Mechanical Design and Motorisation of a Radio Telescope

Radio Waves Small Radio Telescope Project
Callista, astronomy association of EPFL

ME-314 : Projet d'ingénierie simultanée
Supervising Professor : Philippe Müllhaupt

Aguiriano Calvo Malena
Cayroche Louise
Dufour Edouard
Rochet Corentin

June 2021

Acknowledgments

We would like to express our sincere gratitude to our supervisor, Prof. Philippe Müllhaupt. He accompanied us throughout this whole project, dedicating time and attention to our work. Thanks to him, we were able to better develop many aspects of this work.

Also, we obviously thank Callista and Dimitri Hollosi for allowing us to be part of the Radio Waves project. We were happy to be involved in such a promising field that is radio astronomy.

Finally, we are extremely thankful to have worked with the Structure Pole, lead by Audrey Piccini who listened as well as guided us when developing the motor housing.
The Antenna Pole, with Oriane Robin, Joséphine Potdevin and Dominic Dahinden, must not be forgotten as they were of great help for our measurements.

All in all, every person working with us and helping us during this project was of considerable support and we were delighted to be given such an opportunity.

Contents

1 Introduction

1.1 Radio astronomy

Astronomy is a science which studies celestial objects and phenomena. Its oldest form is optical astronomy, a method using the visible light emitted by celestial objects to observe them. It uses photographic equipment such as a telescope coupled with a camera.

Radio astronomy differs from this technique as it uses radio waves to study celestial bodies (a radiation which wavelength is between $1mm$ at 300GHz and $10'000km$ at 30Hz). This facilitate the measuring of the amplitude and phase of said radio waves. Moreover, some spectral lines produced by interstellar gases such as the hydrogen spectral line (H-21) can be measured by the radio telescope. In our project, this is the spectral line of interest.

1.2 Radio Waves' project

Radio Waves' project was initially created in 2018. Its goal is to create the largest radio telescope of Switzerland in order to specifically study the H-21 spectral ray. It is supervised by the Laboratory of Astrophysics of EPFL, LASTRO.

Now in 2021, the project was relaunched with an intermediate goal: the Small Radio Telescope (SRT). It combines different poles of different expertise: the antenna and amplifier pole, the structure pole and the Motorization and Control pole. The data analysis one will appear once the radio telescope is ready.

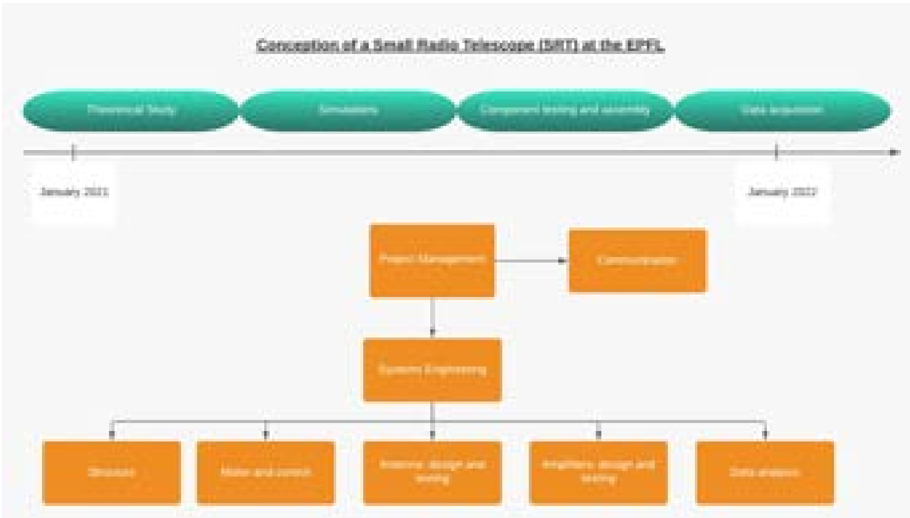

Figure 1: Organisation of the SRT Project[1]

1.3 Our goal

As the Motorization and Control pole our main goal is to have the antenna move precisely and fast enough so that it is able to track celestial objects ranging from low-orbit satellites to deep sky objects.

In order to achieve that, we must determine the mechanical design, the electrical implementation and the control algorithm of the motorization.

1.4 A mathematical introduction

1.4.1 Time

In astronomy, time cannot be measured in days like on earth, since one rotation of the earth in the solar system's reference frame is not 24 hours long. We have to measure time in sidereal time[2].

$$
\begin{aligned}
LST = [&6.697374558 + 0.06570982441908 * JD_0 \\
&+ 24.06570982441908 * (JD - JD_0) \\
&+ 1.948915002475965 * 10^{-14} * JD^2 + L/15] mod 24
\end{aligned}
$$

LST is the local sidereal time
JD is the time (in days) between measuring time and reference time (UTC 01/01/2000-12:00)

JD_0 is the time (in days) between the midnight before measuring time and reference time (UTC 01/01/2000-12:00)
L is the longitude in degrees of measuring position

1.4.2 Equatorial Coordinates

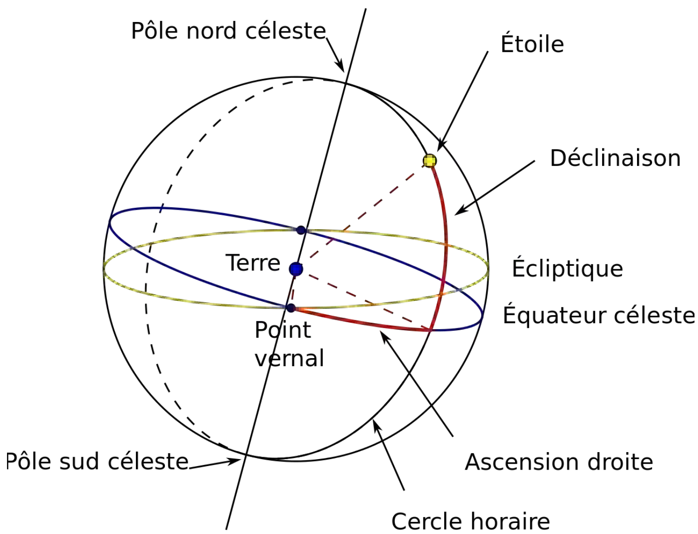

Figure 2: Equatorial Coordinates[3]

α right ascension $[0h; 24h]$ positive to the east
δ declination $[-90\deg; 90\deg]$ positive to the north

1.4.3 Horizontal Coordinates

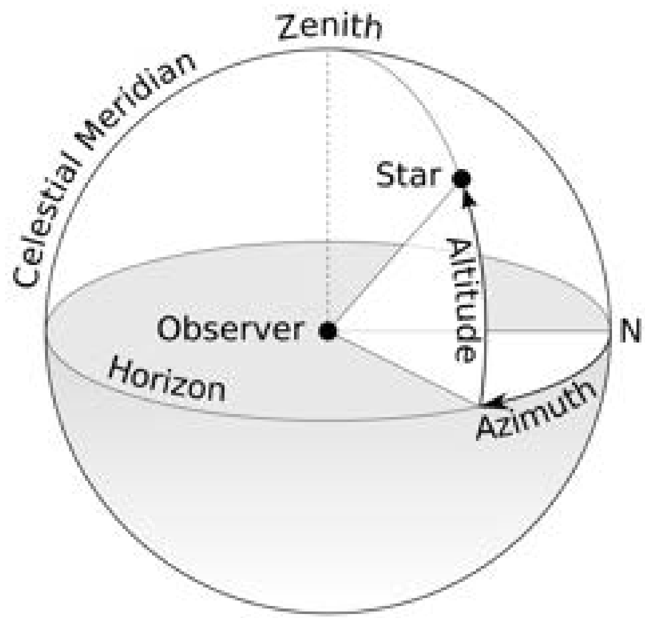

Figure 3: Horizontal Coordinates[4]

h altitude $[0\,\text{deg}; 90\,\text{deg}]$ positive to the zenith
Z azimuth $[0\,\text{deg}; 360\,\text{deg}]$ positive to the east

1.4.4 Conversion from Equatorial to Horizontal Coordinates

$$\sin h = \cos\varphi \cos\delta \cos LST - \alpha + \sin\varphi \sin\delta$$
$$\cos h \sin Z = \cos\delta \sin LST - \alpha$$
$$\cos h \cos Z = \sin\varphi \cos\delta \cos LST - \alpha - \cos\varphi \sin\delta$$
$$\varphi \text{ latitude measurement}$$

2 Engineering specifications

2.1 List of specifications

Table 1: Control Pole Engineering specifications

Ref.	Title	Description	Parent	Rationale	Verification Method
0	Main function	The control system shall point an antenna dish towards a desired space object or earth satellite			
0_1	Precision and accuracy requirements	The control system shall ensure accuracy standards	0	The antenna has a finite field of view, position must be ensured to guarantee good quality observations	
0_1_1	Tracking accuracy	At all time during tracking, the antenna dish shall be closer than [42] arcmin to the target	0_1		Comparison of dish orientation with deep sky object radio signature position
0_1_1_1	Total mechanical play	The mechanical play in the system shall stay under [15] arcmin	0_1_1	Excessive play in the system may render it uncontrollable for small displacements, making it impossible to achieve specified tracking accuracy	
0_1_2	Pointing knowledge	At all time the system shall be aware of its position to an accuracy of [15] arcmin	0_1_1	The knowledge of real position needs to be better than required pointing accuracy	Comparison of predicted dish orientation with real dish position calculated with deep sky object radio signature position
0_1_2_1	Unmeasured mechanical play	Play in the mechanical system that cannot be measured shall be limited to an angle of [5] arcmin	0_1_2	Large play may compromise pointing knowledge, as movement freedom of moving parts induces uncertainty about their positions.	
0_1_3	Movement resolution	The smallest unit of movement possible for the motor system shall be finer than [2] arcsec	0_1_1	Fine movement are needed for smooth control	

Continued on next page

Table1 – continued from previous page

Ref.	Title	Description	Parent	Rationale	Verification Method
0_1_4	Measure resolution	The smallest unit of movement measurable shall be finer than [2] arcsec	0_1_1	Fine movement are needed for smooth control	
0_2	Range requirement	The whole sky half sphere, solid angle [2π] rad^2 shall be accessible	0	Ensures the largest possible pool of observation targets	
0_2_1	Axis	The movement shall be possible along 2 axis : azimuth and altitude	0_2	2 degrees of freedom are needed to cover the surface of a sphere	
0_2_2	Azimuth range	The movement along the azimuth axis shall cover at least a full circle, or [360] deg	0_2 + 0_2_1	Needed to cover the full sky	
0_2_3	Altitude range	The movement along the altitude axis shall cover at least [90] deg starting at the horizon	0_2 + 0_2_1	Needed to cover the full sky	
0_3	Speed requirements	The system shall endure the various tracking speed required by the antenna	0		
0_3_1	Min track speed	The system shall be able to hold still	0_3	Objects such as geostationary satellites may stand still in the sky	We will impose a still position and measure the deviation
0_3_2	Max track speed	The system shall ensure a tracking speed over [1.5] deg/s on both axis	0_3	The motorization shall be able to track low earth orbit satellites, which have a maximum apparent angular speed of [1.07] deg/s	We will input our specification speed as a command and verify with our measurement system it is respected
0_4	Structural integrity	The mechanical system shall endure all stress during operation, or at rest	0		
0_4_1	Motor torque along altitude	The altitude axis transmission shall ensure a torque of [157] Nm	0_4 + 0_5_1	[157] Nm from wind forces, see Appendix, Wind MATLAB code.	We will measure deviation when our spec torque is manually applied
0_4_2	Motor torque along azimuth	The azimuth axis transmission shall ensure a torque of [157] Nm	0_4 + 0_5_1	[157] Nm from wind forces. 2.2.2.	Measure on prototype
0_4_3	Radial forces on altitude axis	The altitude transmission shall support a radial force of [857] N	0_4 + 0_5_1	The weight of the dish, its counterweight and the antenna shall be supported ([70] Kg, [700] N), and wind forces supported ([493] N, see Appendix, Wind MATLAB code). These forces apply orthogonally to one another	

