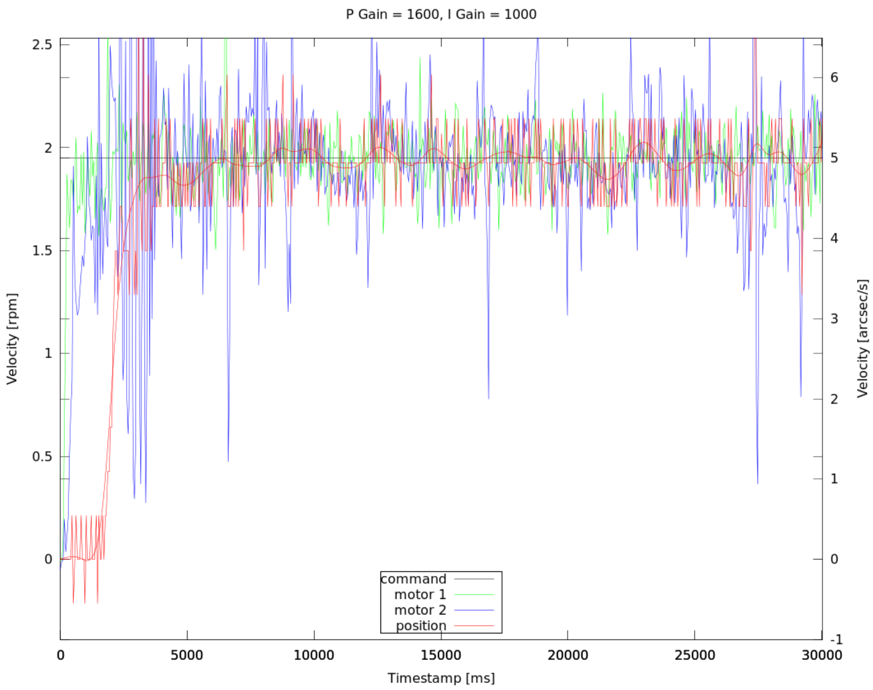
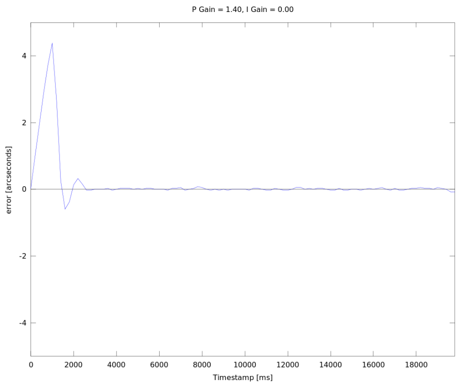
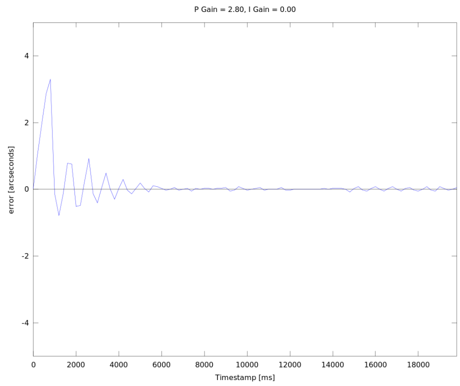


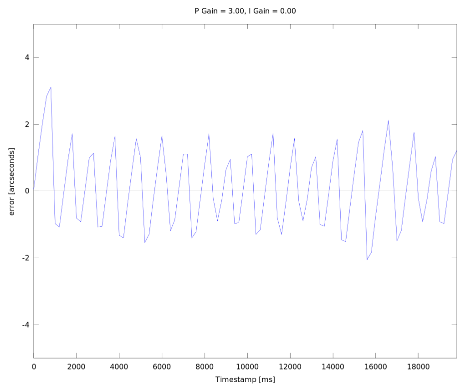
Figure 13: Well chosen parameters (P=1600, I=1000)



(a) Stable, P-only position loop



(b) Stable, with ringing (P=2.8)



(c) Unstable, bounded oscillations (P=3.0)

Figure 14: Position loop behaviour for different P-gains

15 Conclusion

With a desired motor precision of less than 2 arcseconds based on the diffraction limit of the lens and the atmospheric seeing, the final precission of the servo system is less by over a factor of 10 (namely 0.13 arcseconds). This means that the elevation axis has a negligible contribution to disturbances limiting the performance of the telescope.

Since there is less load on the azimuth motors, an equal or higher precision will be achieved when the faulty motor is replaced. A combined precision of around 0.2 is expected, still one order of magnitude less than other disturbances.

