

UNIVERSITY OF GRONINGEN

BACHELOR'S THESIS ASTRONOMY

Remodeling Dome Control: Applications for the Blaauw Observatory

Author:
Mick VELDHUIS
(s3745805)

Supervisor:
Jake NOEL-STORR, PhD

Second Reader:
PROF. DR. M.A.W. VERHEIJEN

July 11, 2021



rijksuniversiteit
groningen

Abstract

Currently, the dome control software for the Blaauw Observatory assumes that the telescope is located in the center of the dome. This causes the dome to significantly obstruct the telescope, autoguider, and finderscope, particularly in the eastern part of the sky, which is detrimental to the science results that can be obtained with the Gratama telescope. In this thesis, we developed a method to quantify the aperture obstruction by the dome, and using that method, determine the optimal position for the dome given the orientation of the Gratama telescope, such that the telescope is never obstructed by the hemispherical dome. In addition, with the method we developed, the dome does no longer have to move continuously, as long as the telescope tracks an object with fixed right ascension and declination, which is beneficial, because previously, whenever the dome would move, it would interfere with other electronic at the Blaauw Observatory, such as the CCD.

Contents

Abstract	i
1 Introduction	1
2 Modelling the Blaauw Observatory	3
2.1 The Gratama Telescope	3
2.2 The Telescope Enclosure	4
2.3 A 3D CAD Model	5
2.4 Visualising Aperture Obstruction	8
3 Gratama Telescope Pointing	9
3.1 Celestial Coordinates	9
3.2 Coordinate Transformations	12
3.3 Telescope Pointing	13
4 Dome Control Model	17
4.1 Basics of Ray Tracing	17
4.2 Ray-Dome Intersections	19
4.3 Quantifying Aperture Obstruction	20
4.4 Determining the Optimal Dome Position	22
4.5 Results & Discussion	23
5 Dome Control System	26
5.1 The New Method for Dome Control	26
5.2 Control Software	28
6 Conclusions & Improvements	31
7 Acknowledgements	33
References	34
Appendices	35
A Additional Figures	35
B Source Code	41
C Telescope & Dome Measurements	42
D 3D Models	45

1 Introduction

Nearly all optical telescopes are enclosed by a dome. They rotate and track the orientation of the telescope, to make sure that the telescope is pointing through the dome slit, such that the telescope can collect as much light as possible, while simultaneously protecting the telescope from hazards from its environment, such as rain, wind, light pollution, or temperature gradients, which can deter image quality, as explained in more detail by Pérez García (2010).

We concern ourselves with the dome of the Blaauw Observatory. The observatory has been active since 2008 and is located on top of the Bernoulliborg, a building of the Faculty of Science and Engineering of the University of Groningen (RUG). Its purpose is mainly educational, it being used as a tool to prepare students for working with professional grade telescopes and observatories. On top of that, the observatory is used for outreach, to teach the general public about the marvels of the universe and astronomy in general, RUG (2011).

The Blaauw Observatory houses a 40 cm Ritchey-Chrétien optical telescope on a German equatorial mount, the Gratama telescope. The telescope is also equipped with an autoguider, a finderscope, and soon also with a fibre-fed spectrometer. These instruments are enclosed by a hemispherical dome, with a diameter of 6.25 m, manufactured by Ash-Dome, a company based in Tuscon, Arizona.

For the telescope to gather light it should be able to point outside, and not obstructed by the dome. As mentioned, to achieve this, the dome tracks the position and orientation of the telescope. This is managed by the dome control software, which feeds the telescope orientation to a program, written by Noel-Storr (2019), that continuously computes the desired dome position and sends the appropriate instructions to the dome to move it to that position.

However, since the telescope is attached to an equatorial mount, the telescope is pointing off-centre with respect to the dome slit. Since the dome control software currently assumes that the telescope is pointing along an axis aligned with the dome slit. This causes the telescope aperture – as well as that of the autoguider and finderscope – to be obstructed by the dome by a certain amount, depending on the orientation of the telescope. Using the method we will develop in Chapter 4 of the thesis, we created Figure 1, which greatly shows the severity of the obstruction of the telescope aperture by the dome with the current dome control system in the eastern part of the sky. Similar graphs have been produced for the apertures of the autoguider and finderscope, these are listed in Appendix A.

A specific example of a scenario where the dome obstructs the aperture(s) is shown in Figure 2. From the Figure it seems that, in these orientations, the primary aperture is slightly obstructed, the finderscope will not gather any light at all, and the autoguider might be partially obstructed as well.

Thus, decreasing the light gathering power of the telescope, which can result in poorer image quality, and thus in a lower signal-to-noise ratio in the collected

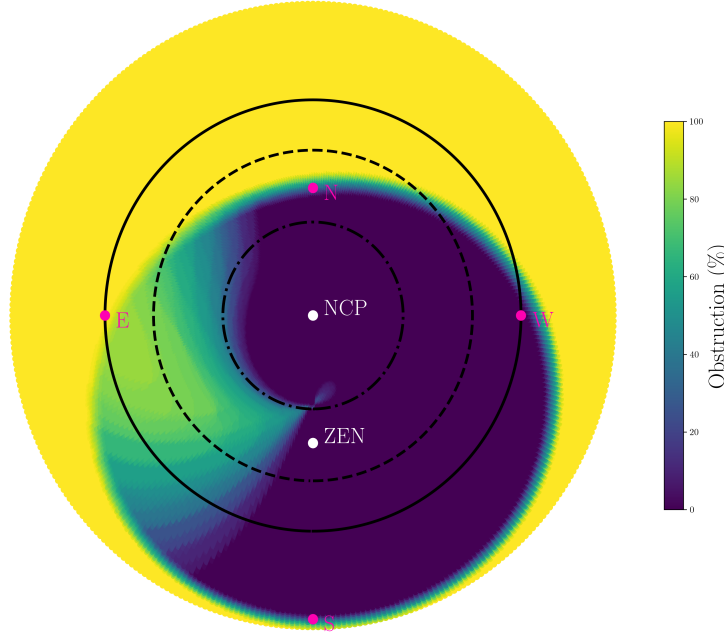


Fig. 1. The percentage of the aperture of the Gracama telescope, modelled using the method we develop in Chapter 4, assuming that the azimuth of the dome is given by Eq. 3, for hour angles from 0 h to 24 h and declinations from -40° to 90° .

data, negatively impacting the quality of the science results that can be obtained with the Gracama telescope.

In this thesis, first we quantify how much of the aperture is obstructed given the position of the dome and orientation of the telescope, then we use that information to devise a new and improved dome control system, for the Blaauw Observatory. With the aim of reducing the obstruction of the telescope aperture by the dome, and thereby improving the overall quality of the data recorded by the Gracama telescope.

In the next chapter (Chapter 2), we discuss the telescope and dome structure, to create a 3D CAD model of the telescope-dome system. After which, in Chapter 3, we shall describe the theory behind coordinate transformations to determine the position of the aperture(s) given the relevant sky-coordinates. In Chapter 4, we use ray tracing techniques to determine where the light entering the telescope intersects the dome. That method we use in Chapter 5 to outline and implement the new dome control scheme. In the final chapters, we will discuss the new control system and comment on its capabilities, further possibilities, and improvements.

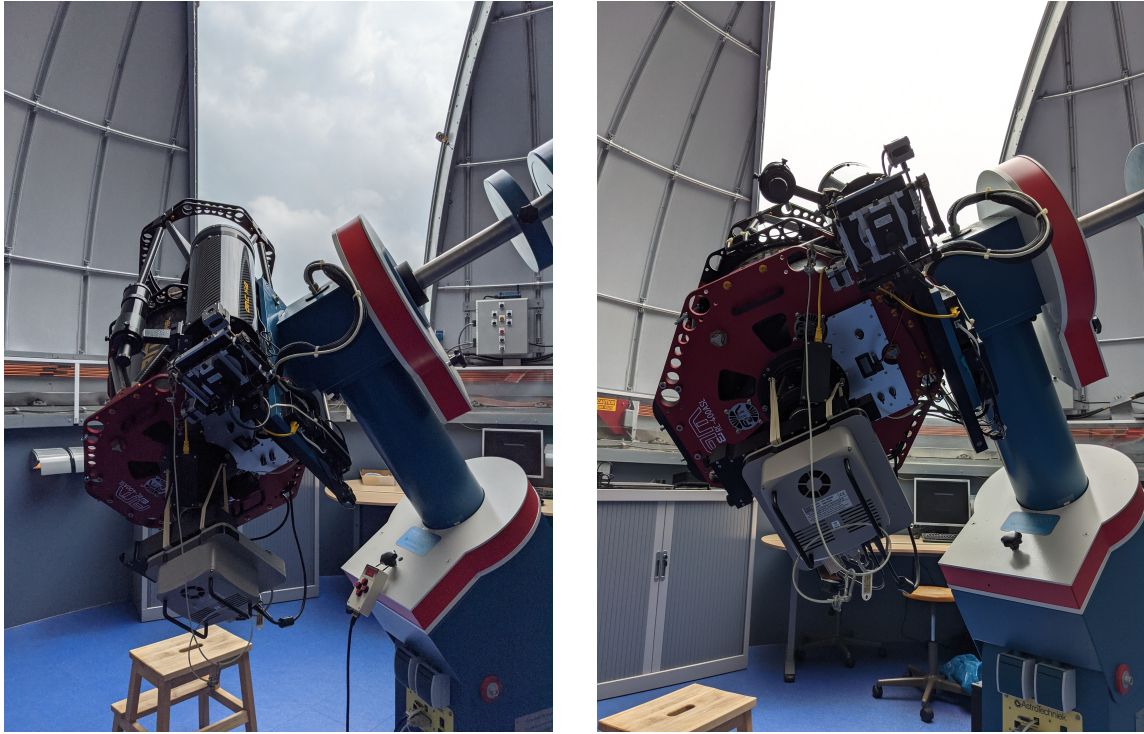


Fig. 2. The Gratama telescope, pointing to an hour angle and declination of $h = -3^{\text{h}}34^{\text{m}}49^{\text{s}}$ and $\delta = 22^{\circ}56'$, and Blaauw Observatory dome, at a azimuth of 105.3° . For this orientation of the telescope the aperture is partially blocked by the dome, and note that the autoguider sits on the edge of the slit and that the finderscope is probably largely blocked by the dome.

2 Modelling the Blaauw Observatory

To examine the problems with the current dome control system of the Blaauw Observatory, we create a 3D CAD model of the telescope and dome. To create the model, we first discuss the telescope and dome setup, emphasising their geometry.

2.1 The Gratama Telescope

The Gratama telescope, as alluded to in the introduction, consists of a Ritchey-Crétien (RC) optical telescope, with a 40 cm primary mirror and a 320 cm focal length, shown in Figure 3. A Ritchey-Crétien telescope is made up of two hyperbolic mirrors, the primary mirror has a hole in it, to allow observing from the back-focus and its encased by a carbon fibre sheet. At the back, a CCD camera (SBIG STL-6303E) is able to record the light entering the optical tube. Note that the secondary mirror is not encased by anything, which is preferable since there could be air building up and vibrating with varying temperatures, which can deter

the image quality.

Attached to the frame of the primary telescope is a 15 cm Maksutov telescope; an OMC 140 from Orion Optics¹, with a 200 cm focal length. This telescope functions as an autoguider, to keep the RC telescope locked at its target, Conrad (2014). Similar to the design of an RC telescope, a Maksutov telescope is obstructed in the center, due to the secondary mirror, which has a diameter of 44 cm.

Additionally, attached to the autoguider, there is a finderscope with a 5 cm diameter, which is generally used at the start of observing, to properly align the telescope with a particular celestial object.

All these instruments are mounted on an equatorial mount produced by AstroTechniek², a company that used to be based in Limburg, the Netherlands.



Fig. 3. The Gratama telescope parked at zenith.