Continued on next page

Table1 – continued from previous page

Ref.	Title	Description	Parent	Rationale	Verification Method
0_4_4	Axial forces on altitude axis	The altitude axis transmission support an axial force of [493] N	0_4 + 0_5_1	Linear wind forces, see 2.2.2	
0_4_5	Tilting forces on altitude axis	The altitude axis transmission support a tilting torque of [267] Nm	0_4 + 0_5_1	Wind may exert tilting forces on the axis, see Appendix, Wind MATLAB code	
0_4_6	Axial forces on azimuth axis	The azimuth transmission shall support an axial force of [700] N + weight of motor housing	0_4	The weight of the dish, its counterweight, the antenna, and the motor housing shall be supported	
0_4_7	Radial forces on azimuth axis	The azimuth transmission shall support radial forces of [493] N + drag force of the motor housing	0_4 + 0_5_1	Linear wind forces may apply, see Appendix, Wind MATLAB code	
0_4_8	Tilting torque on azimuth axis	The azimuth transmission shall support tilting torque of [464] Nm + motor housing forces	0_4 + 0_5_1	Wind forces may apply torque on the transmission	
0_4_9	Cable preservation	The movement shall be limited depending on the design to prevent excessive cable tangling.	0_4 + 0_6_2 + 0_6_3	Limitation of cable entanglement for cable entering moving parts	
0_4_10	Thermal	The components shall stay within their temperature operational domain (especially motors and electronics)	0_4		
0_5	Environment tolerance	The system shall be able to operate long-term outdoor in Lausanne, Switzerland	0		
0_5_1	Wind resistance	The system shall function properly during winds of up to [50] km/h during operation, and withstand a [0.9]kN/m$\hat{2}$ of dynamic pressure at rest	0_5	Allow operations [99]% [5] of the time in Lausanne, and comply to norm SIA261 [6]	
0_5_2	Rain resistance	The system shall withstand heavy rain without damage	0_5	Rain is common	
0_5_3	Humidity control	Humidity in the motor housing shall not disturb our electronics	0_5	Rotary encoders might suffer under high humidity	
0_6	Interfaces and communication	The motorization system shall be integrated in the system and communicate with other poles	0		
0_6_1	Structure base link	The motor housing shall be able to transmit the stresses of 0_4_6, 0_4_7, 0_4_8 to the structure base	0_6	Link with structure base shall be solid enough to sustain wind and weight forces	

Continued on next page

Table1 – continued from previous page

Ref.	Title	Description	Parent	Rationale	Verification Method
0_6_2	Power supply	[230] V AC shall be provided in the motor box to supply on board systems	0_6	Versatile, only requires 2 wires	
0_6_3	Communication between on-board system and distant user	Data i/o cable shall be provided to on-board system to communicate with distant user	0_6		
0_7	Robustness and maintainability	The system shall as much as possible be able to face unforeseen issues. Repairs and maintenance shall be as easy as possible	0		
0_7_1	Mechanical stop for altitude	[+0/+90] deg limit set in 0_2_3 shall be mechanically backed up	0_7 + 0_2_3	increased robustness that a full software solution cannot ensure	
0_7_2	Incoherent inputs handling	System shall handle incoherent inputs	0_7	Increased robustness	
0_7_3	Protection against external disturbance	System shall turn off when external disturbance torque exceeds [157] Nm	0_7 + 0_5_1 + 0_4_1 + 0_4_2	Limit for operations under wind	

2.2 Justifications

2.2.1 Accuracy, resolution and speed

To allow satisfying observations, we must point the antenna to the target with accuracy. Furthermore, we must guarantee near-continuous movement to minimize the disturbance for the observations. After talking to the antenna team, we decided on a pointing accuracy of $42arcmin$ and a movement resolution of $2arcsec$.

Regarding speed, we can find two extreme cases. We want our mount to be able to track low earth orbit satellites. These satellites can go up to $7.66km/s$ at $417km$ in altitude, like the International Space Station[7].For an observer on the ground, the maximal angular speed is $1.05deg/s$. To allow us to catch up to these fast objects, we decide on a maximal angular speed of $1.5deg/s$.
The second extreme case would be a polar object or a geostationary satellite. These objects stay at a roughly constant position. We therefore must be able to point to an unchanging position.

2.2.2 Wind

Calculation of wind-torque

In order to calculate wind torque, we assumed constant dynamic pressure on the parabola (either calculated from a wind speed, or mandated by a construction norm).
Also,exposition of the parabola to wind varies a lot with the elevation angle. For that reason, we calculated the wind exposition of the parabola for various elevation angles, in order to show a profile of torque constraint as a function of elevation angle (see Appendix, Wind MATLAB code).

Zenith position

The reference value for wind constraints in Lausanne is a dynamic pressure of $0.9kN/m^2$[6] (which is equivalent to an approximate wind speed of $140km/h$). To minimize the torque on the elevation axis, as well as the tilting torque on the slewing ring, the standby position will be the zenith position.
For our parabola, in the zenith position, the torque on the elevation axis at this wind speed was calculated to be $289.814Nm$. We must be able to hold a torque higher than this value at any moment.

Operation

To allow us to function 99% of the time, we must be able to endure a wind of $50km/h$[5].
For our parabola, in the worst position, the torque on the axis at this wind speed was calculated to be $156.348Nm$. We must be able to deliver a torque higher than this value at any moment.

Results

The following graphs show us the stress on different parts of the system, for various wind speeds, with respect to the elevation angle.
All our torque specifications related to wind can be read on these following graphs obtained via MATLAB (see Appendix, Wind MATLAB code).

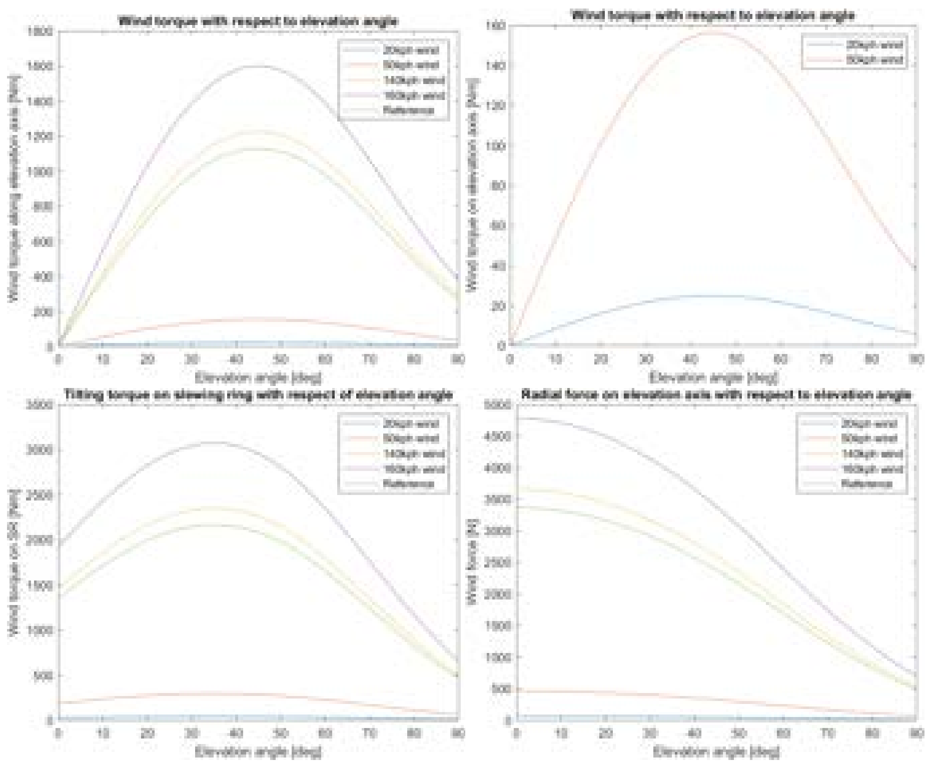

Figure 4: Stress on different parts of the system, for various wind speeds, with respect to the elevation angle

2.2.3 Motor torque

The torque applied on the motors is the sum of wind and idle torque coming from friction in the mechanical transmission.

For the elevation axis, an idle torque of $1.1Nm$[8] is caused by friction in the gearbox. The final torque applied on the elevation motor is therefore :

$$\frac{windforces}{gearratio} + idletorque = \frac{157Nm}{140} + 1.1Nm = 2.22Nm$$

For the azimuth axis, there is a friction torque in the slewing ring dependant of the stresses applied on the ring. With the user manual, we calculated this torque to be $56Nm$[9]. An additional $0.08Nm$ comes from friction in the planetary gearbox[10].

The final torque on the motor is then :

$$\frac{windforces + idle_{slewing}}{gearratio} + idle_{planetary} = \frac{157Nm + 56Nm}{227} + 0.08Nm = 1.02Nm$$

Note
These calculations use data from the specific parts we have chosen later in this report. They are in this section to establish the specifications our motors must comply to, given our engineering specifications and the rest of the system.

3 Design

3.1 General

The end project was to design the Small Radio Telescope. Three main poles then worked hand in hand to come up with the general structure, the antenna and the motorisation. Here is an overview of the final look the SRT will have once assembled.

Figure 5: Design of the structure of the SRT, done by Audrey Piccini of the Structure Pole

Our pole being the Motorization and Control one, we had to design the motor housing so that both azimuthal and altitude movements were made possible.

Azimuthal axis : link between the base and the motor housing of the SRT

The main challenge revolved around the azimuthal axis : we needed to link the motor housing to the base without constraining rotation while still holding a significant weight (96kg according to the Structure Pole) and resisting wind. Additionally, we had to find a place to put the encoder.

Therefore, we designed a few different solutions (see Appendix, Rejected solutions : azimuthal axis) and chose the following one :

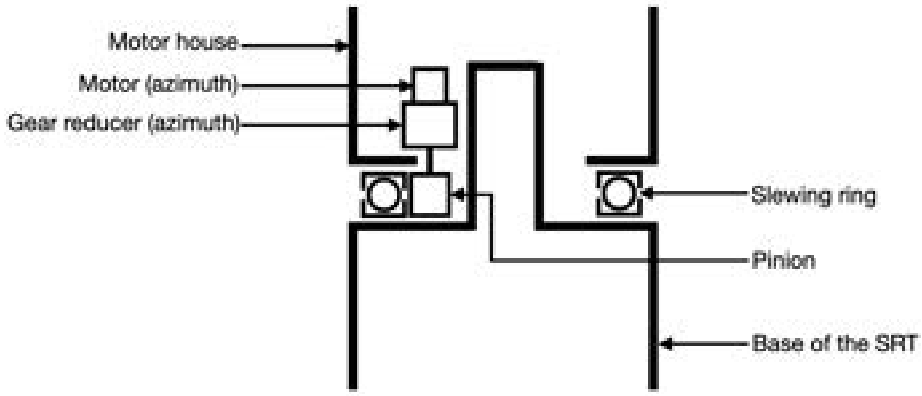

Figure 6: Final azimuthal solution to link the motor housing to the base of the SRT

Elevation axis

Once we agreed on the solution for the azimuthal axis, we had to choose how to set up the elevation axis. We envisaged two solutions (see Appendix, Rejected solution : elevation axis) and elected the following one :

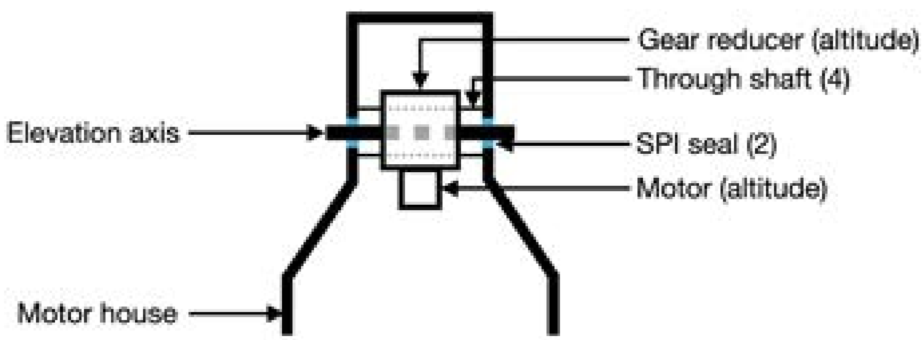

Figure 7: Final solution for the elevation axis

This solution was the most optimal as, when inquiring about our elevation gear reducer[10], we found out it could be directly attached to the elevation axis. This characteristic allowed us to eliminate the gearing wheels showed in the rejected solution (see Appendix, Rejected solution : elevation axis).
In addition, the elevation axis doesn't need ball bearings to be able to rotate as well as being fixed in the motor house walls since it is the gear reducer which is fixed to these walls thanks to the four through shafts, as shown in Figure 7. The justification for the lack of ball bearing is discussed in 3.3, paragraph 7.
The SPI seals are here to ensure impermeability so that no water (or other liquids) may enter the motor housing while still allowing rotation.

3.2 Motor choices

Our telescope should be able to hold still while enduring perturbations (Table 1, Ref. 0_3_1). For this reason, we decided against a synchronous motor and chose a steppermotor for its ability to hold static torque.

The microstepping technology allows steppermotors to have a movement resolution of up to 1/25600 rotation per microstep or even less. A microstepper with 25600 resolution needs a gear reducer of ratio of at least $26 : 1$ to achieve an antenna movement resolution of $2arcsec$ or less (see Table 1, Ref. 0_1_3).