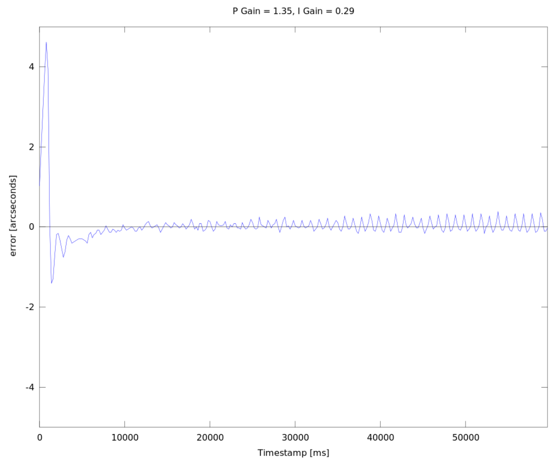


Figure 15: Performance of the servo system using parameters from table 5

16 Possible servo improvements

For a motor controller to function over a wide range of input values, a simple (even cascade) PID-controller might not be enough. Here are some suggested improvements that might help the controller's performance

16.1 Gain scheduling

The motor speed when the turret is homing or moving to objects is about 100 times faster as compared to when it is tracking. Ideal servo parameters for tracking might not be perfectly ideal (even unstable) for homing. Gain scheduling can be introduced to overcome this problem.

It involves changing the servo parameters based on the commanded input. This is done in discrete steps and would be implemented on the S-SST by having two sets of servo parameters; one for tracking and one high speed moves.

16.2 PDFF

To increase the stiffness (tolerance to low frequency disturbances like friction), a PI controller is not always the best solution. A feed forward factor in addition to P and I often does a better job.

It does however require a more thorough tuning process and a good knowledge of the loads and friction forces involved.

16.3 Current control

One option for the inner controller is to use a current loop instead of a velocity loop. This is more closely related to the hardware (the current is directly proportional to the torque) meaning better performance might be achieved at the expense of it being harder to calibrate and tune.

17 Outlook

The telescope is by no means finished. The original project plan did not involve mounting it on the roof, so much has been achieved but there is still more to do. This section outlines several to-do items as well as possible upgrades.

17.1 Mirror alignment

No matter how well the servo system is performing, poor mirror alignment will cause the solar image to drift during lengthy observations. Currently, the projected image is stable for a few hours, but that can be improved.

The mirrors were levelled during the gluing process, but using a mirror instead of the primary lens and a vertical laser beam going up from the lab a much better alignment can be achieved by observing the reflected beam.

17.2 Motor upgrades

The motors are running on the edge of their optimal performance region both torque and resolution wise. Two possible upgrades; an additional gearbox and a higher resolution velocity sensor will remedy this.

17.3 Optics setup

The optics setup is currently being upgraded to proper lab-grade equipment from Edmund Optics. This includes a front surface mirror, a beamsplitter and an achromatic lens among other things. This will allow direct observations and data taking to be made simultaneously.

Still on the wish-list is a CCD or high-end SLR camera and a spectrometer. One possible use of the S-SST involves using a high resolution spectrometer to measure the Zeeman splitting of hydrogen and deducing the magnetic field strength on the Suns surface.

17.4 Endstops and watchdog

To make the S-SST more failproof, the remaining endstop level and the watchdog needs to be implemented. A interface box currently under construction will connect the encoder endstop signals to the EPOS:es (already configured to stop when such a signal is received). The box will also house the watchdog.

17.5 Lenscap

The lenscap is currently controlled manually (with automatic endstop detection) via a table-top box constructed by Digitalmekanik. A possible future upgrade involves interfacing the box to the turret PC and a weather station on the roof.

The idea is to be able to control the lenscap from the PC and have it automatically close when rain or high humidity is detected.

17.6 Manuals

To ensure everyone will be able to use the S-SST, a thorough manual is under development (based on the manual for the SST control software). This in addition to the technical documentation provided in this paper will make it relatively simple for anyone to operate the turret and perform simple observations without help.

17.7 Educational use

The S-SST is intended to be used for educational purposes. When it is considered safe to use by anyone it will be included as part of the lab activities offered at Vetenskapens Hus. Ideas for lab activities will be developed and tested and may include telescope construction, observational technique, solar physics (sun spectrum, sunspot observation etc.) and celestial mechanics.

Other uses will be teacher education, specialised high school projects and remote live viewing (latest sun images shown in the lobby and on the webpage)

18 Final words

It is my hope that this project never is to be considered “finished”, rather a constantly evolving and improving telescope that many people contribute to. This has been a truly amazing project to be part of, helping me to learn many new skills in many different areas. I was not before, nor am I now considering myself an expert in programming, electronics, hardware manufacturing, motion control, telescope technique or solar physics.