2.2 The Telescope Enclosure

The dome of the Blaauw observatory was built by the American company Ash-Dome³. In figure 4a we see their typical dome design specifications; a hemispherical steel dome. The dome of the Blaauw Observatory follows the same design, and has a diameter of 20 feet and 6 inches, or rather about 6.25 m.

As shown in figure 4a, the dome consist of an upper and lower shutter, the latter which is opened via a hydraulic mechanism, whereas the upper shutters can open or close via a motorized worm drive. Note that the shutters are slightly suspended off the hemispherical dome, by so-called weather seals. Ash-Dome domes use an up-and-over type shutter, which allows the observer to keep the lower section – blocking 30° from the horizon – convenient to counter light pollution or can function as a wind screen.

These shutters, when they are both completely open, allow for 105° of visibility (the arc angle of the observing aperture⁴) in altitude, as shown in figure 4b.

¹The OMC 140 product page: <https://www.orionoptics.co.uk/OMC/omc140maksutovca.html>

²Archived AstroTechniek homepage: <https://web.archive.org/web/20180413032003/astrotechniek.net/>

³Ash Manufacturing Company homepage: <https://ashdome.com>

⁴Ash-Dome FAQ: <https://ashdome.com/faq/>

Furthermore, the dome slit (observing aperture) has a width of 1.8 m as measured with a tape measure. Even though the actual slit width is 1.8 m, its effective width is about 6 cm smaller, due to a lever connecting the dome and the lower shutter on the inside of the slit.

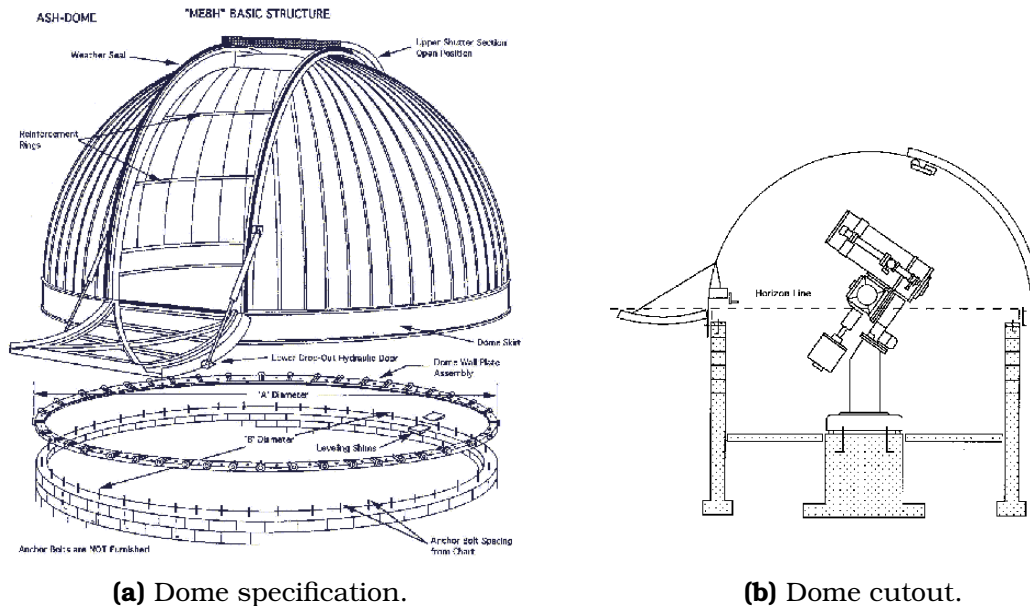


Fig. 4. Drawings of the Ash-Dome hemispherical telescope enclosure with an up-and-over type shutter. Left: a view of specifications of their typical domes, right: a cutout of their domes with an equatorial mount and a telescope.

The dome is supported by ball bearing rollers, locked into the support rails on a cylindrical wall. To move the dome in azimuth, and thus change the orientation of the observing aperture, the dome is equipped with a motorized gear device – shown in figure 5 – allowing the dome to rotate a full 360° . The position of the device in figure 5 marks the reference position for the dome azimuth, which is located at an azimuth angle of 68.2° , which is an important quantity since the dome does not have any means to record its absolute position (in terms of the azimuth angle), but rather relies on the position of the gear device to accurately determine its position. Therefore, the dome needs to be properly calibrated, each time before moving the dome. In a similar fashion, after one has finished observing, the dome needs to be parked a certain distance from the calibration point, such that it is able to find it again during calibration.

2.3 A 3D CAD Model

Using Autodesk Inventor⁵, a CAD application used – among others – for mechanical design and simulations, we create a model of the telescope-dome system. With

⁵Autodesk Inventor product page: <https://www.autodesk.com/products/inventor>

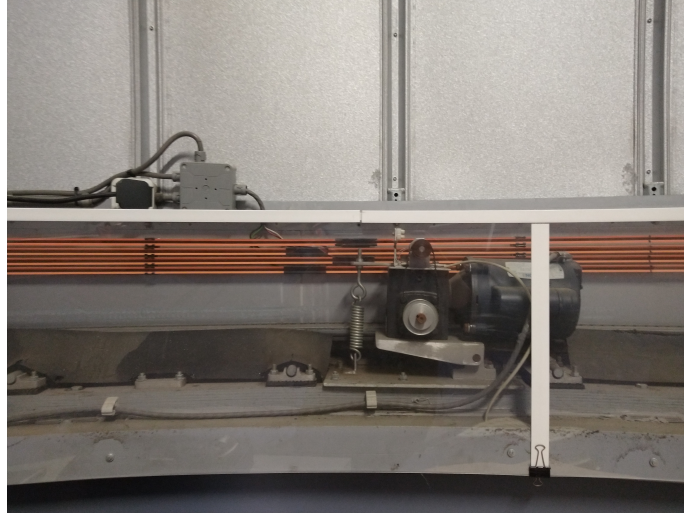


Fig. 5. The motorized gear device, controlling the rotation of the dome.

the goal of visualizing the obstruction issues outlined in the introduction.

Table 1. Geometrical properties of the Gratama telescope.

Joint	Length (cm)
ℓ_1	81.4 cm
ℓ_2	109.8 cm
ℓ_3	43.9 cm
ℓ_4	35.6 cm
ℓ_5	20.0 cm
Aperture	Diameter (cm)
Primary	400
Autoguider	150
Finderscope	50
Aperture obstruction	Diameter (mm)
Primary	175
Autoguider	44

A rendered image of the Gratama telescope model is shown in figure 6. For modelling purposes, we only included the mount, the primary mirror assembly of the Ritchey-Chrétien telescope, the autoguider, and the finderscope. We based the model on tape measurements of the telescope and available product information. These measurements are listed in appendix C. Schematically, the telescope can be described by the measurements in table 1, if we were to imagine the mount, telescope, autoguider, and finderscope, as a simplified system of five joints.

First, the length ℓ_1 , from the ground to the RA axis. Then ℓ_2 , from the RA axis to the Dec axis. From that joint, the distance ℓ_3 extends to the center of the telescope. The last two ℓ_4 and ℓ_5 are the distances from the telescope center to the center of the autoguider and from the center of the autoguider to the center of the finderscope.

Note that the right ascension axis of the telescope is inclined by an angle $90^\circ - \phi$ with respect to the zenith, which we will go into a bit deeper in the next chapter.



Fig. 6. Rendered image of the 3D CAD model of the telescope and equatorial mount, pointing with a zero hour angle and zero declination.

The dome geometry is relatively simple, it being a hemisphere with the properties listed in table 2, and in the previous section. The height of the cylindrical dome walls is 1.66 meters, as evident from plans of the observatory, listed in Appendix C. These properties were used to create a simplified model of the dome, which is shown in figure 7, along with the telescope.

Table 2. Properties of the Blaauw Observatory dome.

Diameter (D)	6.25 m
Slit width (w)	1.74 m to 1.80 m
Dome wall extent (e)	1.66 m
Slit extent (θ_s)	105°
Latitude (ϕ)	53.24°

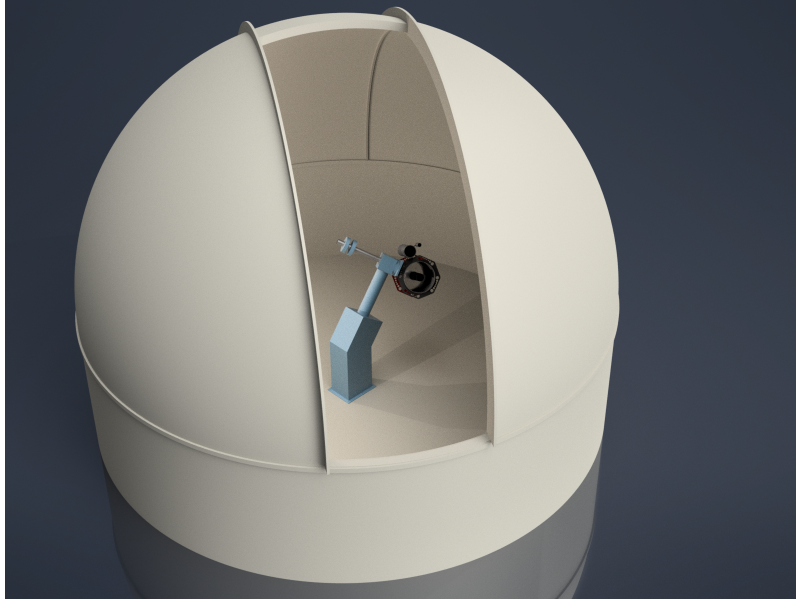


Fig. 7. Rendered image of the 3D CAD model of the observatory, i.e. the telescope and the dome structure.

2.4 Visualising Aperture Obstruction

In figure 2, we show one of the many problem scenarios, where the telescope is being obstructed by the dome. In particular, the current dome control system has trouble when the telescope is pointing in the eastern part of the sky, as evident from Figure 1 and is the case in the aforementioned photos. In those photos the telescope is pointing with an hour angle and declination $h = -3^{\text{h}}34^{\text{m}}49^{\text{s}}$ and $\delta = 22^{\circ}56'$, for which the current dome control system moves the dome to an azimuth of 105.3° .

As is visible in Figure 8 of the observatory model, with the dome and telescope in the same position as figure 2, the dome obstructs the aperture of the telescope and finderscope in a similar manner as Figure 2, and barely allows the autoguider to point out of the dome.

The red cylinders portrait the aperture along its line-of-sight, visualising the obstruction of the telescope, autoguider, and finderscope. As mentioned, the RC telescope and autoguider contain a central obstruction, due to their secondary mirrors, this obstruction is characterised by the hole in the center of those cylinders.

We shall use this model to verify the method to quantify the aperture obstruction, which we will discuss in Chapter 4. The models, with all the relevant Inventor project files, can be found on GitHub, in the [telescope-models](#) repository of the Practical Astronomy Crew (PAC), [Veldhuis & Noel-Storr \(2021a\)](#).

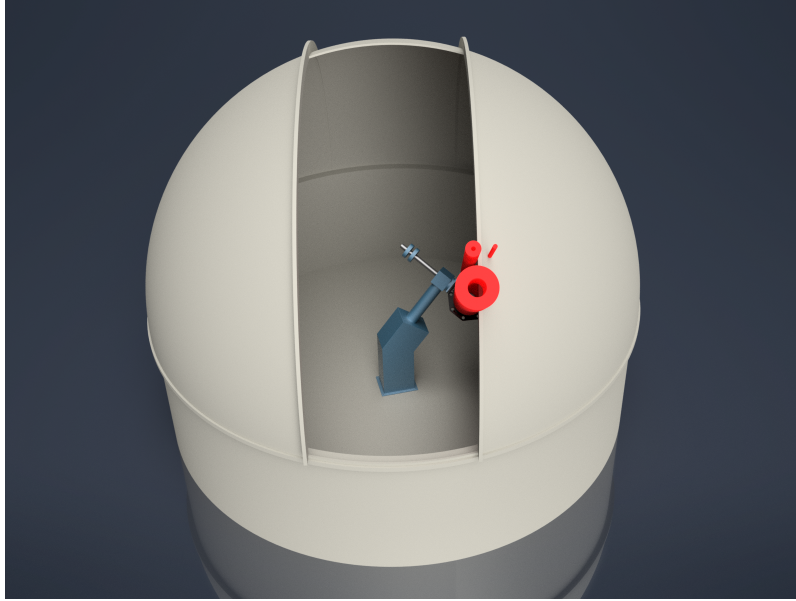


Fig. 8. Rendered image of the 3D CAD model of the observatory, with the dome and telescope oriented as in Figure 2. The red cylinders are a visual guide to show the aperture obstruction.