Because of the high torque which must be held, we decided on reductions of $140 : 1$ and $227 : 1$ (see sections 3.3 and 3.4). To get a maximum speed of $1.5deg/s$ (see Table 1, Ref. 0_3_2), we need a motor rotation speed of less than $1rpm$ for 25600 resolution.

Furthermore to overcome both friction torque and wind torque, the elevation motor and the elevation motor need to be able to hold torques of $2.02Nm$ and $1.02Nm$ respectively, as established in section 2.2.3. In addition, the altitude motor must have a mounting plate larger than a $90mm$ square in order to be mounted in the gear reducer[8].

3.3 Elevation Axis

We decided to use the NVH 063 MF gear reducer from Wittenstein. This gear reducer allows to operate with a torque of $365Nm$ at a reduction ratio of $140 : 1$. It is hollow-shaft, which means we can just pass the axis through. We note an idle torque of $1.1Nm$ with this gear reducer that must be accounted for.

To connect a motor directly on the gear reducer, the motor needs to be of a minimum size of $90mm$. The M1433010 from LAM Technologies is a NEMA 42 motor that can deliver up to $14.4Nm$ of torque for 10 amps of current[11]. Supplying it with only 4 amps of current (the limit of our drivers, see Appendix, Table 4) will make it deliver up to $5.76Nm$ of torque. This configuration satisfies the requirements established in section 2.2.3.

The motor can be connected directly to the gear reducer. The fixation is strong enough to hold the weight of the motor. We can therefore fix the gear reducer to the motor housing and let the motor weight be held by its fixation to the gear reducer.

The transmission play is $10arcmin$. As the encoder is placed after the gear reducer NVH063, the movements of the dish within this play is measured, and it therefore falls under Table 1, Ref. 0_1_1_1, and does not violates it.

The axial load limit for the output shaft is $8250N$, its radial load limit is $6000N$ and its tilting torque limit is $843Nm$, all satisfying Table 1, Refs. 0_4_4, 0_4_3 and 0_4_5 respectively.

The movement within the play in the tilting direction is not measured by our encoders. It therefore falls within Table 1, Ref. 0_1_2_1. The gear reducer hollow shaft is $L = 115mm$ wide. The advised coupling tolerance is H6/h5, resulting in a max play of $\delta = 22\mu m$ for a $28mm$ diameter shaft[12]. The max angle play is therefore $\arctan(\frac{\delta}{L}) = 0.658'$. We see that it has a minimal implication on mechanical play, and therefore does not lead to a violation of Table 1, Ref. 0_1_2_1.

As radial, axial, and tilting load are easily withstood by the gear reducer by itself, and as play in tilting direction is kept to a minimal, no additional bearings are required for the elevation axis.

The encoder's ring can simply be fixed to the axis while the reading head can be fixed on the motor housing. Rotating seals (SPI seals) on each side of the axis will ensure water and wind do not pass through inside the housing.

3.4 Azimuthal Axis

The slewing ring 02-0245-00 from Rollix will make the connection between the base of the structure and the motor housing. With the adapted pinion, we can obtain a reduction of 3 : 17. From the slewing ring manual[9], we calculate that the idle torque is $55Nm$ (see Appendix, Calculation of slewing ring safety coefficient). We make sure to include it when calculating the engineering specification of Table 1, Ref. 0_4_2. Additionally, we connect the pinion to a NP015 gear reducer from Wittenstein, which has a reduction of 1 : 40 and a limit torque of $56Nm$[10]. We take note of an idle torque of $0.08Nm$.
This configuration brings a total reduction of 3 : 680. The gear reducer allows us to connect a NEMA 23 motor. The M1233051 from LAM Technologies can deliver a torque of $1.4Nm$ for a current of 3 amps[13], thus satisfying what was calculated in section 2.2.3.

The play on the transmission axis between the ring and the pinion is very hard to estimate theoretically. Tests must be conducted once we receive the components.
The total play along the axis is :

$$play_{ring} + \frac{play_{reducer}}{ratio_{ring}}$$

Once all the data available by empirical testing, we will be able to conclude if we respect Table 1, Ref. 0_1_1_1.

A study of our application, in addition with the slewing ring manual[9] shows that we have a wide margin within the mechanical limits of the slewing ring, therefore respecting Table 1, Refs. 0_4_6, 0_4_7 and 0_4_8.
Moreover, according to the user manual, play in the tilting direction is minimal, no violation of Table 1, Ref. 0_1_2_1 is therefore expected.

An extension of the base through the bottom of the housing will be a static point on which we can fix the encoder's ring. The readhead will be fixed on the motor housing, and thus rotate around the ring with the housing.

3.5 Electrical Circuit

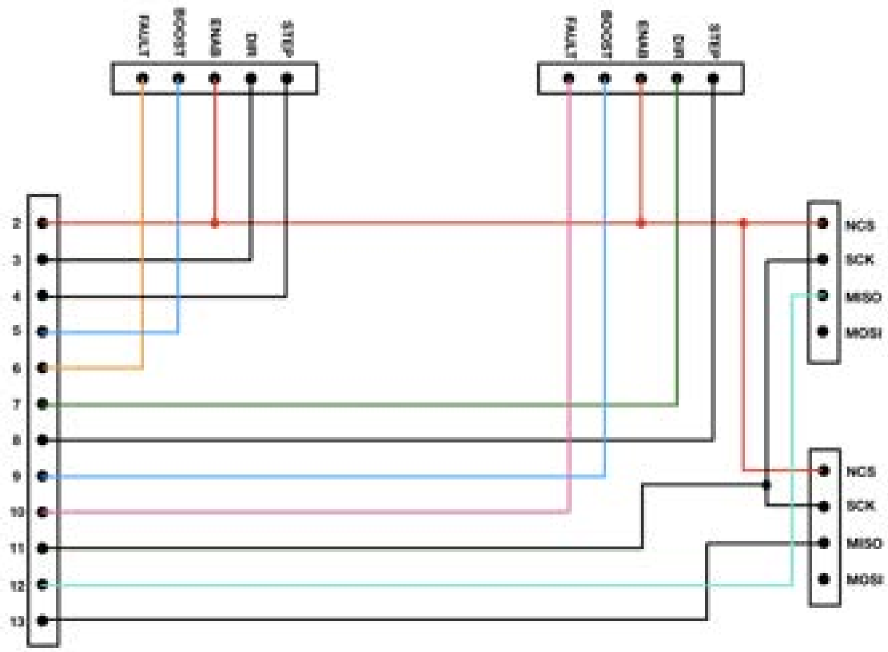

Figure 8: Electrical circuit configuration

Steppers (rectangles at the top)

- STEP : does a step when receiving an impulse
  Separated as our motors will not make the same steps.
- DIRECTION : the state (high/low) gives the direction of the step (right/left)
  Separated as the direction of the steps our motors will make can be different.
- ENABLE : if high, allows steps to be made
  Branched together to turn both motors on simultaneously.
- BOOST : if high, reduces the input current
  Separated to allow better versatility if we decide to implement a more complex control algorithm.
- FAULT : output, if high it indicates a problem

Encoders (rectangles on the right)

- NCS : similar to enable
  Branched together with the ENABLE of each stepper so that everything is turned on simultaneously.
- SCK : clock
  Branched together to be able to read both encoders simultaneously.
- MISO : acquiring data
  This is an output.
- MOSI : Configuration and calibration.
  This does not need to be connected during use.

Micro-controller (rectangle on the left)

- Teensy 4.0
  High frequency micro-controller (see 5.1)

3.6 Motor Housing

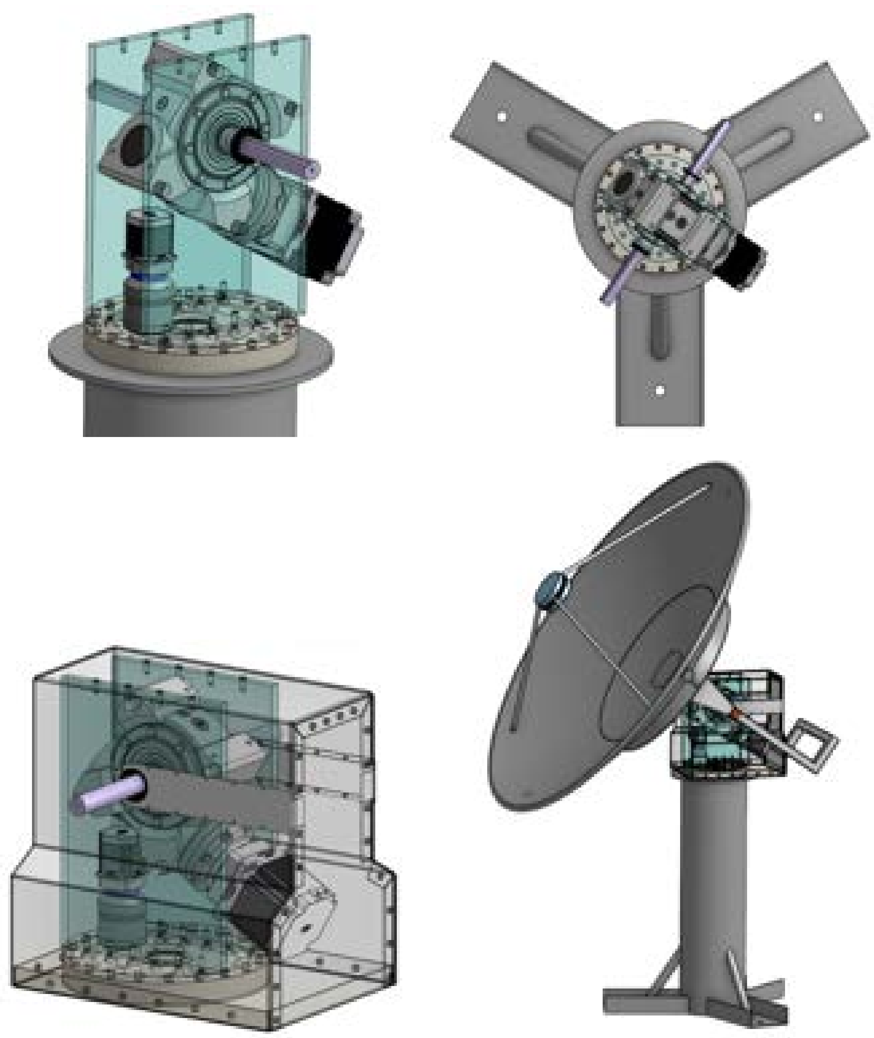

Figure 9: Design of the motor housing, done in collaboration with Audrey Piccini of the Structure Pole

This interdisciplinary project gave us the opportunity to collaborate with other students as Audrey Piccini, responsible of the Structure Pole. This motor housing was designed by her in order to fit our requirements and needs, using the solution discussed in 3.1.