But I’ve learnt enough skills in these areas to feel confident in my abilities and eager to dig even deeper. This project was also not without a social component even though I worked alone as a masters student. Many people with separate areas ef expertise had to be brought together to make this happen.

References

[1] "Institute for solar physics." http://www.solarphysics.kva.se/.

[2] "Departement of astronomy." http://www.astro.su.se/.

[3] G. B. Scharmer, B. V. Gudiksen, D. Kiselman, M. G. Löfdahl, and L. H. M. R. van der Voort, "Dark cores in sunspot penumbral filaments," *Nature*, vol. 420, pp. 151–153, 2002.

[4] S. Wedemeyer-Böhm, E. Scullion, O. Steiner, L. R. van der Voort, J. de la Cruz Rodriguez, V. Fedun, and R. Erdélyi, "Magnetic tornadoes as energy channels into the solar corona," *Nature*, vol. 486, pp. 505–508, 2012.

[5] G. B. Scharmer, K. Bjelksjö, T. Korhonen, B. Lindberg, and B. Petterson, "The 1-meter swedish solar telescope," *Proceedings of the SPIE. Innovative Telescopes and Instrumentation for Solar Astrophysics.*, 2002.

[6] "Maxon motors." http://www.maxonmotor.com. Motor manufacturer.

[7] "Scancon." http://www.scancon.dk. Rotary encoder manufacturer.

[8] "Renishaw." http://www.renishaw.com. Position encoder manufacturer.

[9] "Addi-data." http://www.addi-data.com. Counter board manufacturer.

[10] "Lichtenknecker optics." http://www.lo.be. Optics manufacturer.

[11] "Arduino." http://www.arduino.cc. Open-source electronics prototyping platform.

[12] M. Born and E. Wolf, *Principles of Optics.* Cambridge University Press, 1999.

[13] G. B. Scharmer, 2011. Private communication.

[14] G. Ellis, "20 minute tune-up," *PT Design Magazine*, February 2000. http://motionsystemdesign.com/mechanical-pt/twenty-minute-tune-up-19990201/.

[15] K. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, July 2005. http://eprints.gla.ac.uk/3817/.

[16] G. Ellis, "20 minute tune-up: Zoning in on position loops," *PT Design Magazine*, July 2000. http://motionsystemdesign.com/engineering-basics/twenty-minute-tune-up-0700/.

Appendices

A Personal contribution

This section lists the author's contribution to the S-SST hardware and software and what was already done upon the beginning of the project.

In October 2011 as the author got involved all the main turret pieces (those seen in figure 1, apart from the equipment on the bench) had been manufactured. The turret base, beampipe and azimuth drive housing were at Vetenskapens Hus, the others were at the manufacturer, Stockholms Digitalmekanik AB. Work began with one motor connected via USB to the PC, testing motion and saving sensor data.

More and more sophisticated control programs were written in parallel with servo analysis programs used for tuning. The motor was mounted on the azimuth drive housing to test friction load as well as the position sensor. As support for two motors was enabled, the azimuth mirror housing was attached increasing the load further.

Starting in early December, work on a proper CAN/USB interface began with the goal of using it with the turret control software inherited from the SST (written, in part, by Peter Dettori). This software is used for observations and handles all turret control. However, it had to be modified for the S-SST hardware.

By mid-January the elevation drive and mirror housings were mounted and testing with near-full load began. Software classes for controlling the axes were developed as the turret control software was modified to accept these classes.

As motor control and the servo system got finalized in March, work began with merging motor programs with the turret software. This proved to be a very time consuming task. With the transit of Venus approaching, the author also started planning for roof mounting together with various experts (supervisors, manufacturers, consultants, electricians etc.).

In mid April the turret software was in a state that enabled it to be initialised and tracking reliably. Additional hardware pieces like the lenscap (built by Digitalmekanik) and hand-paddles were also being finalised. With all major components working, preparations for mounting all required pieces in the roof began.

May was spent manufacturing weatherproof cables and preparing the electronics enclosure with all the hardware it would house. The main mirrors were also readied by carefully gluing them to mirror holders to be attached to the mirror housings. This work was planned and executed with much help from Klas Bjelksjö.

The turret was mounted on the roof on May 25:th with help from everyone involved in the project including Peter Dettori, Magnus Näslund, Klas Bjelksjö, Johan and Bertil Pettersson, as well as virtually everyone at Vetenskapens Hus.

In the few days leading up to the transit on June 6:th everything was wired up and waterproofed, endstops mounted, mains power attached etc. The final missing software pieces were added and everything was roughly calibrated to allow first light on June 3:rd.

The transit of Venus attracted a lot of spectators to the first S-SST event, despite the early morning and pessimistic weather forecasts. Unfortunately said weather only allowed for a 2 second observation with no data recorded.