3 Gratama Telescope Pointing

To understand how much of a particular aperture is blocked, we need to be aware of its position and pointing direction, and particularly how it is influenced by the orientation of the telescope. To this extent, we consider the relevant coordinate systems and, with the help of transformation matrices, we describe the position of the aperture(s) in the dome frame.

3.1 Celestial Coordinates

To describe the pointing of the Gratama telescope, we first consider the relevant coordinate systems. In particular, the equatorial coordinate system, which is useful to describe the movement of the telescope (and mount), and the horizontal coordinate system, which is used to describe the motion of the dome.

In a horizontal coordinate system, the position of a celestial object X is described by the azimuth and altitude with respect to the horizon, marked by the points NESW (north-east-south-west), of the observer at the origin of Figure 9. Note that the azimuth angle A is measured with respect to the celestial north (N); as mentioned previously, it is the angle that describes the position of the dome.

The caveat of this coordinate system is that the objects, described by the altitude and azimuth, constantly change position as the Earth rotates around its axis.

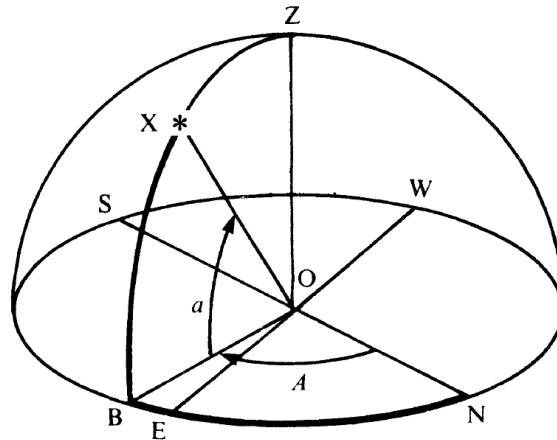


Fig. 9. Horizontal coordinate system. Image from [Duffett-Smith \(1979\)](#).

Therefore, it is more convenient to describe celestial objects in a frame relative to the equator of the Earth; the – aptly named – equatorial coordinate system, as shown in Figure 10. The relevant coordinates in this system are the right ascension α and declination δ , which are determined with respect to the equator, that is inclined with respect to the horizon (still marked NESW) by the angle $90^\circ - \phi$, as mentioned in the previous chapter. The declination, looking at Figure 10, is measured from 0° at R to $\pm 90^\circ$ at the poles, in this figure only the celestial north pole (NCP) is shown, being marked with a P. The right ascension is the angle subtended by the origin O and the vernal equinox Υ , measured eastwards.

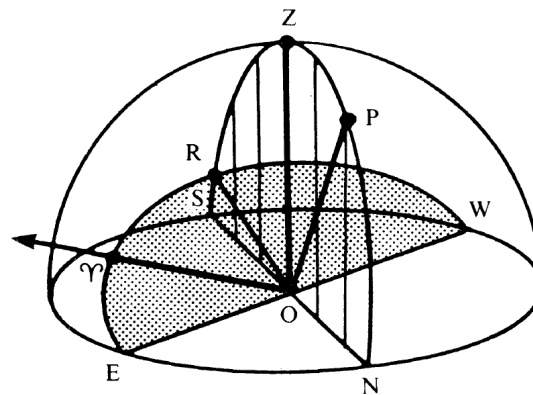


Fig. 10. Equatorial coordinates on the celestial sphere, with the celestial equator inclined by $90^\circ - \phi$, the NCP P, and the horizon circle NESW. Image from [Duffett-Smith \(1979\)](#).

In this coordinate system, the telescope's pointing is described by the hour angle h and declination δ . The hour angle – shown in Figure 11 as H – is measured

with respect to the local meridian (the circle NPZRS), and depends on the right ascension and the local sidereal time (LST),

$$h = \text{LST} - \alpha \quad (1)$$

Contrary to the right ascension, the hour angle is measured westward.

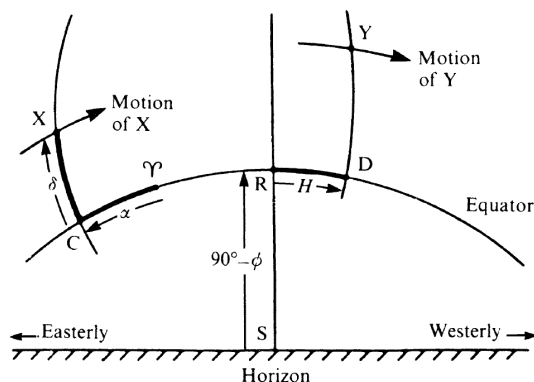


Fig. 11. Equatorial coordinates as seen by an observer in the northern hemisphere, with the hour angle H measured westward. Image from [Duffett-Smith \(1979\)](#).

With the information about the hour angle and declination, we could directly compute the azimuth of the celestial object we are tracking and use that angle to position the dome, which is presumably how the current dome control system functions. In that case, knowing the latitude ϕ of the Blaauw Observatory, the hour angle h and declination δ , we could compute the azimuth by computing the altitude using

$$\sin a = \sin \delta \sin \phi + \cos \delta \cos \phi \cos h, \quad (2)$$

from which we can compute the azimuth with

$$\cos A = \frac{\sin \delta - \sin \phi \sin a}{\cos \phi \cos a}, \quad (3)$$

according to [Duffett-Smith \(1979\)](#), noting that in the western part of the sky – for hour angles from 0 h to 12 h – the true azimuth is actually $360^\circ - A$.

However, the issue is that the telescope, autoguider, and finderscope apertures are not in the center of the dome. Therefore, with Eq. 2 and 3, the dome could possibly block either of those apertures. To account for the off-centre nature of the telescope, particularly visible in figure 6 of the CAD model, we properly consider the geometry of the telescope and its mount in the next two sections.

3.2 Coordinate Transformations

In robotics, to keep track of rigid body motions, we commonly use transformation matrices. These transformation matrices can come in two flavours: active and passive. In this thesis, we will use the active view, due to its simplicity compared to the passive view, even though both are equivalent.

In the active view, we consider a single fixed frame, e.g., the frame of the observatory. The applied transformations then transform the position of the rigid body from the origin of that frame to the desired position, as explained by [Selig \(2006\)](#). This is contrary to the passive view, where the points remain fixed, as opposed to the frames, which are in motion.

Furthermore, We consider a right-handed coordinate system, as the one shown in Figure 12. Where the rotation along the x, y, z -axes, are respectively described by the angles α, β, γ .

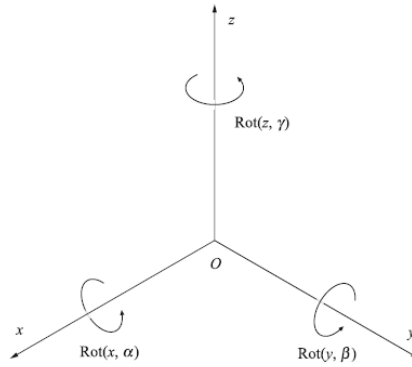


Fig. 12. A right-handed coordinate system. Image from [Mihelj et al. \(2019\)](#).

We can define a translation matrix H as,

$$H = \text{Trans}(x, y, z) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

This matrix can be used, for instance, to translate the point \mathbf{p} ,

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (5)$$

by a distance (x, y, z) along the appropriate axes, where the fourth row is a dummy index.

We can rotate the coordinate system by use of the following rotation matrices $R_x(\alpha)$, $R_y(\beta)$, and $R_z(\gamma)$:

$$R_x = \text{Rot}(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

$$R_y = \text{Rot}(y, \beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (7)$$

$$R_z = \text{Rot}(z, \gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (8)$$

Note that, e.g., R_x assumes that the angle between the x -axes in both the original and rotated frame is zero. In the next sections of this thesis we shall write the trigonometric functions in the rotation matrices as c_α, s_α instead of $\cos \alpha, \sin \alpha$, for brevity.

These translational and rotational transformations are applied from left to right. For instance,

$$H = \text{Trans}(x, y, z) \text{Rot}(x, \alpha),$$

is a transformation in which we first translate a given point by (x, y, z) and then rotate along the x -axis.

We can denote the pose of a certain reference frame with respect to another. Consider the reference frames (x_0, y_0, z_0) and (x_1, y_1, z_1) , the transformation 0H_1 denotes the pose of a vector in frame 1 as a vector in reference frame 0. Consider the application of n consecutive transformations, then a vector in the n th frame can be described in the fixed origin frame, by the transformation 0H_n :

$${}^0H_n = {}^0H_1 {}^1H_2 {}^2H_3 \cdots {}^{n+1}H_n. \quad (9)$$

3.3 Telescope Pointing

The orientation of the telescope aperture in the frame of the observatory can be described as a series of translations and rotations. In robotics, the orientation, is generally called the pose of the robot, as mentioned by Mihelj et al. (2019).

As mentioned, the goal is to describe the (origin of) any of the apertures in the base frame, the frame of the Blaauw Observatory dome. Thus, the initial frame is a frame whose origin is the floor of the dome, where the telescope mount is connected to the building. To translate the frame from the floor of the observatory

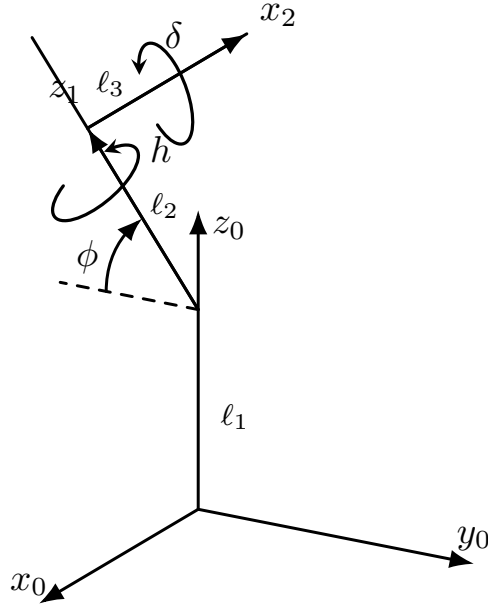


Fig. 13. The pose of the telescope mount at a position, with the rotation direction of the RA and Dec axes. Note that z_0 is pointing to zenith and z_1 is pointing to the NCP.

– along the mount pier – to the origin of the right ascension (RA) axis of the mount, we perform a translation along the z -axis, which is represented by the following transformation matrix,

$${}^0H_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \ell_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (10)$$

where ℓ_1 is the distance from the floor to the origin of the RA axis, as marked in Figure 13, which shows a schematic drawing of the telescope mount, with the lengths from Table 2.

Then, we transform the origin to a frame which intersects the RA axis with the declination axis. This second frame is inclined by an angle $90^\circ - \phi$, such that the RA axis points to the north celestial pole (NCP). Note that ϕ is the geographical latitude of the location of the observatory, which is 53.24° as listed in Table 2. Furthermore, to simulate the rotation of the telescope along the RA axis, we rotate the mount about the z -axis, by the hour angle, h . Lastly, we translate the frame a distance ℓ_2 along the inclined axis, to obtain the position of the intersection of the RA and declination axes.