3.7 Assessment of validity regarding engineering specifications

Table 2: Control Pole Engineering specifications assessment of validity

Ref.	Title	Description	Assessment of validity
0	Main function	The control system shall point an antenna dish towards a desired space object or earth satellite	
0_1	Precision and accuracy requirements	The control system shall ensure accuracy standards	Will be measured on the built SRT
0_1_1	Tracking accuracy	At all time during tracking, the antenna dish shall be closer than [42] arcmin to the target	Will be measured on the built SRT
0_1_1_1	Total mechanical play	The mechanical play in the system shall stay under [15] arcmin	Will be measured on the built SRT
0_1_2	Pointing knowledge	At all time the system shall be aware of its position to an accuracy of [15] arcmin	Will be measured on the built SRT
0_1_2_1	Unmeasured mechanical play	Play in the mechanical system that cannot be measured shall be limited to an angle of [5] arcmin	Will be measured on the built SRT
0_1_3	Movement resolution	The smallest unit of movement possible for the motor system shall be finer than [2] arcsec	Elevation: The total reduction is [140], which means that with a microstepping resolution of [6400] rot^{-1}, the movement resolution is [1.45] arcsec Azimuth: The total reduction is [227], which means that with a microstepping resolution of [3200], the movement resolution is [1.78] arcsecs
0_1_4	Measure resolution	The smallest unit of movement measurable shall be finer than [2] arcsec	The position on our encoder is encoded on 20 bits according to RLS[14]. That translates to a resolution of $\frac{360°}{2^{20}} = [1.24] arcsec$
0_2	Range requirement	The whole sky half sphere, solid angle $[2\pi]$ rad^2 shall be accessible	Verified by proxy, as 0_2_1 0_2_2 0_2_3 all are

Continued on next page

Table2 – continued from previous page

Ref.	Title	Description	Assessment of validity
0_2_1	Axis	The movement shall be possible along 2 axis : azimuth and altitude	Verified by our general design, see section 3.1
0_2_2	Azimuth range	The movement along the azimuth axis shall cover at least a full circle, or [360] deg	Verified by our general design, see section 3.1
0_2_3	Altitude range	The movement along the altitude axis shall cover at least [90] deg starting at the horizon	Verified by our general design, see section 3.1
0_3	Speed requirements	The system shall endure the various tracking speed required by the antenna	Verified by proxy, as 0_3_1 0_3_2 are
0_3_1	Min track speed	The system shall be able to hold still	The motors we selected are able to hold a still position, see section 3.2
0_3_2	Max track speed	The system shall ensure a tracking speed over [1.5] deg/s on both axis	For a microstepping resolution of [6400] rot^{-1} and a reduction of [227], a stepping frequency of [6054] Hz is required to move at the desired speed. This will be easily achieved with our microcontroller.
0_4	Structural integrity	The mechanical system shall endure all stress during operation, or at rest	Will have to be tested on the built SRT. First data indicates it is respected.
0_4_1	Motor torque along altitude	The altitude axis transmission shall ensure a torque of [157] Nm	Our motor can deliver a torque of [6.05] Nm. After the reduction of [140] and removing the idle torque of [1.1] Nm, the motor torque can go up to [693] Nm.
0_4_2	Motor torque along azimuth	The azimuth axis transmission shall ensure a torque of [157] Nm	Our motor can deliver a torque of [1.4] Nm. After the reduction of [227] and removing the idle torque of the gear reducer of [0.08] Nm and the idle torque of the slewing ring of [56] Nm (post-reduction), the motor torque can go up to [243.6] Nm.
0_4_3	Radial forces on altitude axis	The altitude transmission shall support a radial force of [857] N	The gear reducer can take [6000] Nm radial force.

Continued on next page

Table2 – continued from previous page

Ref.	Title	Description	Assessment of validity
0_4_4	Axial forces on altitude axis	The altitude axis transmission support an axial force of [493] N	The gear reducer can take [8250] Nm axial force.
0_4_5	Tilting forces on altitude axis	The altitude axis transmission support a tilting torque of [267] Nm	The gear reducer can take [843] Nm radial force.
0_4_6	Axial forces on azimuth axis	The azimuth transmission shall support an axial force of [700] N + weight of motor housing	By performing the analysis given in the user manual of the slewing ring[9], we find that our security factor is 22.5, we are therefore easily within the ring specification. Calculation are detailed in the Appendix, Calculation of slewing ring safety coefficient.
0_4_7	Radial forces on azimuth axis	The azimuth transmission shall support radial forces of [493] N + drag force of the motor housing	Respected, see 0_4_6
0_4_8	Tilting torque on azimuth axis	The azimuth transmission shall support tilting torque of [464] Nm + motor housing forces	Respected, see 0_4_6
0_4_9	Cable preservation	The movement shall be limited depending on the design to prevent excessive cable tangling.	Software limits the movement to [+720,-720] degrees.
0_4_10	Thermal	The components shall stay within their temperature operational domain (especially motors and electronics)	Our power supply has a rated power of [300] W. This is therefore the most power we can dissipate as heat. The thermal resistance of the motor housing is $R = \frac{e}{kA} + \frac{2}{hA}$, which is approximately [0.138] K/W. The largest temperature delta between the outside and the motor housing is therefore $\Delta T = 41K$. Even with high outdoor temperature, this would still be within the functioning range of our material, and we remind this is calculated as a worst case where every watt the power supply can provide is wasted as heat.
0_5	Environment tolerance	The system shall be able to operate long-term outdoor in Lausanne, Switzerland	0_5_3 needs to be addressed

Continued on next page

Table2 – continued from previous page

Ref.	Title	Description	Assessment of validity
0_5_1	Wind resistance	The system shall function properly during winds of up to [50] km/h during operation, and withstand a [0.9]kN/m$\hat{2}$ of dynamic pressure at rest	As seen in the assessments of points 0_4, we will function under these conditions.
0_5_2	Rain resistance	The system shall withstand heavy rain without damage	SPI seals will guarantee that there will be no leakage allowing rain to flow inside the motor housing.
0_5_3	Humidity control	Humidity in the motor housing shall not disturb our electronics	This question was not addressed due to missing time. The best but most complicated solution would be sealing the motor housing with SPI seals as best we can and using an air dryer to keep the internal humidity low.
0_6	Interfaces and communication	The motorization system shall be integrated in the system and communicate with other poles	Interfaces were taken into consideration and are sufficient for our use. 0_6_3 is planned to be improved.
0_6_1	Structure base link	The motor housing shall be able to transmit the stresses of 0_4_6, 0_4_7, 0_4_8 to the structure base	Respected, see 0_4_6
0_6_2	Power supply	[230] V AC shall be provided in the motor box to supply on board systems	The power supply takes [230] V AC.
0_6_3	Communication between on-board system and distant user	Data i/o cable shall be provided to on-board system to communicate with distant user	A serial interface is implemented until discussions with EPFL allow us to use a WiFi antenna.
0_7	Robustness and maintainability	The system shall as much as possible be able to face unforeseen issues. Repairs and maintenance shall be as easy as possible	0_7_1 needs to be addressed.
0_7_1	Mechanical stop for altitude	[+0/+90] deg limit set in 0_2_3 shall be mechanically backed up	Not designed yet.

Continued on next page

Table2 – continued from previous page

Ref.	Title	Description	Assessment of validity
0_7_2	Incoherent inputs handling	System shall handle incoherent inputs	Software currently handles incoherent coordinate inputs. Incoherent time inputs don't need particular attention, as they will induce error in tracking but will not cause damage to the system.
0_7_3	Protection against external disturbance	System shall turn off when external disturbance torque exceeds [157] Nm	This feature was not implemented. Currently the user has to make sure meteorological conditions are fine before using the SRT.

4 Physical Modelling

We used Simulink and Simscape, as well as analytical frequency response analysis to simulate the dynamics of our system at various levels.

- We calculated the transfer function (TF) of our systems along our two axis, assuming linear behavior.
- We implemented this TF in a numerical simulation and applied perturbations on the system.
- We programmed an electromechanical simulation using Simscape to identify the dynamics induced by our stepper motors.

4.1 Transfer function and frequency response

For this part we made some assumptions in order to analyse the system with linear time invariant (LTI) tools.
This is the list of hypotheses :

- The motor behaves as a synchronous motor
- The mechanical transmission is ideal
- In order to reduce the order of the system, we neglected the impedance of the motor winding
- The inertia of the system is considered constant, whatever the position of the system
- Friction is assumed to be viscous only
- Wind forces are modelled as an oscillatory perturbation

Equation of the electrical system :

$$u_c = u_i + Ri$$

Equation of the mechanical system :

$$J_{eq}\ddot{\theta_p} + c\dot{\theta_p} = M_p$$

Electromechanical equations :

$$M_m = ki$$

$$U_i = k\dot{\theta_m}$$

As we have perfect transmission between the motor and the parabola, with gear ratio r :

$$M_p = M_m * r$$

$$\dot{\theta_p} = \dot{\theta_m}/r$$

Combining the equations found above, we obtain :

$$J_{eq}\ddot{\theta_p} + c\dot{\theta_p} = rk(\frac{u_c - kr\dot{\theta_p}}{R})$$

$$J_{eq}\ddot{\theta_p} + (c + \frac{k^2r^2}{R})\dot{\theta_p} = \frac{kr}{R}u_c$$

Written in the Laplace domain, assuming zero initial conditions :

$$(J_{eq}s^2 + (c + \frac{k^2r^2}{R})s)\Theta = \frac{kr}{R}U_c$$

$$\Theta(s) = \frac{kr/R}{J_{eq}s^2 + (c + \frac{k^2r^2}{R})s)}U_c(s)$$

We can easily compute the TF, $G(s) = \frac{\Theta(s)}{U_c(s)}$:

$$G(s) = \frac{kr}{RJ_{eq}}\frac{1}{s^2 + (\frac{c}{J_{eq}} + \frac{k^2r^2}{RJ_{eq}})s}$$

Finally, the poles of this TF are :

$$0 \text{ and } -\frac{c}{J_{eq}} - \frac{k^2r^2}{RJ_{eq}}$$

- The pole on 0 is an integrator. It is expected, as we are acting on a speed control while observing a position
- The second pole is the inverse time constant $\frac{1}{\tau}$ of our motorisation system. If we plug the numerical values for our axis, we obtain $\tau_{az} = 0.0012s$ and $\tau_{elev} = 1.03 * 10^{-4}s$

Then, the time constants are very small compared to the time we have to wait in order to have an appreciable variation in our input. This comes from the fact that, after reduction, our motors can apply very large torques to a modest inertial mass. Therefore, we can study the system with a constant input as a reference for future iterations without any issues.

However, there already is an integrator in the system, thus an integral controller is not necessary in order to follow a step. However, we may want to add one later in order to reject constant external perturbations.

For now, we want to study the reaction of the system to an oscillatory perturbation. An integral term is therefore not necessary and would have minimal impact. For this simple controller, we have :

$$KG = \frac{K_pkr}{RJ_{eq}}\frac{1}{s^2 + (\frac{c}{J_{eq}} + \frac{k^2r^2}{RJ_{eq}})s}$$

Our proportional controller transforms the integrator into a low frequency pole.

We model the wind as a $1rad/s$ oscillatory torque, with a magnitude of $157Nm$ (see table Table 1, Ref. 0_4_1).

By plotting the Bode plots, it is observed that with a synchronous motor, we would need a very large proportional gain unless we have very high frequencies of oscillation for the wind or very high gain in the controller. A gain that is too high would drive the motors outside of their torque limits, this is therefore not applicable.

As wind oscillations are kind of slow, we hoped that they could be better rejected with an integral term in our controller. We therefore tried to include a PI controller :

$$K(s) = K_p(1 + \frac{1}{T_i s}) = K_p \frac{s + \frac{1}{T_i}}{s}$$

However, we must have :

$$1/T_i >> \omega_w$$

(where ω_w is the frequency of the oscillation induced by the wind in $[rad/s]$) in order to efficiently reject wind perturbation.
Empirically, this means :

$$T_i > \frac{100}{\omega_s}$$

to decrease perturbation to a satisfactory level. Satisfying this condition would lead to an integral gain which would be too high, thus destabilizing the system.

We saw that, coupled with a linear system, neither a proportional controller nor an integral controller are satisfactory in order to reject wind perturbation enough for our needs. We therefore turn to a stepper motor, which are able to hold high torque while remaining static. The system would however no longer be linear time invariant.
A more complex model must therefore be implemented, as we will see in the next part.