Since then, the S-SST servo system has been tuned further, more hardware peripherals added and the optics bench upgraded. Work remains on streamlining the interface and incorporating the S-SST into the daily activities at Vetenskapens Hus.

B Endstop reset procedure

- Start the service program runMotorsCAN in reset mode with the command `./runMotersCAN -r`
- Note which axis triggered the endstop, and in which direction to move it to clear the endstop
- Open the electronics enclosure
- Locate the two circuit breakers in the middle of the lower DIN-rail and flip the switch of the right axis to the “ON” position
- Check that all four EPOS:es are on (green flashing LED)
- Switch on the remote handpaddle and press the centre button to enable the service program
- Use the directional keys on the handpaddle to move the axis clear of the endstop switch
- **IMPORTANT: Reset the circuit breaker, both should be in the “OFF” position**
- Close the enclosure and return to the control room

C Photos

(a) The SST during assembly

(b) The S-SST after mounting. Left to right: Peter Dettori, Magnus Näslund and Bertil PEttersson

Figure 16: Comparison between the SST and the S-SST

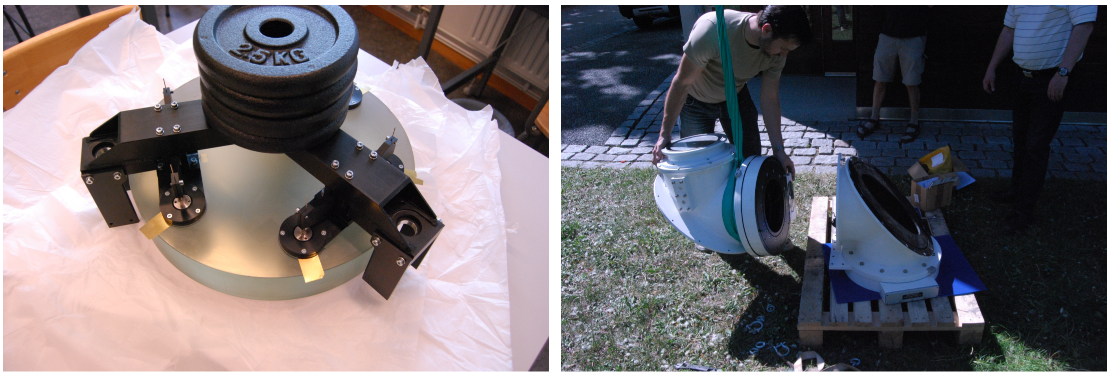


Figure 17: Mirror gluing and turret assembly prior to mounting

Figure 18: Lifting and mounting the beampipe

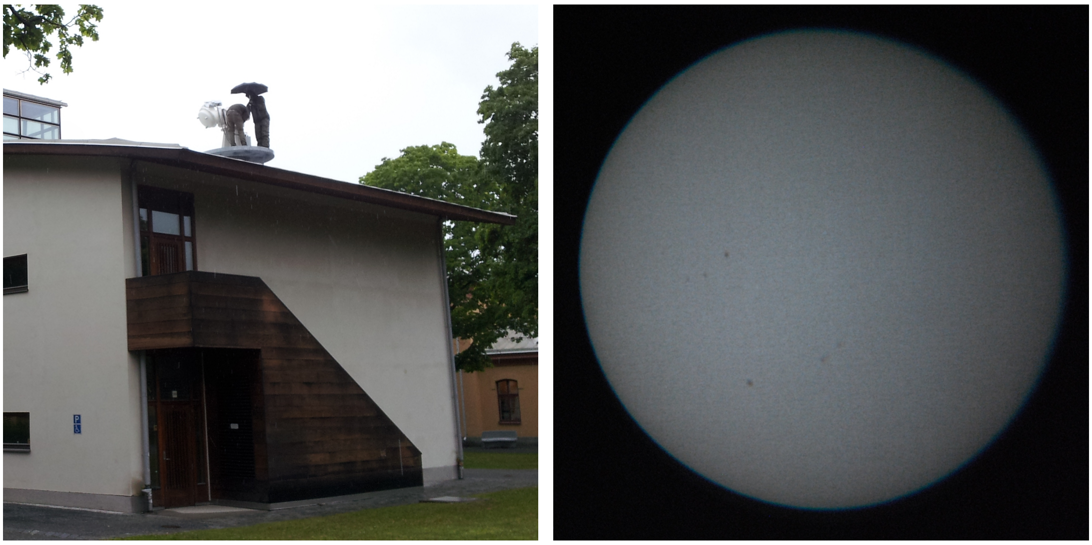


Figure 19: Last minute fix and first solar image