The resulting pose in the second frame – with respect to the first frame – can thus be written as,

$$\begin{aligned}
{}^1H_2 &= \text{Rot}(x, 90^\circ - \phi) \text{Rot}(z, h) \text{Trans}(0, 0, \ell_2) \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & s_\phi & -c_\phi & 0 \\ 0 & c_\phi & s_\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_h & -s_h & 0 & 0 \\ s_h & c_h & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \ell_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11)
\end{aligned}$$

The subsequent, third, frame – signifying the declination axis – its rotation is controlled by the declination and it is displaced a length ℓ_3 to the left, to the center of the optical axis of the primary aperture. The transformation from the second to third frame results in the following pose

$${}^2H_3 = \text{Rot}(x, \delta) \text{Trans}(-\ell_3, 0, 0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\delta & -s_\delta & 0 \\ 0 & s_\delta & c_\delta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\ell_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (12)$$

Therefore, the pose of the primary aperture in the frame of the observatory dome can be obtained via the following transformation matrix,

$${}^0H_3 = {}^0H_1 {}^1H_2 {}^2H_3 = \begin{bmatrix} c_h & -s_h c_\delta & s_h s_\delta & \ell_3 c_h \\ c_\phi s_h & c_\phi c_h c_\delta - s_\phi s_\delta & -c_\phi c_h s_\delta - s_\phi c_\delta & \ell_3 c_\phi s_h - \ell_2 s_\phi \\ s_\phi s_h & s_\phi c_h c_\delta + c_\phi s_\delta & -s_\phi c_h s_\delta + c_\phi c_\delta & \ell_3 s_\phi s_h + \ell_2 c_\phi \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (13)$$

Now we have the position of the primary aperture as a function of the hour angle h , the declination δ , the latitude of the observatory, and the various distances between the considered frames, as listed in Table 2. However, to determine the optimal dome position, we also need to take into account the apertures of the autoguider and finderscope.

With respect to the frame of the primary aperture, we assume that the frame of the aperture of the autoguider is displaced by a distance ℓ_4 along an axis rotated 45° with respect to the y -axis of the frame of the primary aperture, as is visible in Figure 6. Therefore, the transformation from the RC frame to the autoguider is given by the following transformation matrix,

$${}^3H_4 = \text{Rot}(y, 45^\circ) \text{Trans}(0, 0, \ell_4) = \begin{bmatrix} \frac{1}{2}\sqrt{2} & 0 & \frac{1}{2}\sqrt{2} & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{2}\sqrt{2} & 0 & \frac{1}{2}\sqrt{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \ell_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (14)$$

Finally, the frame of the finderscope with respect to the guider scope can be

obtained by a simple translation along the x -axis,

$${}^4H_5 = \text{Trans}(-\ell_5, 0, 0) = \begin{bmatrix} 1 & 0 & 0 & -\ell_5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (15)$$

since the finderscope is located at an angle of 90° from the axis connecting the autoguider and RC telescope.

Therefore, to get from the autoguider aperture to a point in the dome frame, we require the transformation matrix

$${}^0H_4 = {}^0H_1 {}^1H_2 {}^2H_3 {}^3H_4, \quad (16)$$

and, for the finderscope we need

$${}^0H_5 = {}^0H_1 {}^1H_2 {}^2H_3 {}^3H_4 {}^4H_5. \quad (17)$$

Thus, to convert any arbitrary point 5 in the relevant apertures to the frame of the dome, we compute the matrix product of any of the following matrices 0H_n – for $n = 3, 4, 5$ – with said point.

4 Dome Control Model

We aim to quantify the extent to which the aperture is obstructed by the dome. To achieve that goal, we use ray tracing, or rather the mathematical basis that is commonly used in 3D game engines.

First, we introduce the mathematical background of ray tracing, after which we incorporate that into a method to quantify how much the aperture is obstructed by the dome given the orientation of the telescope characterised by the hour angle and declination and the rotation of the dome characterised by its azimuth.

4.1 Basics of Ray Tracing

First we consider some basics of ray tracing, focusing on the intersection of rays with spheres and capsules. Here we use the books by [Eberly \(2006\)](#) and [Shirley & Morley \(2008\)](#).

Rays, Spheres, and Capsules

We are interested in describing the intersections of rays extruded from the telescope aperture and the dome. Rays are line segments and thus are described by an origin \mathbf{o} and a (unit) direction vector \mathbf{d} ,

$$\mathbf{p}(t) = \mathbf{o} + t\mathbf{d}, \quad (18)$$

where $t \in [0, \infty)$. The dome, or rather its hemispherical cap, could be described as a sphere, parameterised by

$$|\mathbf{p} - \mathbf{c}|^2 = r^2, \quad (19)$$

where r would be the radius of the sphere, centered at \mathbf{c} , and \mathbf{p} is any point on the sphere. A possibly more accurate description of the Blaauw Observatory dome would be that of (the upper half) of a capsule. A capsule consists of a cylindrical side of height $2e$ (where e is the extent) and two hemispherical caps of radius r . Thus, in the capsule frame, the capsule can be described as,

$$\begin{cases} x^2 + y^2 = r^2 & -e \leq z \leq e \\ x^2 + y^2 + (z - e)^2 = r^2 & z > e \\ x^2 + y^2 + (z + e)^2 = r^2 & z < e \end{cases}. \quad (20)$$

Note that the coordinates are in the capsule frame, thus to describe the origin of the ray in the capsule, we should write,

$$\mathbf{o} = \mathbf{c} + x_0\mathbf{u} + y_0\mathbf{v} + z_0\mathbf{w}, \quad (21)$$

where \mathbf{c} is the center of the considered frame.

Ray-Sphere Intersections

The simplest case is the intersection of the ray with a sphere, which could represent the hemispherical cap of the dome. The intersection between a ray and sphere is given by substituting Eq. 18 into Eq. 19 and solving the resulting quadratic equation in t , $|\mathbf{o} + t\mathbf{d} - \mathbf{c}|^2 = r^2$, which can be written as,

$$t^2 + 2a_1t + a_0 = t^2 + 2t\mathbf{d} \cdot (\mathbf{o} - \mathbf{c}) + |\mathbf{o} - \mathbf{c}|^2 - r^2 = 0, \quad (22)$$

Therefore, we find that the line and sphere intersect for t -values satisfying,

$$t = -a_1 \pm \sqrt{a_1^2 - a_0}. \quad (23)$$

We define $\Delta \equiv a_1^2 - a_0$, such that only when $\Delta \geq 0$ there are intersections. Note that in the case of rays extruded from inside the sphere (as is always the case inside the dome), only the largest value for $t \geq 0$ should be considered, as the smaller value will point in the opposite direction of the aperture. Since we will always consider the origin of the ray to be in the sphere, we can test whether the ray intersects the sphere by verifying that $a_0 > 0$.

Ray-Capsule Intersections

Now a more physical representation of the dome. We consider the intersection of a ray (Eq. 18) with a capsule, centered at the point \mathbf{c} , extruded in a unit-length direction \mathbf{w} (see Eq. 21), with extent e , and radius r .

Consider the point \mathbf{o} in the reference frame of the capsule, such that it can be described by the set (x_0, y_0, z_0) (Eq. 21). And similarly for the direction vector, with (x_d, y_d, z_d) . Initially, we can consider two cases: (1) when the ray is parallel to the capsule axis \mathbf{w} , with $|z_d| = 1$, and (2) when its not, when $|z_d| \neq 1$.

First, consider case 1, where the ray intersects an infinite solid cylinder with $x^2 + y^2 \leq r^2$, which occurs when the origin (x_0, y_0, z_0) lies inside the radius of that cylinder. The intersection with the top hemisphere occurs at $(x_0, y_0, z_{\text{top}})$ for $x_0^2 + y_0^2 + (z_{\text{top}} - e)^2 = r^2$. The intersection point along the z -axis is given by,

$$z_{\text{top}} = e + \sqrt{r^2 - x_p^2 - y_p^2}, \quad (24)$$

where we should that $z_{\text{top}} \leq e$. When the ray intersects with the bottom hemisphere, the point where the intersection occurs is simply the mirror image along the vertical axis of the top intersection: $(x_0, y_0, -z_{\text{top}})$.

The t -values we find for respectively the upper and lower hemisphere intersection are given by $t_{\text{top}} = z_{\text{top}} - z_p$ and $t_{\text{bottom}} = z_{\text{bottom}} - z_p$ when $z_d = 1$. When the rays are directed downwards, meaning the direction vector has a negative z -component (i.e. $z_d = -1$), the t -values are negated to obtain the appropriate results.

In the second scenario, case 2, we can again assume that the ray intersects the capsule with an infinite cylinder for $x^2 + y^2 = r^2$, where $x = x_p + tx_d$ and

$y = y_p + tx_d$. However, now the ray does not necessarily intersect with either hemisphere. Solving the resulting quadratic equation, we obtain,

$$a_2 t^2 + 2a_1 t + a_0 = (x_d^2 + y_d^2)t^2 = 2(x_0 x_d + y_0 y_d)t + (x_0^2 + y_0^2 - r^2) = 0. \quad (25)$$

In this case a_2 is always positive and nonzero, otherwise the line would be parallel to the capsule axis. By solving the above equation, we find the following t -values,

$$t = \frac{-a_1 \pm \sqrt{a_1^2 - a_0 a_2}}{a_2}. \quad (26)$$

Thus, defining $\Delta \equiv a_1^2 - a_0 a_2$, we find intersections when $\Delta \geq 0$. The ray intersects the cylindrical wall of the capsule when $|z| = |z_p + tz_d| \leq e$, otherwise it intersects either of the two hemispheres.

The top hemisphere is the only relevant hemisphere for the problem we discuss in the next section. For a more complete treatment, the reader should refer to chapter 15.5.1 of [Eberly \(2006\)](#). The top hemisphere is defined as,

$$x^2 + y^2 + (z + e)^2 = r^2, \quad z \geq e. \quad (27)$$

The solution to this equation takes a similar form as Eq. 23, given that this is just a sphere displaced from the centre of the frame. Thus,

$$t = -a_1 \pm \sqrt{a_1^2 - a_0}, \quad (28)$$

but now with $a_0 = x_0^2 + y_0^2 + (z_0 - e)^2 - r^2$ and $a_1 = x_0 x_d + y_0 y_d + (z_0 - e)z_d$.

We can now use these equations in the next section(s) to determine where the rays extruded from the telescope aperture(s) are intersecting the hemispherical dome of the Blaauw Observatory.

4.2 Ray-Dome Intersections

Simplifying the observatory, we can consider the dome as – the upper half of – a capsule. With the cylindrical walls of the observatory having an extent e and a hemispherical cap of radius r . Thus, to find the intersection of the rays extruded from the aperture, we follow the method outlined in the previous section.

First, consider the origin of the rays \mathbf{o} , which is the origin of the telescope aperture, which is obtained by applying the transformation given by the equations from the previous chapter, on a zero vector (Eq. 5, with $x = y = z = 0$), i.e. $\mathbf{o} = {}^0\text{H}_3 \mathbf{o}_{\text{dome}}$. Note that \mathbf{o}_{dome} represents the origin of the observatory, which we assume is the dome floor. The only rays we are interested in, i.e., those that intersect with the dome, always have a positive z -component for the direction vector. Therefore, any $d_z \geq 0$ we can directly discount.

In case the direction vector \mathbf{d} is nearly parallel with the vertical axis, with $d_z \sim 1$, we can readily determine the intersection, noting that,

$$t = e - o_z + \sqrt{r^2 - o_x^2 - o_y^2}, \quad (29)$$

we find the intersection $(o_x, o_y, o_z + td_z)$, with the hemispherical dome.