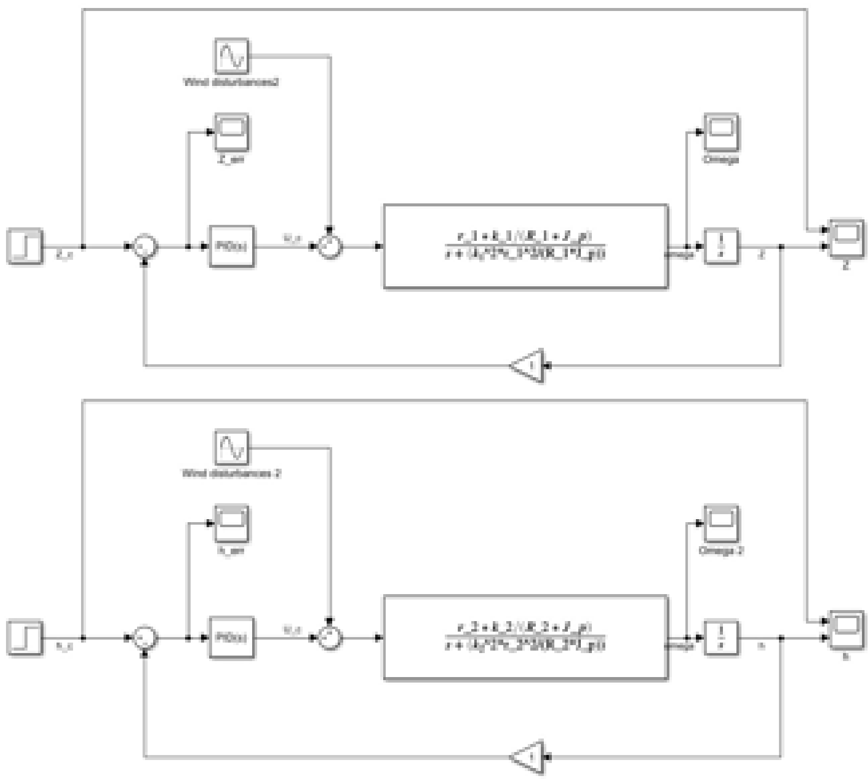

Figure 10: Synchronous Motor Block

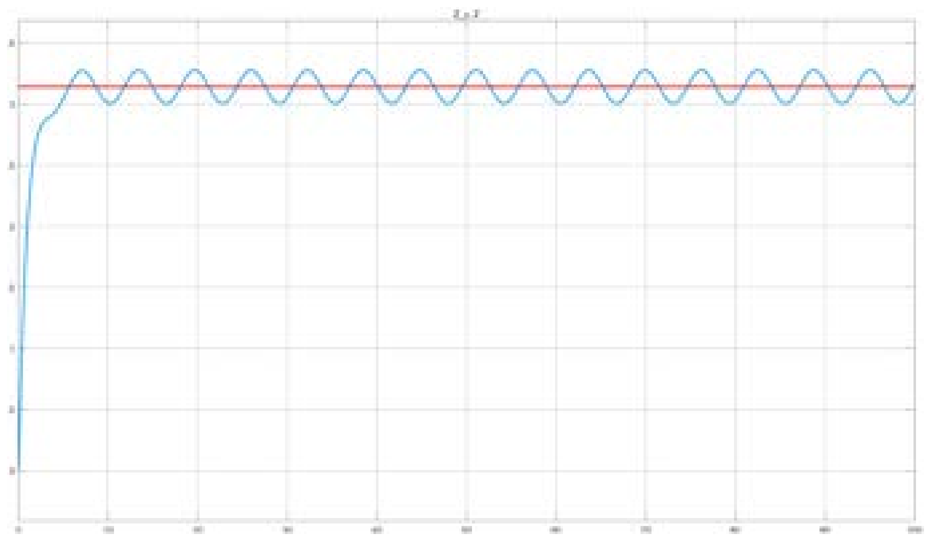

Figure 11: Synchronous Motor Perturbation Azimuth

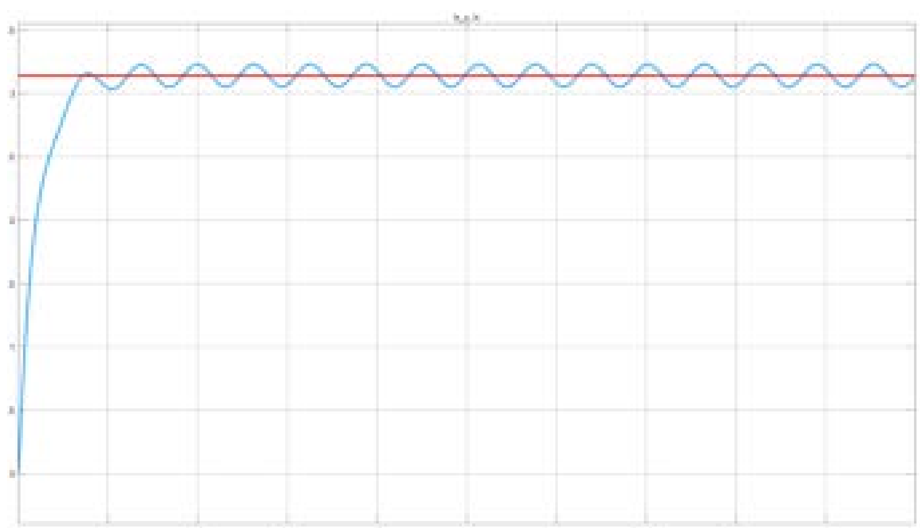

Figure 12: Synchronous Motor Perturbation Elevation

The figures above show that there is an unacceptable perturbation (the amplitude of the oscillation is too high) for our requirements, and this happens for any reasonable value we can use for the gain. Hence, a synchronous motor is not a viable option for our telescope.

4.2 Simulink

We modeled a stepper motor system using simscape tools. We chose the parameters of the system to fit the specification of the real life material we ordered.

We applied two different kinds of perturbation in order to evaluate the performances of the system. First, a 500 step offset is applied as a perturbation. This will evaluate the ability of the system to overcome a deviation. Then, we apply an oscillatory torque to model wind forces and ensure that such disturbances will not compromise observation.

4.2.1 Model 1 : Step motor Closed loop (azimuth)

PI : $P = 5$, $I = 2$

The equation of a PI controller implemented in Simulink is

$$K = P * (1 + I\frac{1}{s})$$

Perturbation : 500 Steps (step motor has very abrupt upwards steps so the D term would be huge and it would cause chaos in the control)

We used Simscape tools in order to have such a model.

Here below are two examples with different integrator values for a step perturbation. We can see that with a higher I value the time to reject the disturbance is shorter, but a smaller I would allow a smoother, more linear-behaving response.

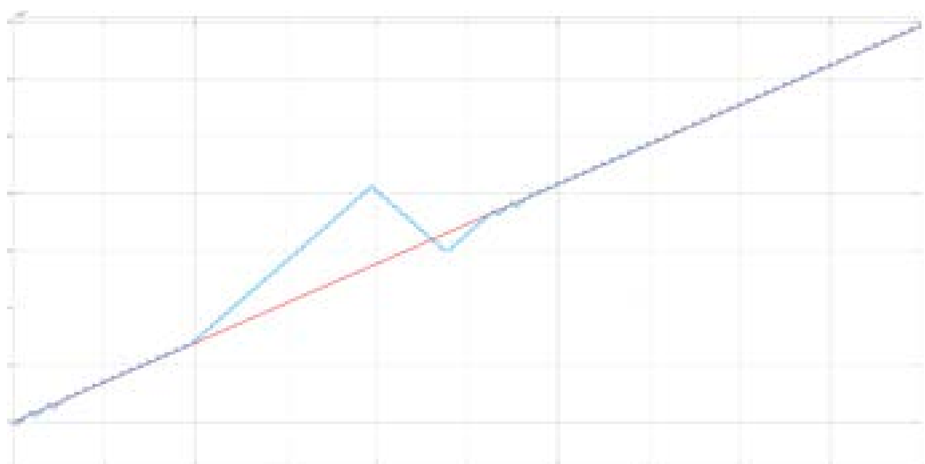

Figure 13: View of the signal $P = 5$, $I = 20$

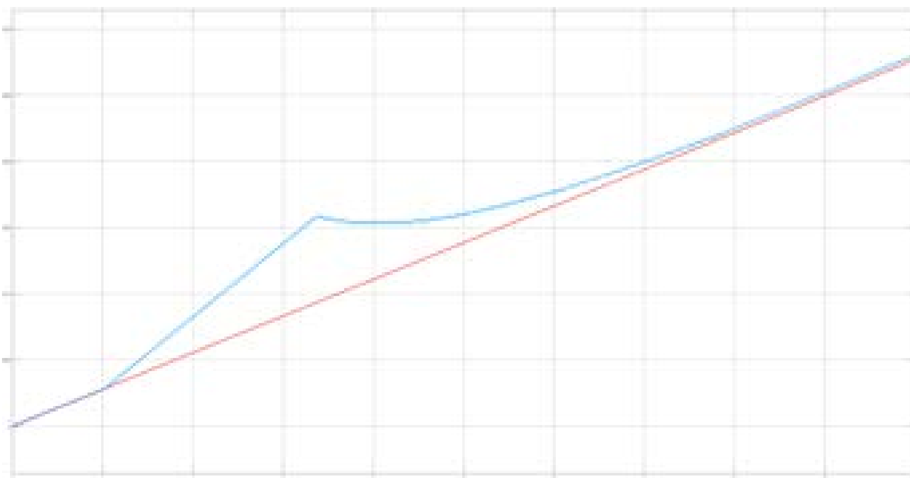

Figure 14: View of the signal $P = 5$, $I = 2$

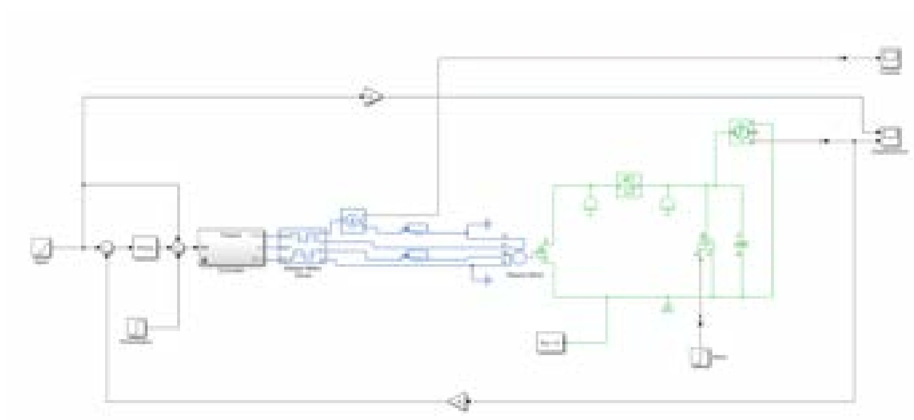

Figure 15: Block diagram for closed loop step motor

We can observe above that a closed loop implementation leads to satisfactory behaviour.

4.2.2 Model 1 : Step motor Closed loop (elevation)

Similar behaviour to Step motor Closed loop (azimuth), section 4.2.1.

4.2.3 Open Loop Step comparisons

Figure 16: Azimuth Step

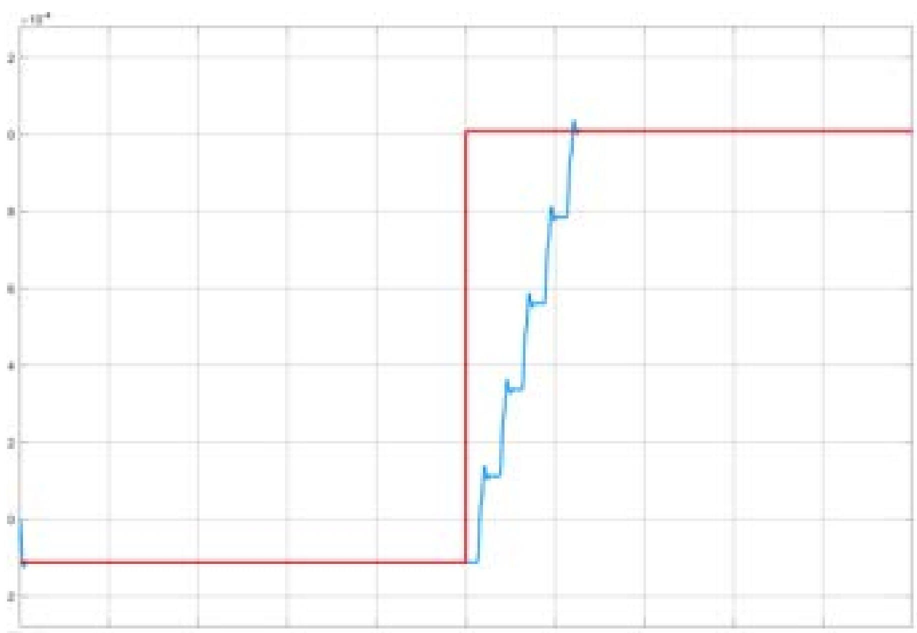

Figure 17: Elevation step

We observe a lot of oscillations in position before settling, especially in the azimuth axis, where the motor is physically smaller, and therefore the damping is less important. We will mitigate this effect by using microstepping.