For $0 < d_z < 1$, we can imagine that the ray intersects with an infinitely long cylinder, whose intersection is characterised by,

$$t = \frac{-a_1 + \sqrt{a_1^2 - a_0 a_2}}{a_2}, \quad (30)$$

where $a_0 = o_x^2 + o_y^2 - r^2$, $a_1 = o_x d_x + o_y d_y$, and $a_2 = d_x^2 + d_y^2$. In case the aperture is pointing above the walls of the dome, the intersection point $\mathbf{p}(t)$ its z -components is greater than the wall extent, i.e. $p_z > e$. Meaning the rays ejected from the aperture will intersect with the dome, rather than the wall. This intersection point is given by,

$$t = a_1 + \sqrt{a_1^2 - a_0}, \text{ where } \begin{cases} a_0 = o_x^2 + o_y^2 + (o_z - e)^2 - r^2 \\ a_1 = o_x d_x + o_y d_y + (o_z - e) d_z \end{cases}. \quad (31)$$

In the end, the intersection is given by $\mathbf{p}(t) = \mathbf{o} + t\mathbf{d}$.

4.3 Quantifying Aperture Obstruction

In the previous section, we described where a ray, emanating from the aperture, intersects the hemispherical dome of the Blaauw Observatory. We now describe a method to figure out whether that ray is blocked, given the orientation of the telescope and dome.

The apertures of the primary telescope, autoguider, and finderscope can be described as circles, as shown in in figure 14, which change orientation depending on the right ascension and declination of the telescope. We assume that the light rays, collected by the aperture, form a cylindrical beam, representing the aperture along the optical axis of the telescope, can be approximated as a collection of rays emanating from the aperture.

Generally, for all three apertures, these rays point along the (unit length) direction,

$$\mathbf{d} = {}^0H_n [\text{Trans}(0, 1, 0) - \mathbf{I}_{4 \times 4}] \mathbf{o}_0 \quad (32)$$

where $n = 3, 4, 5$, for respectively the primary aperture (Eq. 13), autoguider (Eq. 16), and finderscope (Eq. 17), and $\mathbf{I}_{4 \times 4}$ is a 4 by 4 identity matrix. From this circular aperture, we sample a large number, N (preferably above 100), of equidistantly spaced points \mathbf{o}_i , for $i = 1, 2, 3, \dots, N$ and test whether they intersect with the dome and compare their position with the position of the dome slit.

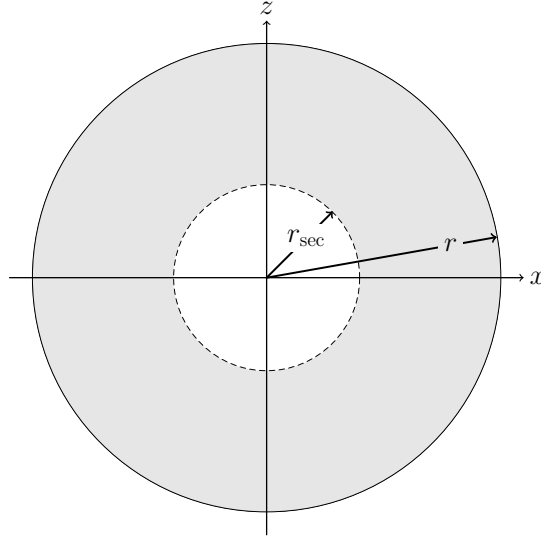


Fig. 14. View of the circular aperture or radius r , with a secondary obstruction with radius r_{sec} , listed in Table 1 for the primary aperture and autoguider, the finderscope has $r_{\text{sec}} = 0$.

Those rays intersect the hemispherical part of the dome at the points $\mathbf{p}_i(t)$, where t is determined by either Eq. 29 or 31. We can use the x and y coordinates of those intersections to figure out whether the rays are passing through the slit, or not. Consider a top-down view of the dome, as shown in Figure 15, the rays pass through the slit when the points (p_x, p_y) lie within the rectangle of width w (the slit width) and height $R + r$, the radius R of the dome plus the length of the part of the slit that extends past zenith r , which is given by $R \sin(\theta_s - \pi/2) \approx 0.8$ m.

Therefore, rays pass through the slit when

$$-\frac{w}{2} < p_x(t) < \frac{w}{2} \quad \text{and} \quad -r < p_y(t) < R. \quad (33)$$

However, note that Eq. 33 is only valid when the dome is positioned at an azimuth of 0° . In general we need to rotate the frame of the intersections by A_d along the z -axis, to correct for this issue. Therefore, before applying the conditions, we rotate the points $\mathbf{p}_i(t)$ by an angular distance A_d in the clock-wise direction with Eq. 8.

Thus, briefly restating the above, to quantify the obstruction of an aperture by the dome, we sample a large number of rays, compute their intersection with a hemisphere the same size as the dome and verify whether they are obstructed or not using the conditions from Eq. 33. The obstruction is then the ratio of rays **not** meeting the conditions with respect to the number of sampled rays.

We implemented this method as a command line utility called **MOCCA**, which can be found on GitHub, and relies on the obstruction package, from **Veldhuis**

& Noel-Storr (2021a), that implements the discussed method; see Appendix B for more details. The application allows one to compute and visualise the percentage obstruction of the aperture by the dome, for hour angles ranging from 0 h to 24 h, declination ranging from 0° to 360° , and dome azimuth angles ranging from 0° to 360° .

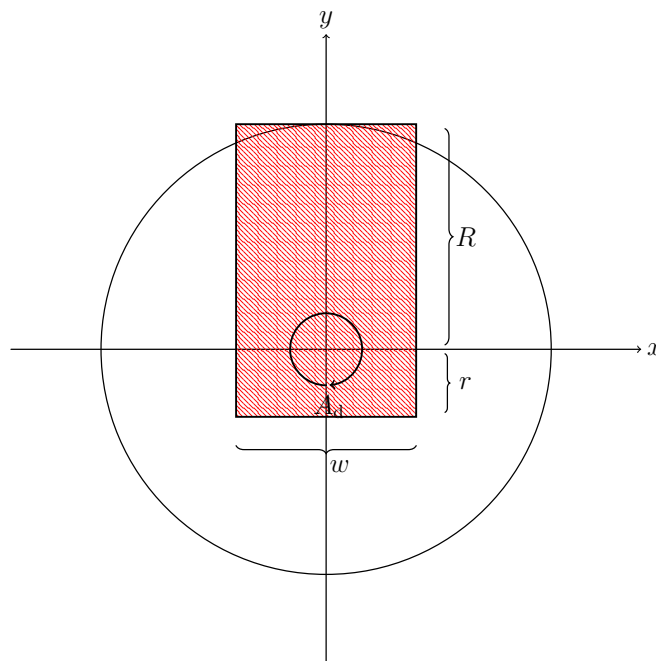


Fig. 15. Top-down view of the dome with the slit dimensions marked.

4.4 Determining the Optimal Dome Position

With the method outlined in the previous section, we can determine, for a certain orientation (h, δ) of the telescope, how severe the obstruction of the telescope by the dome is. In general we are interested in 100% visibility, which implies that for a certain hour angle and declination, there are dome positions A_d that are off-limits. Simultaneously, there will – depending on the (h, δ) pair – multiple azimuths A_d which ensure that the aperture is 100% pointing out of the dome, which could all potentially be the optimal dome position. Therefore, we need a way to quantify the optimal azimuth; we list our definition below.

The optimal dome azimuth

The optimal dome azimuth is the azimuth A_d , for which the hour angle h can extend for the longest amount of time, at a constant declination δ . Such that the dome has to move as little as possible whilst the telescope is tracking an object with fixed (α, δ) on the sky.

The reasoning behind this definition is as follows: generally, we track objects whose right ascension and declination remain constant over time, therefore, the hour angle is the only quantity that changes. The hour angle, as explained in Chapter 3, affects the orientation of the telescope, thus, at some point as the telescope rotates about the right ascension axis, the dome will obstruct the aperture(s) again. Thus, we want the telescope to be able to traverse the longest distance in terms of the hour angle, while the dome remains at the same azimuth.

Additionally, noting that celestial objects move to the west as the Earth rotates along its axis eastward, the longest extent in hour angle, is the farthest distance the hour angle can increase in the eastward direction, i.e. the most positive hour angle. Furthermore, since observations at the Blaauw Observatory never last longer than 24, we can safely ignore the cyclic nature of the hour angle.

Thus, if we assume that the current hour angle is h_0 , we define $\Delta h = h - h_0$, where h is any other hour angle. Presumably, the optimal dome azimuth would be the value A_d that maximizes Δh , under the assumption that we have a relationship between the hour angle and the dome azimuth, $h = h(A_d)$. We use this reasoning to implement a method for calculating the optimal dome azimuth consisting of a set of two Python scripts, which are listed in Appendix B.

As mentioned, with the method from the previous section, we have a grid of (h, δ) pairs, and their associated obstruction ratio. From that grid we select only the coordinate pairs corresponding to 0% obstruction, most – if not all – of those points, as mentioned, correspond to multiple azimuths A_d , as shown in figure 16. Note that this figure focuses on the primary aperture of the RC telescope, similar graphs for the autoguider and finderscope are listed in appendix A, which show a similar trend.

4.5 Results & Discussion

With this method, we obtain the optimal dome azimuth angles, as shown in Figure 17 for the primary aperture. This figure projects the (h, δ) coordinates onto a circle, with along the radial axis $90 - \delta$, ergo, the NCP is located at the center, and the hour angle increases in the usual, westward direction, as explained in Chapter 3. Similar to Figure 9, the NESW mark the horizon, the outer circle marks the celestial equator, the middle circle marks an artificial horizon, and the innermost circle marks the circumpolar region. The artificial horizon marks the region where the telescope points approximately 15° above the horizon, which is favourable to deter any stresses on the primary mirror of the Gracata telescope.

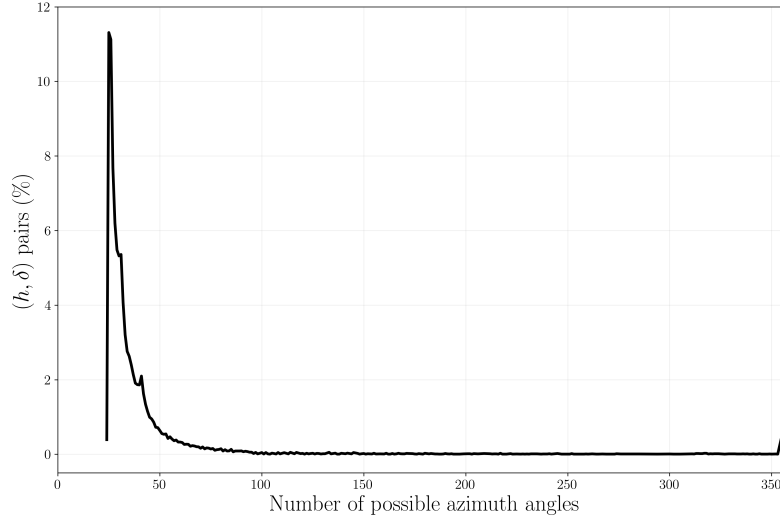


Fig. 16. The percentage of (h, δ) pairs, as a function of the number of possible dome positions, where the telescope aperture is always completely unobstructed by the dome. Assuming a 1° grid spacing in the hour angle, declination, and dome azimuth.

The plot only contains hour angles and declinations where the telescope is unobstructed, therefore, since the telescope is mounted off-centre in the dome frame, the graph is slightly lopsided. Additionally, the NCP is encased by a region of near constant azimuth, where $A_d \sim 0^\circ$.

In the western part of the sky, we observe a rather peculiar set of solutions, where the optimal azimuth seems to radically differ between values around 40° , on the lower end, and about 100° , on the higher end. These are presumably computational artifacts surrounding a region with setting objects. There is a similarly distinctive feature between the two innermost dotted and dashed circles in the south-east quadrant, which may be related to the fact that the dome slit is rectangular, rather than, e.g., circular. That might also explain the region surrounding the zenith.

The optimal dome azimuth when tracking the autoguider, finderscope, or the telescope and autoguider simultaneously are listed in appendix A for completeness, but omitted here for brevity.

These graphs all look very similar on the surface, however, in the case of the finderscope shown in Figure 25, the optimal dome azimuth seems vastly different, because in case we assumed a different criterion for selecting the optimal dome position. For the optimal dome azimuth when tracking the finderscope, we assumed that the finderscope should always point in the center of the dome slit, because we would like to have enough room to the left and right of the finderscope its aperture, to be able to properly align the telescope.

In Figure 18, we see the duration for which the dome can remain at its optimal position. The figure very clearly shows the region in the west where objects will be

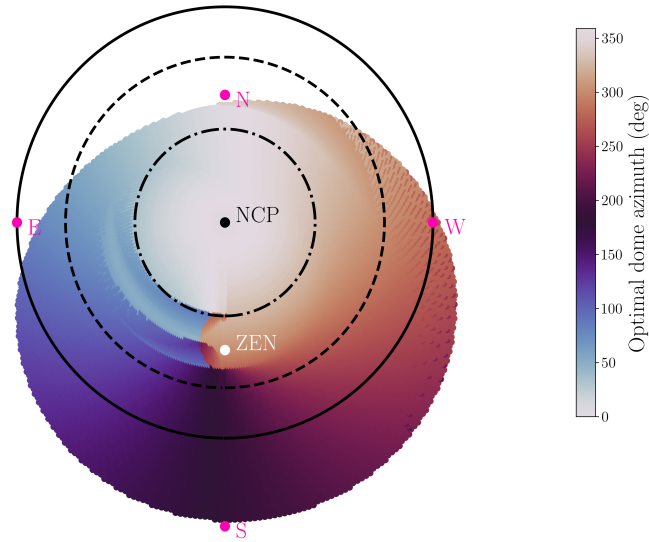


Fig. 17. Optimal dome azimuth for the RC telescope. The outer circle is the celestial equator, the innermost the circumpolar region, and in between there is the artificial horizon, in accordance with the 15° pointing requirement.

setting, where the dome will soon be obstructed by the dome, since the telescope is almost sitting at the edge of the walls of the dome. On the contrary, in case one is observing objects close to the NCP, the dome does not have to rotate for a very long time.

Note, however, that in comparison to Figures 27 and 29 listed in the appendix, this maximum region around the NCP is smaller for the telescope than the autoguider and and telescope + autoguider combination. Presumably, because the aperture of the autoguider is much smaller than the 40 cm diameter of the RC telescope.

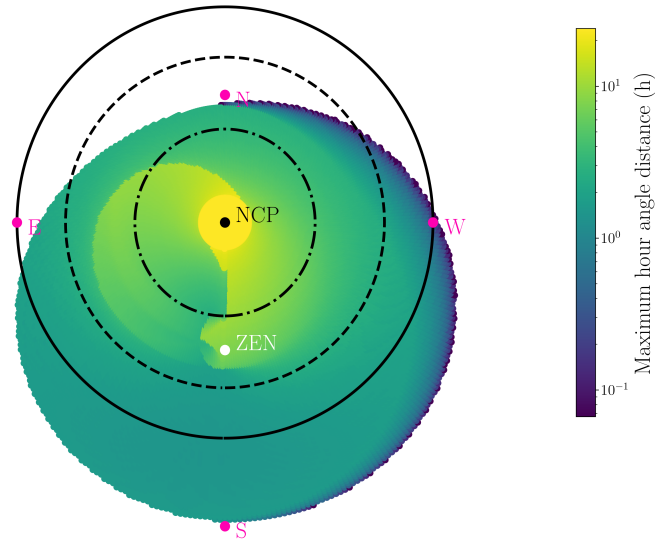


Fig. 18. The duration (HA distance) that the dome could potentially remain at its optimal position, assuming the telescope is tracking an object with a constant right ascension and declination.

5 Dome Control System

5.1 The New Method for Dome Control

In Chapter 4, we determined the optimal dome azimuth on an (h, δ) grid, corresponding to declinations ranging from -34° to 90° and hour angles from 0 h to 24 h. Thus, knowing what aperture(s) to track and the current orientation of the telescope, we are able to determine where to position the dome.

Requirements

As stated many times throughout the thesis, the ultimate goal of the upgrade of the dome control (DCS) system is to position the dome such that the telescope is always pointing through the dome slit. Nevertheless, we have a set of additional requirements for the DCS, in no particular order:

1. Generally, when the autoguider is in use, its together with the RC telescope, and thus they should simultaneously be able to point outside. In this scenario, we could use a slightly more relaxed condition for the autoguider, where the obstruction of the aperture could potentially be as bad as 50% covered by the dome, whilst the telescope should always be completely unobstructed.

2. The finderscope never has to point through the slit at the same time as the autoguider or the RC telescope, as its only ever used to align the telescope before observations.
3. Whenever the telescope is used for direct observation, e.g., with an eyepiece, we can ignore the autoguider.
4. Generally, the telescope tracks an object with constant declination, but, since the Earth rotates, varying hour angle. Thus, as long as the declination remains approximately constant, and the object is not moving out of view of the dome slit, we can keep the dome position fixed until either of those conditions change. This should significantly decrease the movement frequency of the dome, which currently continuously responds to any, however small, changes in the telescope orientation, which causes problems with the electronics at the Blaauw Observatory. For instance, when the CCD is reading out or exposed and the dome moves, the CCD crashes.
5. Thus, following from the previous point, the dome should not move whenever the CCD is exposed or reading out, or the telescope is not currently slewing to a target.

The Control Method

With the requirements from the previous section, we can develop a method for optimal dome control, using the generated *optimal azimuth grids*, containing the hour angle and declination of the telescope, the dome azimuth angle, and the time that the dome can remain at a fixed position at a near constant declination. We can outline the control method for tracking the dome in the following three steps:

Initiating the tracking mode: Whenever the tracking mode is initiated, the software will attempt to obtain the pointing of the telescope – its hour angle h and declination δ – to look up the optimal dome position in the $(h, \delta, A_d, \Delta h)$ grid corresponding to the aperture the dome should be tracking, and move to that position as long as requirement 5 is being met.

Checking the dome & telescope status: Every 5 seconds, the dome control system should verify that, in line with requirement 4, the declination is not varying significantly and that the time, since the dome moved to its current position, has not exceeded the limit Δh , imposed by the limited size of the dome slit.

Feedback: In case it has exceeded those limits, the dome should look up a new value in the appropriate grid and move to that position, when requirement 5 is being met. Otherwise, the dome can remain at its position.

Note that, since the coordinates h , δ , and A_d have a 1° spacing, the current orientation of the telescope will need to be rounded to the nearest degree, before looking up a new position in the grid.

5.2 Control Software

With the developed algorithm, we have created a new version of the dome control system (DCS). As with the previous version, the software is written in Python and uses the Qt framework for the graphical user interface (GUI).

Interfacing With the Dome and Telescope

The dome control method requires information about the pointing of the telescope, the status of the CCD, and the position of the dome.

At the Blauw Observatory, the telescope control is managed via TheSkyX, a program developed by Software Bisque⁶, a company based in the United States. TheSkyX, among its many other features, connects to the telescope mount and allows one to select objects for the telescope (mount) to slew to. TheSkyX is able to connect to any mount that uses ASCOM-compatible drivers. ASCOM, which stands for the Astronomy Common Object Model, is a standard for developing astronomy applications, such as observatory control software, initially developed by Denny (2002). ASCOM is a driver-client layer between equipment and software, which is based on the Microsoft Windows Component Object Model (COM), such that the data is accessible by any programming language that has a library able to deal with these data objects. In such a manner, any piece of equipment of a similar type, i.e., a telescope mount, can, for instance, communicate with TheSkyX.

TheSkyX in turn provides a scriptable interface, which we can access to obtain the orientation of the telescope mount, stored in a COM object created by TheSkyX. With this interface, we are able to, for instance, get the right ascension and declination of the mount, which we can use to determine the optimal position of the dome.

However, the dome does not have such an ASCOM driver, but it can be controlled via a serial interface. The software that interacts with the dome via that interface is called KoepelX, written by Noel-Storr (2019). This software keeps track of the position of the dome and is able to respond to a number of commands to control the dome. Among these commands, we can retrieve the current position of the dome, whether its moving or not, move it to a desired position, calibrate the position of the dome as mentioned in Chapter 2, or stop the dome from moving. Previously, KoepelX was also in charge of keeping the dome slit aligned with the telescope.

⁶Software Bisque homepage: <https://www.bisque.com/>

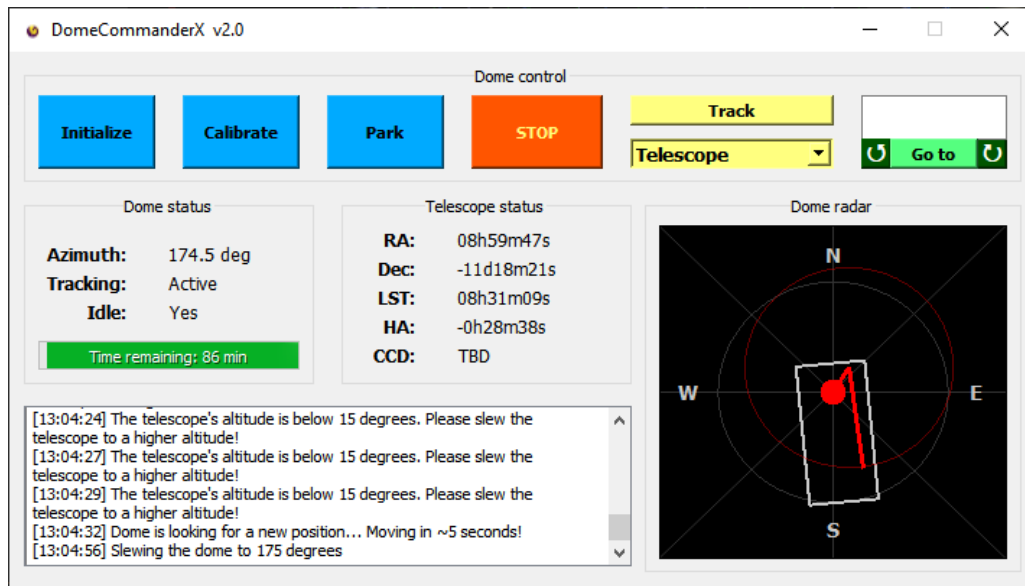


Fig. 19. The updated user interface of the DCS for the Blaauw Observatory.

Dome Telescope Tracking

As mentioned in the previous section, KoepelX used to control the tracking of the dome. We move that logic to DomeCommanderX, previously exclusively used for the user interface of the control software. There we implement a thread – see [Veldhuis & Noel-Storr \(2021b\)](#) – using the control method, outlined in the previous section. The method needs to run in a separate thread in order to not block the user interface. The thread continuously pulls data from KoepelX and TheSkyX, to determine when the dome should move to a new position.

The User Interface

The graphical user interface (GUI) of the dome control software is shown in figure 19, and – as mentioned – was built using the Qt framework (version 5).

In the top of the user interface, in the *Dome control* box shown in Figure 19, we see the various commands that can be sent to the dome to control its position. It contains all the functionality that was available in the previous version, i.e., buttons to initialize and calibrate the dome (as mentioned in Chapter 2), to park the dome at a known distance from the calibration point, at the end of observations, the the stop button, which halt the dome movement and stops the dome from tracking the telescope, if tracking is engaged, a button to engage the tracking mode, and additionally, the dome can manually be moved to a given azimuth with the *go to* button.