4.2.4 Closed loop Step Motor with oscillatory perturbation

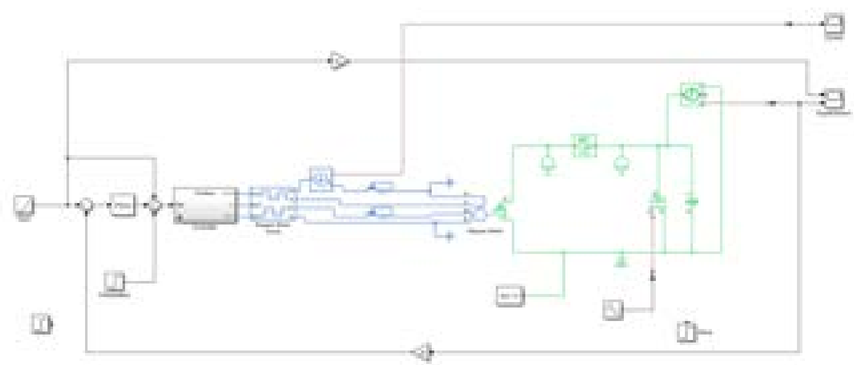

Figure 18: Block diagram

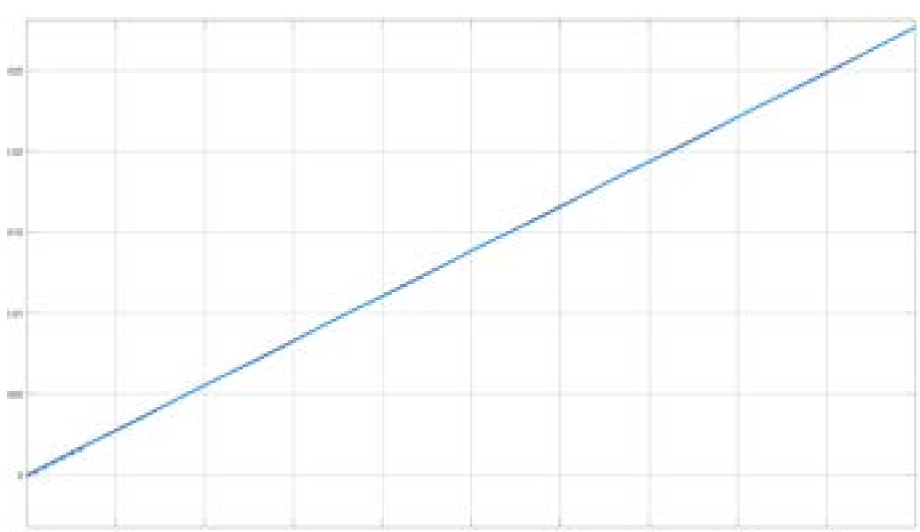

Figure 19: Azimuth

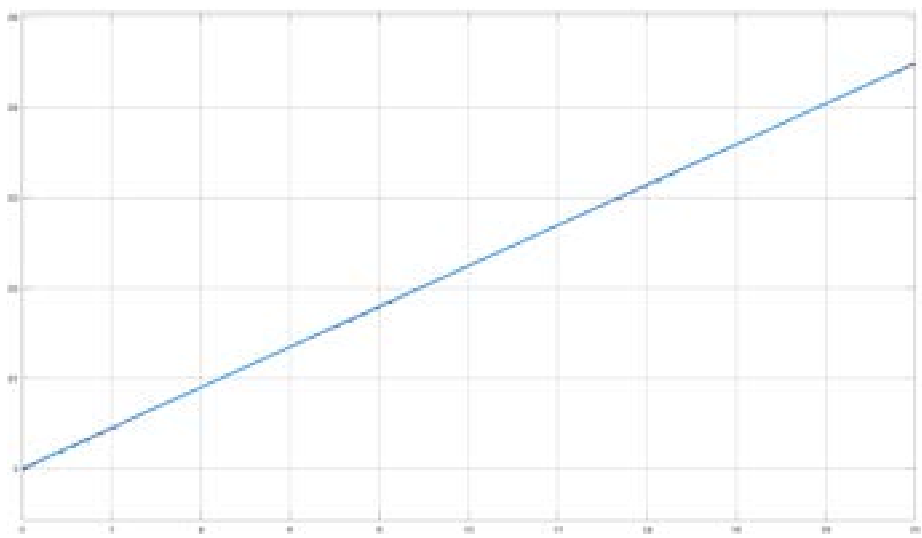

Figure 20: Elevation

We observe that we are capable of following a reference with minimal deviation despite strong perturbations. These graphs represent our system following a reference, with a wind perturbation of $160Nm$ with a frequency of $2rad/s$.

Overall, our simulations show that a stepper based build is highly robust to external perturbation, either static or oscillatory. Moreover, static errors, as small as our stepping would allow, can be maintained with a very basic controller, even with perturbation torque near the theoretical limit.

5 Implementation

5.1 Choice of micro-controller

Our micro-controller was chosen so that it was able to accomplish four different tasks simultaneously. These tasks are the following :

– stepping

– reading the encoders

– converting coordinates from equatorial to horizontal

– choosing the direction and number of steps

Our first implementation was done using an Arduino Uno micro-controller (16MHz of clock speed) which we had laying around. After implementing an asynchronous interruption programming, we noticed this micro-controller had a hard time going over 15kHz microstepping frequency. This was shown by the motor still making noise but not moving when going at higher frequencies (as if the slope of the electrical signal when going from low value to high value was too slow) and also by the Arduino Uno not being able to do the other needed operations correctly.

Finally, knowing we were controlling only one microstepper (therefore only one motor) and we already found ourselves limited, we chose to use a Teensy 4.0 of 600MHz clock speed.

Figure 21: Arduino Uno (left) and Teensy 4.0 (right) micro-controllers

5.2 Micro-controller Code

The implemented code uses two timer interruptions. The first timer interruption will check a *steps* variable, and if positive, will execute a step and decrement the *steps* variable. This allows a fast execution of steps only when needed. The second timer interruption will fetch the positions measured by the encoders and store the results in global variables. This allows a controllable frequency at which we re-actualize the measured position.
Additionally, an interruption on the serial port allows us to communicate with the system.

The main program loop will calculate the position to point to according to given parameters and current time. It will then compare the objective to the position measured by the encoders. Afterwards, it will set the number of steps and direction needed for each motor to reach its goal position.
To optimize the calculation of coordinates, we tried two methods :

- The first is by implementing fixed point calculations. The sines and cosines were implemented by a Taylor expansion to the sixth term to prevent sacrificing accuracy.
  After testing, this method doesn't allow us to gain time with our capabilities.

- The second method is pre-calculating sines and cosines.
  After trying to optimize memory usage, we determined we would not be able to reduce memory usage enough without sacrificing accuracy.

Eventually, we decided that normal float operations, done by the FPU of our micro-controller, will allow us to operate at our desired frequency.

5.3 Assessment of validity and precision

To confirm that our code gives us a correct position, a LabVIEW program compares our output to positions calculated by the standard Python library *astropy*.
After testing various sky objects, the object that gives us the highest theoretical positional error is Vega with a maximal error of $33.4 arcsecs$ and a RMS error of $32.4 arcsecs$ (simulated over 24hours from 2021/5/8 0h00 to 2021/5/9 0h00).

6 Limitations

6.1 Design limitations

Humidity

We currently cannot guarantee that the inside of the motor housing will stay dry enough to prevent any risk, notably to the encoders. The SPI seals will prevent any influx of rainwater or other fluids, but condensation might become a problem depending on external temperature. We will have to make measurements once the SRT is assembled to ensure the components won't get damaged.

Time

Currently, our micro-controller gets the current time in the request to track an object and the knowledge of time will be via relative calculations.
Depending on the interface with the user (potential delays between the time it makes the request and the time when the system accepts the request), we currently cannot guarantee a precise knowledge of the time. Additionally, any slight power source problem will make the micro-controller loose track of time. The best way to solve this problem is to put an RTC under battery or PSU. After adequate calibration, accurate knowledge of current time is guaranteed even during not too long losses of power.

6.2 Administrative limitations

This project was launched again fairly from scratch at the beginning of this semester. Therefore, we suffered from a lack of investment, which had a considerable impact on our capacity to order our required high-precision mechanical pieces.
Moreover, the Covid-19 crisis generated an electronics shortage. Thus, receiving components such as the Teensy 4.0 took more time than expected, preventing us from fully testing it.
Furthermore, due to delays in equipment delivery (in particular the encoders), we were unable to test our motor and driver and to compare it to our simulations. Therefore, we cannot be certain of how accurately the simulations predict our system's behavior.

However, as we will discuss in the Next Steps section, these create interesting challenges for the next semester, during which we plan to carry on this project.

7 Conclusion

Throughout the semester, we were able to design simulations to predict our motors' behaviour and implement their positioning algorithm. Also, we determined which mechanical and electronic pieces were needed in order to fit our specifications, trying to balance low budget and high precision.

However, given our current budget we were unable to start the motor housing, as well as the SRT, construction which will then take place later in 2021.
We also were unable to compare our simulations to the behaviors of our system due to delays with the production of the encoders.

Next steps

This project is not yet over and many steps in which we hope to participate are still to be completed.

- Piece testings
  Before assembly, we will test all our pieces separately to check that they behave as expected.
- Addressing the humidity issue
  Seals prevent macroscopic droplets from entering, but not humid air. We will study the option of placing a fan in the motor housing to prevent this.
- Construction
  Every pole has a proper design ready to be implemented. We should be able to proceed to construction early next semester once all the pieces are in our possession.
- Time
  We have to find a more efficient way to stock/keep the time so that if there is a shutdown in our system, it will not re-initiate at zero. A possible solution would be to install a RTC connected to a battery, as mentioned in section 6.1.
- Interface
  Add a better communication interface.
- Final testings
  Once the SRT is fully assembled, tests will have to be done to ensure its functionality.

Bibliography

[1] E. Callista. (). "Conception of a small radio telescope (srt) at epfl," [Online]. Available: `https://callistaepfl.ch/projets-events/`.

[2] U. S. N. Observatory. (). "Approximate sidereal time," [Online]. Available: `https://web.archive.org/web/20190524114447/https://aa.usno.navy.mil/faq/docs/GAST.php`.

[3] Wikipédia, *Système de coordonnées équatoriales — wikipédia, l'encyclopédie libre*, [En ligne; Page disponible le 9-janvier-2021], 2021. [Online]. Available: `http://fr.wikipedia.org/w/index.php?title=Syst%C3%A8me_de_coordonn%C3%A9es_%C3%A9quatoriales&oldid=178604600`.

[4] Wikipedia contributors, *Horizontal coordinate system — Wikipedia, the free encyclopedia*, [Online; accessed 28-May-2021], 2021. [Online]. Available: `https://en.wikipedia.org/w/index.php?title=Horizontal_coordinate_system&oldid=1023938049`.

[5] Meteoblue. (). "Climat lausanne - vent," [Online]. Available: `https://www.meteoblue.com/fr/meteo/historyclimate/climatemodelled/lausanne_suisse_2659994`.

[6] *Sia 261: Actions sur les structures porteuses*, SIA, (accessed: 18.05.2021).

[7] C. Peat. (). "Heavens-above: Iss - orbit," [Online]. Available: `https://www.heavens-above.com/orbit.aspx?satid=25544`. (accessed: 13.05.2021).

[8] NVH063, *Alpha value line réducteurs à roue et vis sans fin nvh / nvs*, Wittenstein, (accessed: 03.06.2021).

[9] 02-0245-00, *Couronne d'orientation, roulements spéciaux*, Rollix, (accessed: 03.06.2021).

[10] NP015, *Alpha value line réducteurs planétaires np / npl / nps / npt / npr*, Wittenstein, (accessed: 03.06.2021).

[11] M1433010, *Nema 42/2 phase 1.8° step angle/hybrid stepping motor*, LAM Technologies, (accessed: 03.06.2021).

[12] *Extrait de Normes 2018 pour écoles et professions de la mécanique*, ser. 12. SWISSMEM, 2018, ISBN: 9783038661627.

[13] M1233051, *Nema 23/2 phase 1.8° step angle/hybrid stepping motor*, LAM Technologies, (accessed: 03.06.2021).

[14] AksIM-2, *Programming aksim-2 encoders with encolink communication protocol*, RLS®, (accessed: 03.06.2021).

[15] DS1044, *Ds10 series*, LAM Technologies, (accessed: 03.06.2021).

[16] Teensy4.0, *I.mx rt1060 crossover processors for consumer products/document number: Imxrt1060cec/rev. 0.1, 04/2019*, SparkFun, (accessed: 03.06.2021).

[17] USP-350-48, *350w single output with pfc function, usp-350 series*, MEAN WELL, (accessed: 03.06.2021).

[18] Wikipedia, *Drag coefficient — Wikipedia, the free encyclopedia*, `http://en.wikipedia.org/w/index.php?title=Drag\%20coefficient&oldid=1009830709`, [Online; accessed 04-June-2021], 2021.

Appendix

Analytical confirmation of wind matlab code in zenith position for elevation axis

We can calculate the wind drag area in the shape of a circular segment with chord length c (diameter of the parabola) and height h.
We can calculate the curvature of the parabola:

$$R_c = \frac{h}{2} + \frac{c^2}{8h}$$

Each infinitesimal area of the parabola gets a force of

$$dF = \frac{1}{2} c_d dA \rho v^2$$

The corresponding torque created around the elevation axis is:

$$dT = \frac{1}{2} c_d \rho v^2 \left(L + y\right) dA$$

With L the distance between axis and parabola.
By integrating over the area we can estimate a total torque:

$$T = \int_{-\frac{\phi}{2}}^{\frac{\phi}{2}} \int_{0}^{R_c} \frac{1}{2} c_d \rho v^2 * \left(L + R_c - r\cos\theta\right) r dr d\theta \\ - 2 \int_{0}^{\frac{c}{2R_c - 2h} y} \int_{0}^{R_c - h} \frac{1}{2} c_d \rho v^2 * \left(L + R_c - y\right) dx dy$$

With:

$$\phi = 2 arcsin\left(\frac{c}{2R_c}\right)$$

$$\Rightarrow T = \frac{1}{2} c_d \rho v^2 \left[\frac{1}{6}\left(c\left(h - R_c\right)\left(2h + 3L + R_c\right) + 3\phi R_c^2 \left(L + R_c\right) - 4R_c^3 \sin\left(\frac{\phi}{2}\right)\right)\right]$$

With $L = 0.35m$, $h = 0.32m$, $c = 1.86m$, $R = 1.5144m$, $\phi = 1.32559rad$ and $c_d = 1.38$, $\rho = 1.292kg/m^3$.