The most noteworthy upgrade of the dome control GUI is the ability to select what aperture, or a combination of apertures, the dome should track, based on the observation needs. For instance, one might want to prioritise the finder scope

at the beginning of an observation session. In any case, when the track button is pressed, the dome will track the aperture selected from the drop-down menu, and thus base its position (azimuth) on what was determined to be the optimal azimuth for the selected aperture(s), which we determined in the previous chapter. When the tracking (mode) is engaged, we spawn a new thread controlling the position of the dome, based on the requirements outlined in the previous section.

Additionally, in the *Dome status* and *Telescope status* boxes, we present relevant information about the dome and telescope. The dome status consists of information of its position, whether the tracking (of the telescope) is engaged or not, whether or not it is currently moving, and a progress bar, which displays the time the dome should be able to remain at the position it last moved to, assuming that the dome is tracking the same right ascension and declination.

The telescope status, on the other hand, displays the right ascension, declination, the local sidereal time for the Blaauw Observatory, and the hour angle, as well as the CCD status (e.g. whether the CCD is exposed or not), as an indicator for when the dome should not be moving, given the electrical issues mentioned in the previous section.

However, at the moment of writing the thesis, we could not establish a connection with the CCD and readout its status, specifically whether it is exposed. Therefore, until we get access that information, the CCD status reads *To Be Determined* (TBD).

The information about the dome position is also conveyed via a top-down plot of the telescope and dome system, where the dome slit is marked by a rectangle that updates orientation every second. Furthermore, the telescope is shown as a line extending from the center of the dome to the center of the RC telescope, and another line indicating the pointing direction of the telescope. The length of the line segment indicating the direction depends on the altitude of the telescope, computed using Eq. 2, to give an indication of where the line-of-sight of the telescope intersects the dome. The altitude is also visualised by a red circle, centered at the origin of the telescope aperture, which grows with decreasing altitude and shrinks when the altitude increases. Therefore, practically, disappearing whenever the telescope is pointing to the zenith.

Lastly, we added a timestamped log to the GUI, which keeps the user updated whenever the dome control software does anything noteworthy, from switching aperture priority to reporting errors when, e.g., there is no available position for the dome such that the selected aperture is unobstructed.

6 Conclusions & Improvements

During this project, we have modeled the obstruction of the Gratama telescope by the hemispherical dome of the Blaauw Observatory using 3D CAD models and we developed a ray tracing method, which we used to determine the optimal dome azimuth angles given the hour angle and declination of the telescope. This allowed us to update the dome control software such that the dome no longer obstructs the Gratama telescope whenever it is tracking an object on the sky. Additionally, with the updated software, the dome does no longer have to track the telescope continuously, a major benefit over the previous system, causing the CCD to crash from time to time, whenever the dome would move while the CCD was exposed.

There are still, however, several areas within the project that could be improved: (1) the grid based control, (2) the speed of the code, (3) the fact that the lower shutter can remain closed, and (4) the dome server Koepe1X.

A rather large disadvantage of the method we developed in this thesis, is the fact that the new control system relies on a grid of (h, δ, A_d) data to position the dome. Consequently, because of limited resources, one can not compute the optimal dome azimuth on an infinitesimally spaced grid, therefore, leaving room for error in the optimal position of the dome. A dynamic solution, say a functional form, would circumvent these issues.

In the same vein, generating the (h, δ, A_d) grid is rather computationally intensive. This is evident in multiple layers of the code, starting with the – simplistic – ray tracing, which is already somewhat slow when running the code for a single hour angle and declination, and checking the obstruction for a single dome azimuth angle, with a large number of rays. Thus, performing that operation for a large three dimensional grid, with a 1° spacing, can easily take up to 12 hours, even when that code was already using parallelisation routines provided by the `joblib` package⁷, which is a package provides an easy-to-use interface for parallel computing.

The next part of the pipeline is generating the optimal azimuth grid from the obstruction data. Where, for each orientation of the telescope, there are a multitude of possible azimuth angles available, as mentioned in Chapter 4, which we verify one by one using standard for-loops, given that the code could not be easily vectorized, which could presumably have given a considerable time advantage.

Nevertheless, the *optimal azimuth generation code* could possibly be sped up by using a compiled language, such as C++, rather than Python, which for nested for loops could potentially up to 1330 times faster than Python, when iterating over a three-dimensional grid, according to Cai et al. (2005). Note, however, that the paper by Cai is 16 years old at this point, presumably Python has gained

⁷Joblib their GitHub repository: <https://github.com/joblib/joblib>

some speed improvements in the mean time. A more realistic option would be to rewrite the code using Cython, which can achieve similar speeds as C/C++, according to [Wilbers et al. \(2009\)](#), while simultaneously retaining most of Python's syntax, thus only requiring a minimal rewrite of the codebase.

We made the assumption that both shutters – the lower and upper shutters as shown in figure 4b – are always open when performing observations at the Blaauw Observatory. However, sometimes one might want to keep the lower shutter closed to counter light pollution. This would mean we would have to generate separate grids for this scenario.

Lastly, the code for the dome server is still written in Python 2, rather than Python 3, which was used to write the code for the new version of DomeCommanderX. It would be useful to upgrade the code to Python 3, even if only to circumvent the issue of managing multiple Python installations.

7 Acknowledgements

Many thanks to my supervisor Dr. Jake Noel-Storr, for their help throughout the project, and particularly for helping me make sense of my many – still existing – confusions regarding celestial coordinate systems. The project was a great experience, allowing me to apply and, to a certain extend, relearn theory from past courses, as well as giving me an opportunity to gain new skills, such as 3D CAD modelling with Inventor and writing GUI applications with the Qt framework.

References

- Cai X., Langtangen H. P., Moe H., 2005, *Scientific Programming*, 13, 31
- Conrad A. R., 2014, *Software Systems for Astronomy*. SpringerBriefs in Astronomy, Springer-Verlag, New York, doi:10.1007/978-1-4614-7058-8
- Denny R. B., 2002, *Society for Astronomical Sciences Annual Symposium*, 21, 39
- Duffett-Smith P., 1979, *Practical Astronomy with your Calculator*, 3 edn. Cambridge University Press, Cambridge
- Eberly D., 2006, *3D Game Engine Design*, 2 edn. CRC Press, doi:10.1201/b18212
- Mihelj M., Bajd T., Ude A., Lenarčič J., Stanovnik A., Munih M., Rejc J., Šlajpah S., 2019, *Robotics*, 2 edn. Springer
- Noel-Storr J., 2019, *PracticalAstronomyCrew/dome-control*: 2019 Release, doi:10.5281/zenodo.5054723
- Pérez García A. J., 2010, Master's thesis, Politechnika Ślaska
- RUG 2011, *Blaauw Observatory*, <https://www.rug.nl/research/kapteyn/sterrenwacht/>
- Selig J., 2006, *IEEE Robotics & Automation Magazine*, 13, 79
- Shirley P., Morley R. K., 2008, *Realistic Ray Tracing*, Second Edition. A K Peters, Ltd.
- Veldhuis M., Noel-Storr J., 2021b, *PracticalAstronomyCrew/dome-control*: 2021 Release, doi:10.5281/zenodo.5086408
- Veldhuis M., Noel-Storr J., 2021a, *PracticalAstronomyCrew/telescope-models*: Aperture Obstruction Models, doi:10.5281/zenodo.5086375
- Wilbers I. M., Langtangen H. P., Ødegård Å., 2009, *Proceedings of MekIT*, 9, 495

Appendices

A Additional Figures

Here we list a number of figures that we left out of the main part of the thesis, for the sake of brevity.

Obstruction with the Old Dome Control System

In addition to Figure 1, which shows the obstruction assuming the 2019 version of the dome control system, written by Noel-Storr (2019), we list the same figure but modelled for the autoguider and finderscope, respectively in Figures 20 and 21.

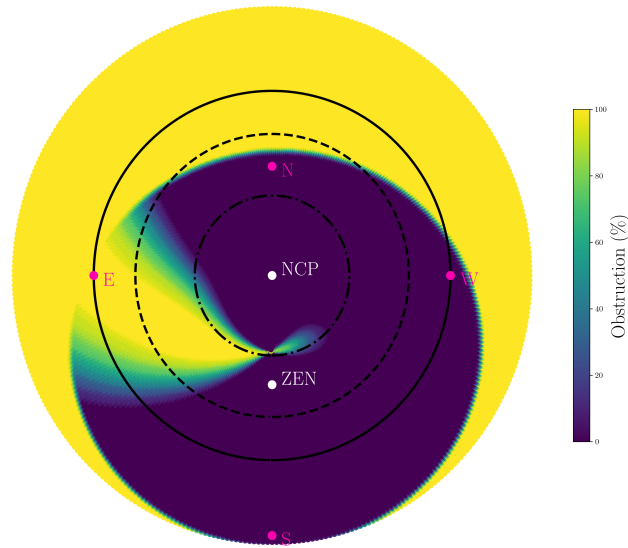


Fig. 20. The percentage of the aperture of the autoguider, modelled using obstruction code we developed in Chapter 4, assuming that the azimuth of the dome is given by Eq. 3, for hour angles from 0 h to 24 h and declinations from -40° to 90° .

Percentage of (h, δ) Pairs Versus the Number of Azimuth Solutions

In addition to Figure 16, we include Figures 22 and 23, showing the number of possible azimuth angles versus the percentage of (h, δ) pairs with that many solutions.

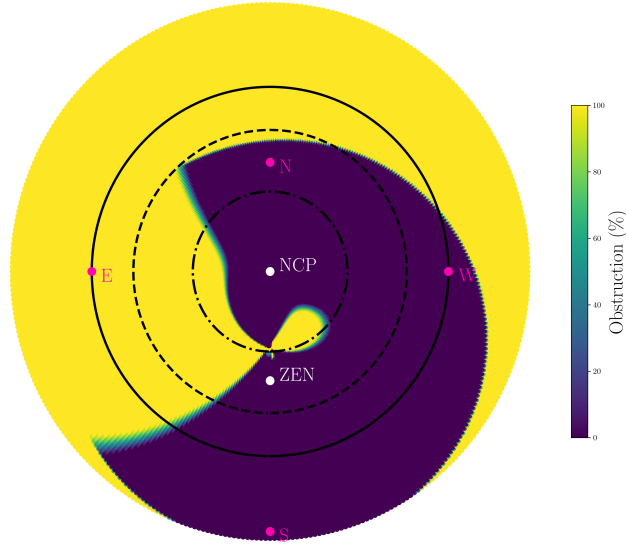


Fig. 21. The percentage of the aperture of the finderscope, modelled using obstruction code we developed in Chapter 4, assuming that the azimuth of the dome is given by Eq. 3, for hour angles from 0 h to 24 h and declinations from -40° to 90° .

The Optimal Dome Azimuth

In addition to Figure 17, we list the optimal azimuth for the autoguider, finderscope, and the combination of the telescope and autoguider (with the relaxed requirement on the obstruction, as stated in Chapter 5), respectively in Figures 24, 25, and 26.

The Hour Angle Distance

In addition to Figure 18, we list the *hour angle distance* for the autoguider, finderscope, and the combination of the telescope and autoguider corresponding to their optimal azimuth maps, respectively in Figures 27, 28, and 29. Note that the region with a maximum *hour angle distance* gets larger with decreasing aperture diameter. Additionally, note that Figure 28 has a vastly different structure than the other three aperture options, because we want the finderscope to be in the middle of the dome slit, rather than maximising the time the dome can remain at its optimal position.