Rejected solutions : azimuthal axis

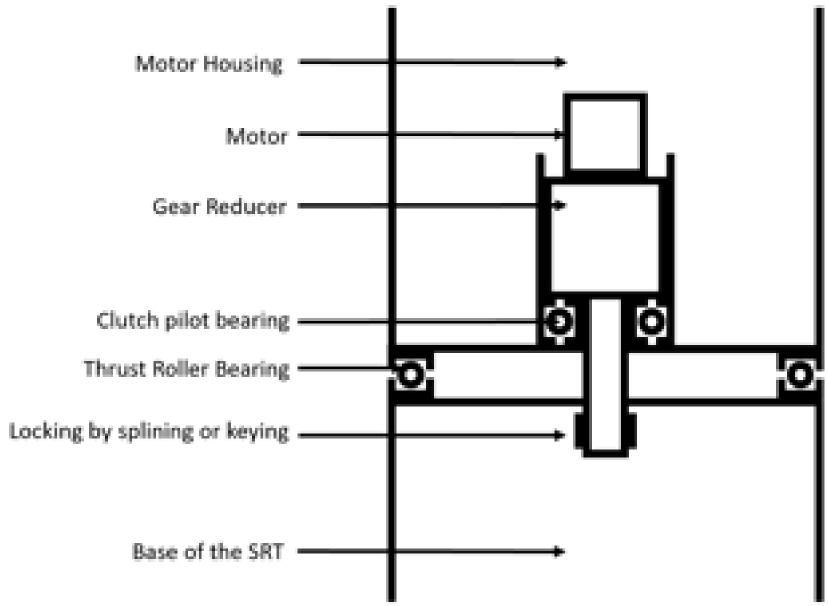

Figure 22: Rejected solution 1

In this solution, the motor is connected to a gear reducer, both inside the motor housing. The output of the gear reducer is locked to the base and the gear reducer is locked to the housing. This means that if the motor delivers a torque, the housing will start moving. A thrust bearing is placed between the housing and the base to handle the weight.
The main appeal of this solution is the price : thrust bearing are considerably cheaper than either a slewing ring or a harmonic drive.
However, in this design the mechanical linkage between the thrust bearing, the base and the housing may not be strong enough. Most thrust bearings only support compression loads, making wind tilting a major issue. Axial-radial bearings may be a solution, but that would reach the price range of a slewing ring, thus invalidating the main advantage of this solution.

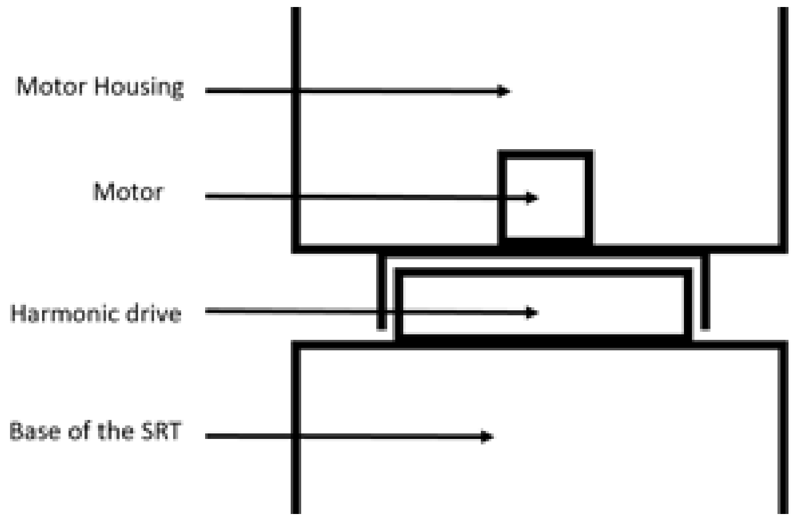

Figure 23: Rejected solution 2

This solution was considered as there is no backlash in the harmonic drive, which would have been ideal. However, the price was considerably higher than the one for the slewing ring.
In addition, as the unit is physically smaller, lever arm between the structural attachment is shorter, and therefore resistance to tilting torque is smaller. This could be problematic in regard to wind resistance. Then, even though there is backlash in the slewing ring, we estimated that it would still fit within our requirements.

Rejected solution : elevation axis

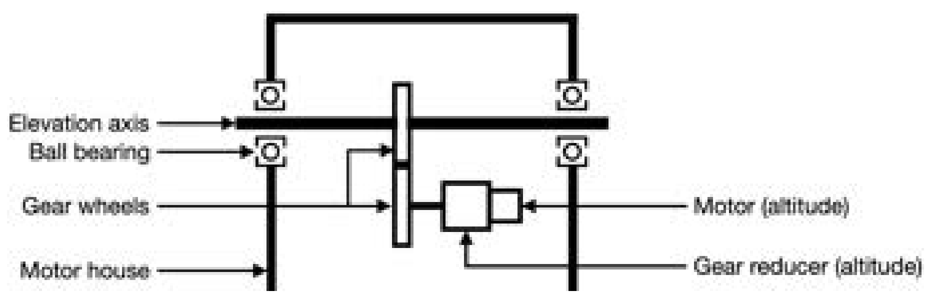

Figure 24: Rejected solution for the elevation axis

This solution was rejected as it did not optimise the placement of our mechanical pieces. Our final solution, a worm gear reducer with a pass-through axis, is all contained thus rendering the additional gears useless.

Moreover, we needed to align the two gears by ourselves, and we would not have been able to guarantee sufficient transmission accuracy. This issue is mended by the final design, as the transmission is a single unit, whose accuracy is guaranteed by the supplier.

Calculation of slewing ring safety coefficient

The user manual[9] details a procedure to calculate whether we perform or not within the specification of a given slewing ring. We have the following values for the constraints of the slewing ring :

- Axial force $F_A = 941.76N$
- Radial force $F_R = 693N$
- Tilting torque $M_T = 520Nm$

We have to use a coefficient K_R to combine the axial and radial force into a single equivalent force. In our case, the ratio between the radial force and the axial force is comprised between 0.25 and 1, so we have to take $K_R = 1.5$.
The equivalent force is then :

$$F_A + K_R F_R = 1981.26N$$

We have to apply an application coefficient, dependant of our application. The user manual describes a case for slow radars. This is close enough to our application, we will therefore take the advised K_A for this situation of 1.35.

We then have to place on a load graph the points

$$P_x = K_A * F_{eq} = 2670Nm$$

$$P_y = K_A * M_T = 702Nm$$

and check if it is below a resistance line.
Comparing our point to the line for the graph of the slewing ring 02-0245-00, we calculate that our safety coefficient is 22.

Mechanical pieces

Name	Model	Supplier
Stepper motor (azimuth)[13]	M1233051	LAM Technologies
Stepper motor (altitude)[11]	M1433010	LAM Technologies
Gear reducer (azimuth)[8]	NVH063	Wittenstein
Gear reducer (altitude)[10]	NP015	Wittenstein
Slewing ring[9]	02-0245-00	Rollix®

Table 3: Mechanical pieces

Electronic pieces

Name	Model	Supplier
Microstepping drivers (azimuth and altitude)[15]	DS1044	LAM Technologies
Encoders (azimuth and altitude)[14]	AksIM-2	RLS®
Micro-controller[16]	Teensy 4.0	SparkFun
Power supply[17]	USP-350-48	MEAN WELL

Table 4: Electronic pieces

Wind MATLAB code

The numerical value that were chosen where the following :

- drag coefficient $C_d = 1.38$. Drag coefficient of a small sphere portion [18]
- air density $\rho = 1.292\frac{kg}{m^3}$. Dry air density at $T = 0°C$, $p = 1013hPa$.
- distance between the elevation axis and the bottom of the antenna dish $L = 35cm$, given by the structure pole
- distance between the elevation axis and the slewing ring plane $L1 = 40cm$, given by the structure pole
- radius of the antenna dish $R = 0.93m$, given by the system engineering pole
- depth of the antenna dish $h = 0.32m$, given by the system engineering pole
- wind speed of $v = 50km/h$ from Table 2, Ref. 0_5_1, other for illustrative purposes
- dynamic wind pressure of $900N/m^2$ from Table 2, Ref. 0_5_1

```
clc;clear;close all

%% problem data

L = 0.35; % [m] Distance entre l'axe et la base de la parabole
L1 = 0.4; % [m] Distance between the slewing ring and the elevation axis
R = 1.86/2; % [m] Rayon de la parabole
h = 0.32; % [m] Profondeur de la parabole
ref_pres = 900; % [N/m^2] Max wind dynamic pressure according to SIA261
```

```
rho = 1.292; % [kg/m^3] Densit air
Cd = 1.38; % [] drag coefficient
v = [20/3.6 ; 50/3.6 ; 120/3.6 ; 160/3.6]; % [m/s] wind speed

%% draws the parabola in 3D space

r = linspace(R/100,R,100);
theta = linspace(0,2*pi,100);

x = r.*cos(theta');
y = r.*sin(theta');
z = (L + r.^2 * h/R^2) .* ones(size(theta'));

x = x(:)'; % turns x,y,z from a matrix (whose lines and collumns correspond to r and theta value) to a vector
y = y(:)';
z = z(:)';

%% calculating wind torque for every angle

alpha = linspace(0,pi/2,50); % alpha is the angle between zenith and dish orientation. alpha = pi/2 - elevation

surface_lever = zeros(size(alpha)); % "surface_lever" is the surface of the parabola exposed to wind, multiplied by the lever arm of wind forces with respect to the elevation axis
SR_surface_lever = zeros(size(alpha)); % surface_lever, applied on slewing ring
exposed_A = zeros(size(alpha)); % Total dish area exposed to wind
lever_arm_elevation = zeros(size(alpha)); % lever arm of wind forces with respect to elevation axis
SR_lever_arm = zeros(size(alpha)); % lever arm of wind forces with respect to slewing ring
for i = 1:numel(alpha)
    [xrot,yrot,zrot] = rot(x,y,z,alpha(i)); % rotates the parabola of an angle alpha along the y (elevation) axis

    % the folowing lines of code will draw the projection of the tilted dish
    % on the y-z plane. This projection is equivalent to the area exposed to
```

```
    % wind
    k = boundary(yrot',zrot',0);
    warning('off')
    parabola_projection = polyshape(yrot(k),zrot(k));
    warning('on')
    % uncomment following line ;for debug purposes
%     plot(parabola_projection)
%     hold on
%     axis('equal')
%     ylim([-1 1.5])
%     title('projection of the parabola dish')
%     pause(0.1)
%     hold off
    A = area(parabola_projection);
    [~,z0] = centroid(parabola_projection);

    surface_lever(i) = A*z0;
    SR_surface_lever(i) = A*(z0+L1);
    exposed_A(i) = A;
    lever_arm_elevation(i) = z0;
    SR_lever_arm(i) = L1 + z0;

end

wind_pres = Cd*[1/2*rho*v.^2;ref_pres];

wind_torque_elevation = wind_pres * surface_lever; % drag force equation, multiplied by the arm_lever
wind_force_normal2elev = wind_pres * exposed_A;
wind_force_alongElev = wind_pres * exposed_A(1);
tilting_torque_elevation = wind_pres * surface_lever(1);
tilting_torque_SR = wind_pres * SR_surface_lever; % !! THIS NEGLECTS THE EFFECT OF THE WIND ON THE MOTOR HOUSING !!

figure()
plot((pi/2-alpha)*180/pi,wind_torque_elevation)
title('Wind torque with respect to elevation angle')
xlabel('Elevation angle [deg]')
ylabel('Wind torque along elevation axis [Nm]')
legend('20kph wind','50kph wind','120kph wind','160kph wind','Reference')

figure()
plot((pi/2-alpha)*180/pi,wind_torque_elevation(1:2,:))
title('Wind torque with respect to elevation angle')
```

```
xlabel('Elevation angle [deg]')
ylabel('Wind torque on elevation axis [Nm]')
legend('20kph wind','50kph wind')

figure()
plot((pi/2-alpha)*180/pi,wind_force_normal2elev)
title('Radial force on elevation axis with respect to elevation angle')
xlabel('Elevation angle [deg]')
ylabel('Wind force [N]')
legend('20kph wind','50kph wind','120kph wind','160kph wind','Reference')

figure()
plot((pi/2-alpha)*180/pi,tilting_torque_SR)
title('Tilting torque on slewing ring with respect of elevation angle')
xlabel('Elevation angle [deg]')
ylabel('Wind torque on SR [Nm]')
legend('20kph wind','50kph wind','120kph wind','160kph wind','Reference')

function [xrot,yrot,zrot] = rot(x,y,z,alpha)

init_vector = [x;y;z];
rot_mat = [[cos(alpha) 0 sin(alpha)];[0 1 0];[-sin(alpha) 0 cos(alpha)]];
rot_vector = rot_mat*init_vector;

xrot = rot_vector(1,:);
yrot = rot_vector(2,:);
zrot = rot_vector(3,:);

end
```

Arduino Uno code

This code was done for an Arduino Uno and we cannot guarantee that this code will respect our speed/resolution requirements nor that it will be reliable. However, only minor tweaks are needed to make this code work on the Teensy 4.0, which will allow us to work reliably. Of course, extensive tests are needed with the constructed telescope.

```
#include <math.h>

```

```c
#define MICROSTEPS 10000.0
#define AltREDUCTION 140.0
#define AzREDUCTION 227.0

#define ENABLEon PORTD |= (1<<2)
#define ENABLEoff PORTD &=~(1<<2)

#define AltDIRon PORTD |= (1<<3)
#define AltDIRoff PORTD &=~(1<<3)
#define AltDIRtoggle PORTD ^= (1<<3)
#define AltDIRisHIGH (PORTD&(1<<3))
#define AltSTEPon PORTD |= (1<<4)
#define AltSTEPoff PORTD &=~(1<<4)
#define AltSTEPisHIGH (PORTD&(1<<4))
#define AltBOOSTon PORTD |= (1<<5)
#define AltBOOSToff PORTD &=~(1<<5)
#define AltFAULT (PORTD&(1<<6))

#define AzDIRon PORTD |= (1<<7)
#define AzDIRoff PORTD &=~(1<<7)
#define AzDIRtoggle PORTD ^= (1<<7)
#define AzDIRisHIGH (PORTD&(1<<7))
#define AzSTEPon PORTB |= (1<<0)
#define AzSTEPoff PORTB &=~(1<<0)
#define AzSTEPisHIGH (PORTB&(1<<0))
#define AzBOOSTon PORTB |= (1<<1)
#define AzBOOSToff PORTB &=~(1<<1)
#define AzFAULT (PORTB&(1<<2))

#define CLOCKon PORTB |= (1<<3)
#define CLOCKoff PORTB &=~(1<<3)
#define AltEncHIGH (PORTB&(1<<4))
#define AzEncHIGH (PORTB&(1<<5))

bool track = 0, gohome = 0;
int i = 0;
uint32_t AltSteps = 0, AzSteps = 0;
uint32_t AltEncPos = 0, AzEncPos = 0;
uint16_t AltEncTurnCnt = 0, AzEncTurnCnt = 0;
uint8_t AltEncCRC = 0, AzEncCRC = 0;
bool AltEncError = 0, AzEncError = 0, AltEncWarning = 0,
    AzEncWarning = 0;
uint32_t AltRef = 0, AzRef = 0, AltDiff = 0, AzDiff = 0;
bool AltDir = 0, AzDir = 0;
double SinAltRef = 0, SinAzRef = 0, CosAzRef = 0;

```

```
double LST = 0, julianDays = 0, julianDays0 = 0; //julianDays variable will be defined by the serial interface

double RightAscention = 0, Declination = 0; //These variables will be defined by the serial interface

uint32_t lastTime = 0, currentTime = 0;

void init_pins(){
  pinMode(2,OUTPUT);          //PD2 //ENABLE
  pinMode(3,OUTPUT);          //PD3 //AltDIR
  pinMode(4,OUTPUT);          //PD4 //AltSTEP
  pinMode(5,OUTPUT);          //PD5 //AltBOOST
  pinMode(6,INPUT_PULLUP);    //PD6 //AltFAULT
  pinMode(7,OUTPUT);          //PD7 //AzDIR
  pinMode(8,OUTPUT);          //PB0 //AzSTEP
  pinMode(9,OUTPUT);          //PB1 //AzBOOST
  pinMode(10,INPUT_PULLUP);   //PB2 //AzFAULT
  pinMode(11,OUTPUT);         //PB3 //CLOCK
  pinMode(12,INPUT_PULLUP);   //PB4 //AltMISO
  pinMode(13,INPUT_PULLUP);   //PB5 //AzMISO
}

void init_timer_steps(){
  TCCR2A = 0 | (1<<WGM21);   // set CTC mode
  TCCR2B = 0 | (1<<CS21);    // set prescaler to 8
  TCNT2  = 0;                // set counter to 0
  OCR2A = 99;                // set compare match register for 20kHz increments // = (16*10^6) / (frequency*prescale) - 1 (must be <256)
  TIMSK2 |= (1 << OCIE2A);   // enable timer compare interrupt
}

ISR(TIMER2_COMPA_vect){ //Interruption for steps. Will execute a step only if needed.
  if(AltSTEPisHIGH) AltSTEPoff;
  else if (AltSteps){AltSTEPon; AltSteps--;}
  if(AzSTEPisHIGH) AzSTEPoff;
  else if (AzSteps){AzSTEPon; AzSteps--;}
  TCNT2  = 0;                // set counter to 0
}

void init_timer_encs(){
  TCCR0A = 0 | (1<<WGM01);             // set CTC mode
```

```
  TCCR0B = 0 | (1<<CS00) | (1<<CS02); // set prescaler to 1024
  TCNT0  = 0;                         // set counter to 0
  OCR0A = 155;                        // set compare match register for 100Hz increments // = (16*10^6) / (frequency*prescale) - 1 (must be <256)
  TIMSK0 |= (1 << OCIE0A);            // enable timer compare interrupt
}

ISR(TIMER0_COMPA_vect){ //Interruption to read the encoders. Protocol described in user manual of the AksIM-2. Will update the AltEncPos and AzEncPos variables.
  AltEncPos = 0, AzEncPos = 0;
  AltEncTurnCnt = 0, AzEncTurnCnt = 0;
  CLOCKoff;
  delayMicroseconds(5);
  for(i=15; i>=0; i--){ //b47-b32
    CLOCKon;
    delayMicroseconds(1);
    CLOCKoff;
    if(AltEncHIGH) AltEncTurnCnt |= (1<<i);
    if(AzEncHIGH) AzEncTurnCnt |= (1<<i);
    delayMicroseconds(1);
  }
  for(i=21; i>=0; i--){ //b31-b10
    CLOCKon;
    delayMicroseconds(1);
    CLOCKoff;
    if(AltEncHIGH) AltEncPos |= (1<<i);
    if(AzEncHIGH) AzEncPos |= (1<<i);
    delayMicroseconds(1);
  }
  CLOCKon;
  delayMicroseconds(1);
  CLOCKoff;
  AltEncError = !AltEncHIGH;
  AzEncError = !AzEncHIGH;
  delayMicroseconds(1);
  CLOCKon;
  delayMicroseconds(1);
  CLOCKoff;
  AltEncWarning = !AltEncHIGH;
  AzEncWarning = !AzEncHIGH;
  delayMicroseconds(1);
```

```
  for(i=7; i>=0; i--){ //b7-b0
    CLOCKon;
    delayMicroseconds(1);
    CLOCKoff;
    if(AltEncHIGH) AltEncCRC |= (1<<i);
    if(AzEncHIGH) AzEncCRC |= (1<<i);
    delayMicroseconds(1);
  }
}

void getLST(){ //Calculates LST according to the formula given by the U. S. N. Observatory.
  julianDays0 = (int)(julianDays + 0.5)-0.5;
  LST = 6.697374558;
  LST += 0.06570982441908 * julianDays0;
  LST += 24.06570982441908 * (julianDays - julianDays0);
  LST += 1.948915002475965E-14 * julianDays * julianDays;
  LST += 0.437598666666667; //add lausanne latitude in hours
  LST = fmod(LST, 24.0);
}

void UpdateReference(){ //Calculates the position of chosen deep sky object. Updates the variables AzRef and AltRef.
  getLST();
  SinAltRef = 0.688112221615796*cos(Declination)*cos(LST*0.261799388 - RightAscention) + 0.725604279523607*sin(Declination);
  AltRef = asin(SinAltRef);
  if (AltRef < 0){
    AltRef = 262144; //point to zenith
    AzRef = 0;
  }
  else{
    SinAzRef = -cos(Declination)*sin(LST*0.261799388 - RightAscention)/cos(AltRef);
    CosAzRef = (sin(Declination)-0.725604279523607*SinAltRef)/(0.688112221615796*cos(AltRef));
    if(SinAzRef >= 0.0) AzRef = acos(CosAzRef);
    else AzRef = -acos(CosAzRef);
    AzRef = fmod(AzRef, 6.283185307179586);
    AltRef = 166886.053607527243599*AltRef; //from radians to 2^20 per rotation
    AzRef = 166886.053607527243599*AzRef; //from radians to 2^20 per rotation
```

```
  }
}

void serialEvent(){ //Reads the serial line to update tracking informations.
  if((char)Serial.read() == '1'){
    track = 1;
    RightAscention = Serial.parseFloat();
    Declination = Serial.parseFloat();
    julianDays = Serial.parseFloat();
  }
  else if((char)Serial.read() == '0'){
    track = 0;
  }
  while (Serial.available()) Serial.read(); //flush rest
}



void setup() {
  init_pins();
  init_timer_steps();
  init_timer_encs();
  sei();
  Serial.begin(9600);
}

void loop() {
  currentTime = millis();
  julianDays += (currentTime - lastTime)/86400000.0; // Keeps the track of time.
  if(AzEncTurnCnt >= 2 || AzEncTurnCnt <= -3) gohome; // Prevent turning too much (cable entaglement)
  if(track && !gohome){
    UpdateReference(); //Update object position
    AltDiff = AltRef - AltEncPos;
    while(AltDiff > (1ul<<19)) AltDiff -= (1ul<<20);
    while(-AltDiff > (1ul<<19)) AltDiff += (1ul<<20);
    AzDiff = AzRef - AzEncPos;
    while(AzDiff > (1ul<<19)) AzDiff -= (1ul<<20);
    while(-AzDiff > (1ul<<19)) AzDiff += (1ul<<20);
    if(AltDiff < 0){
      AltDir = 0;
      AltSteps = -AltDiff/((1ul<<20)/(MICROSTEPS*AltREDUCTION));
    }
```

```
    else if(AltDiff > 0){
      AltDir = 1;
      AltSteps = AltDiff/((1ul<<20)/(MICROSTEPS*AltREDUCTION));
    }
    if(AzDiff < 0){
      AzDir = 0;
      AzSteps = -AzDiff/((1ul<<20)/(MICROSTEPS*AzREDUCTION));
    }
    else if(AzDiff > 0){
      AzDir = 1;
      AzSteps = AzDiff/((1ul<<20)/(MICROSTEPS*AzREDUCTION));
    }
    if(AltRef == 262144){ //If pointing at zenith, go to untangle the cables
      AzSteps += AzEncTurnCnt*(1ul<<20)/((1ul<<20)/(MICROSTEPS*AzREDUCTION));
      if(AzSteps < 0){
        AzDir != AzDir;
        AzSteps = -AzSteps;
      }
    }
  }
  else{ //if not tracking or ordered to go home, go to zenith and untagle the cables
    if(AzEncTurnCnt < 0){
      AzSteps = (-(AzEncTurnCnt+1)*(1ul<<20)+((1ul<<20)-AzEncPos))/((1ul<<20)/(MICROSTEPS*AzREDUCTION)); //point to zenith
      AzDir = 1;
    }
    else{
      AzSteps = -(AzEncPos+AzEncTurnCnt*(1ul<<20))/((1ul<<20)/(MICROSTEPS*AzREDUCTION)); //point to zenith
      AzDir = 0;
    }
    AltSteps = (262144 - AltEncPos)/((1ul<<20)/(MICROSTEPS*AltREDUCTION));
    AltDir = 1;
  }
  if((AzEncTurnCnt == 0 || AzEncTurnCnt == -1) && (AzEncPos < 1000 || AzEncPos > 1047576)) gohome = 0; // clear order to go home if already home
```

```
    if(AltDir ^ AltDIRisHIGH){AltDIRtoggle;
        delayMicroseconds(16);}
    if(AzDir ^ AzDIRisHIGH){AzDIRtoggle; delayMicroseconds
        (16);}
    lastTime = currentTime;
}
```