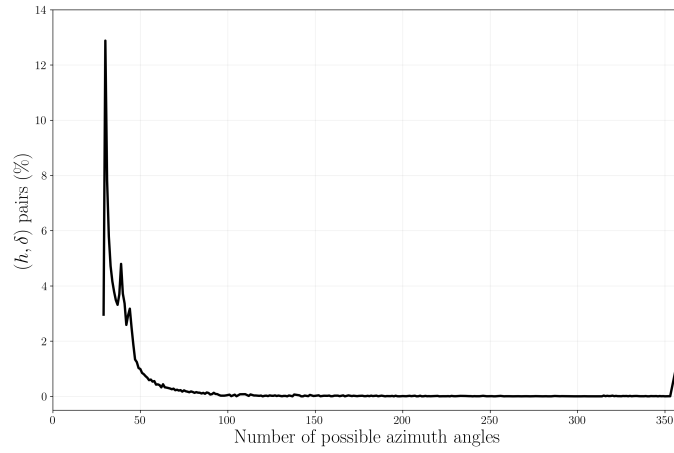


Fig. 22. The percentage of hour angle and declination pairs, as a function of the number of possible dome positions, such that the autoguider aperture is completely unobstructed by the dome.

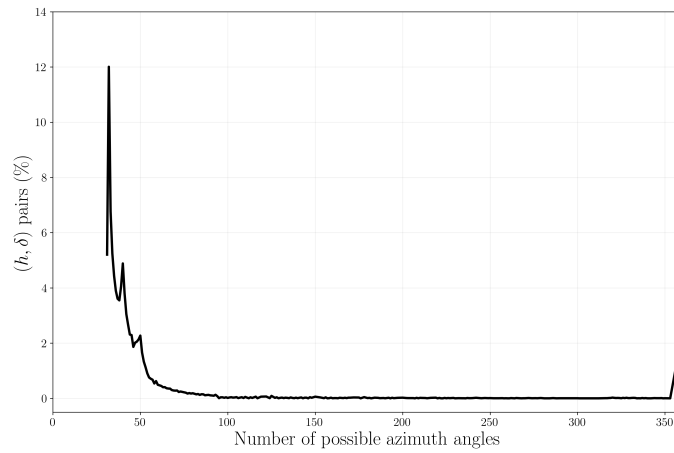


Fig. 23. The percentage of hour angle and declination pairs, as a function of the number of possible dome positions, such that the finderscope aperture is completely unobstructed by the dome.

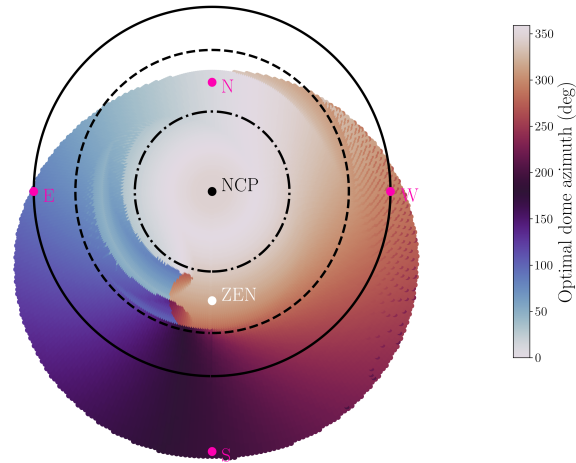


Fig. 24. Optimal dome azimuth for the autoguider.

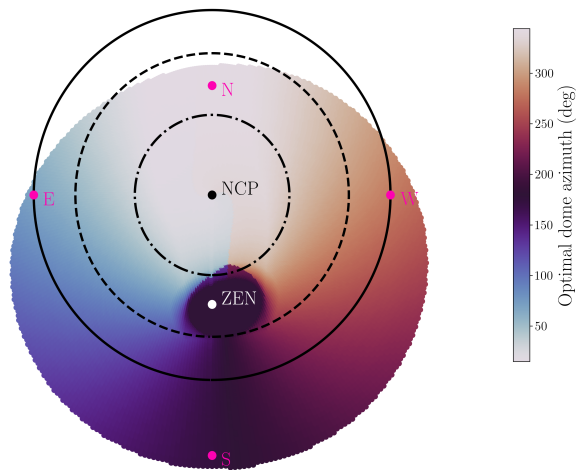


Fig. 25. Optimal dome azimuth for the finderscope, assuming that the finderscope needs to be positioned in the middle of the slit, rather than according to the *maximum hour angle distance* condition from Chapter 4.

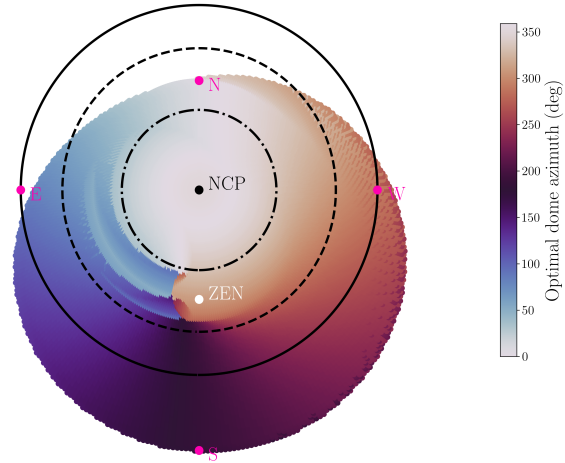


Fig. 26. Optimal dome azimuth for the RC telescope and autoguider, where – for the autoguider – we assumed a relaxed condition, whereby the autoguider could still be 50% obstructed in the worst-case scenario.

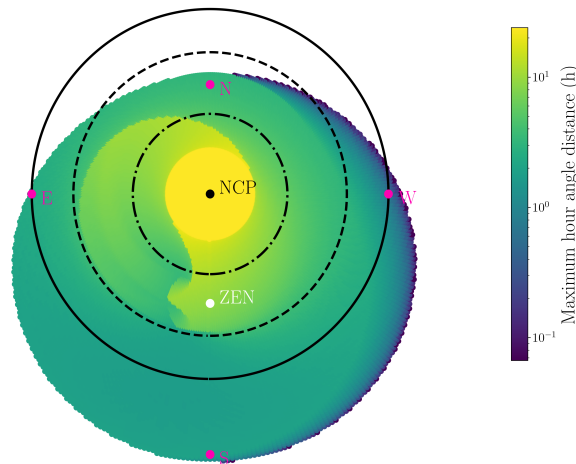


Fig. 27. Duration that the dome could potentially remain at its optimal position shown in Figure 24, assuming the autoguider is tracking an object with a fixed right ascension and declination.

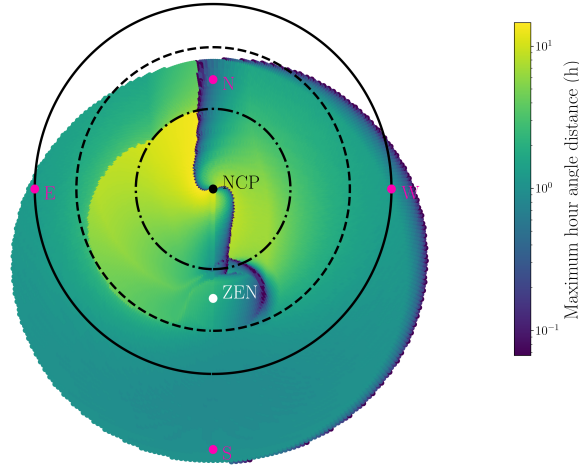


Fig. 28. Duration that the dome could potentially remain at its optimal position – in the center of the dome slit – shown in Figure 25, assuming the finderscope is tracking an object with a fixed right ascension and declination. Since the optimal azimuth condition is different for the finderscope, this graph looks vastly different than those for the telescope, autoguider, and telescope + autoguider combination.

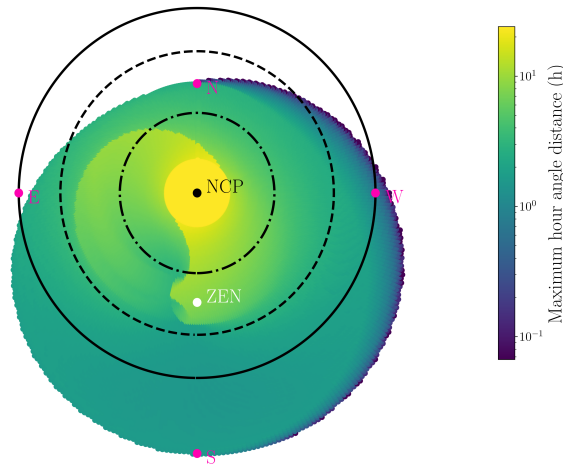


Fig. 29. Duration that the dome could potentially remain at its optimal position shown in Figure 26, assuming the telescope and autoguider are simultaneously tracking an object with a fixed right ascension and declination.

B Source Code

Aperture Obstruction Package

The code for quantifying the obstruction of an aperture by a hemispherical dome, in accordance with the method outlined in Chapter 4, can be found in the [telescope-models](#) GitHub repository, in the `bin/obstruction` folder. This package contains the `Aperture` class, located in `aperture.py`, which initialises an aperture with a certain radius, and if applicable the radius of the secondary obstruction, as is the case for the Gracemore telescope and autoguider. Furthermore, it contains a bunch of private methods, (1) to compute the coordinate transformations for the appropriate aperture using the equations derived in Chapter 3, (2) to sample points equidistantly in a circle, taking into account the secondary obstruction if applicable, (3) compute the pointing direction of the telescope using Eq. 32, and (4) determine whether the rays extending from the aperture are blocked using the conditions from Eq. 33.

Most important is the public method `obstruction`, which uses the four aforementioned private methods to compute the percentage obstruction of the aperture given a certain hour angle, declination, and dome azimuth; additionally, it can create a scatter plot showing the rays and marking whether they are obstructed or not. For specifics, e.g. function parameters, I would like to refer the reader to the documentation strings.

Additionally, in `aperture.py`, the package contains the functions `find_intersection` and `get_ray_intersection` for finding the intersection of a ray with a hemispherical dome (i.e., a half-capsule), according to the method from Chapter 4. Furthermore, the `Aperture` class relies on `transformation.py`, to compute the appropriate coordinate transformations, which contains a collection of functions to compute the translation and rotation matrices, and to convert between 3- and 4-vectors.

Code to Generate the Optimal Azimuth Grids

To generate the optimal azimuth grid we used to construct the updated DCS, we use the scripts `obstruction_grid.py` and `optimal_azimuth.py`, located in the `bin` folder of the [telescope-models](#) repository, by [Veldhuis & Noel-Storr \(2021a\)](#).

- The `obstruction_grid.py` script computes the percentage obstruction on a (h, δ, A_d) grid, for the desired aperture (the aperture of the telescope, the autoguider, or the finderscope), and stores the result in an `.npy` file.
- The `optimal_azimuth.py` script uses the `.npy` appropriate `.npy` file(s) to extract the optimal dome azimuth, by maximising the duration that the dome can remain at its previously moved to position; it stores the result in a `.csv` file.

MOCCA

Also located in the bin folder of the [telescope-models](#) repository, is the `mocca.py` script. This is a command-line utility that can be used to compute the percentage obstruction of an aperture and create a visualisation based on which rays are blocked or not. For more information, see the README or run the script with the `--help` flag.

Dome Control Software

The source code for the updated dome control software, by [Veldhuis & Noel-Storr \(2021b\)](#), can be found in the [dome-control](#) GitHub repository.

It contains `KoepelX.py`, which runs the dome server, responding to commands from the GUI. The updated tracking code is located in `DomeCommanderX.py`, where we added a new thread, sending a `goto` command to `KoepelX` whenever it should move to a new optimal position.

C Telescope & Dome Measurements

Here we list a drawing of the measurements of the telescope and dome, used to create the model in chapter 2, to determine the pointing in chapter 3, and determine the obstruction in chapter 4.

Telescope

For the telescope, we used primarily direct measurements using a tape measure at the Blaauw Observatory, to get the proper measurements of the mount. We summarise these measurements in Figure 30.

Dome

For the dome, we used a drawing – see Figure 31 kindly supplied by the FSE backoffice, and information listed on the Ash-Dome website.

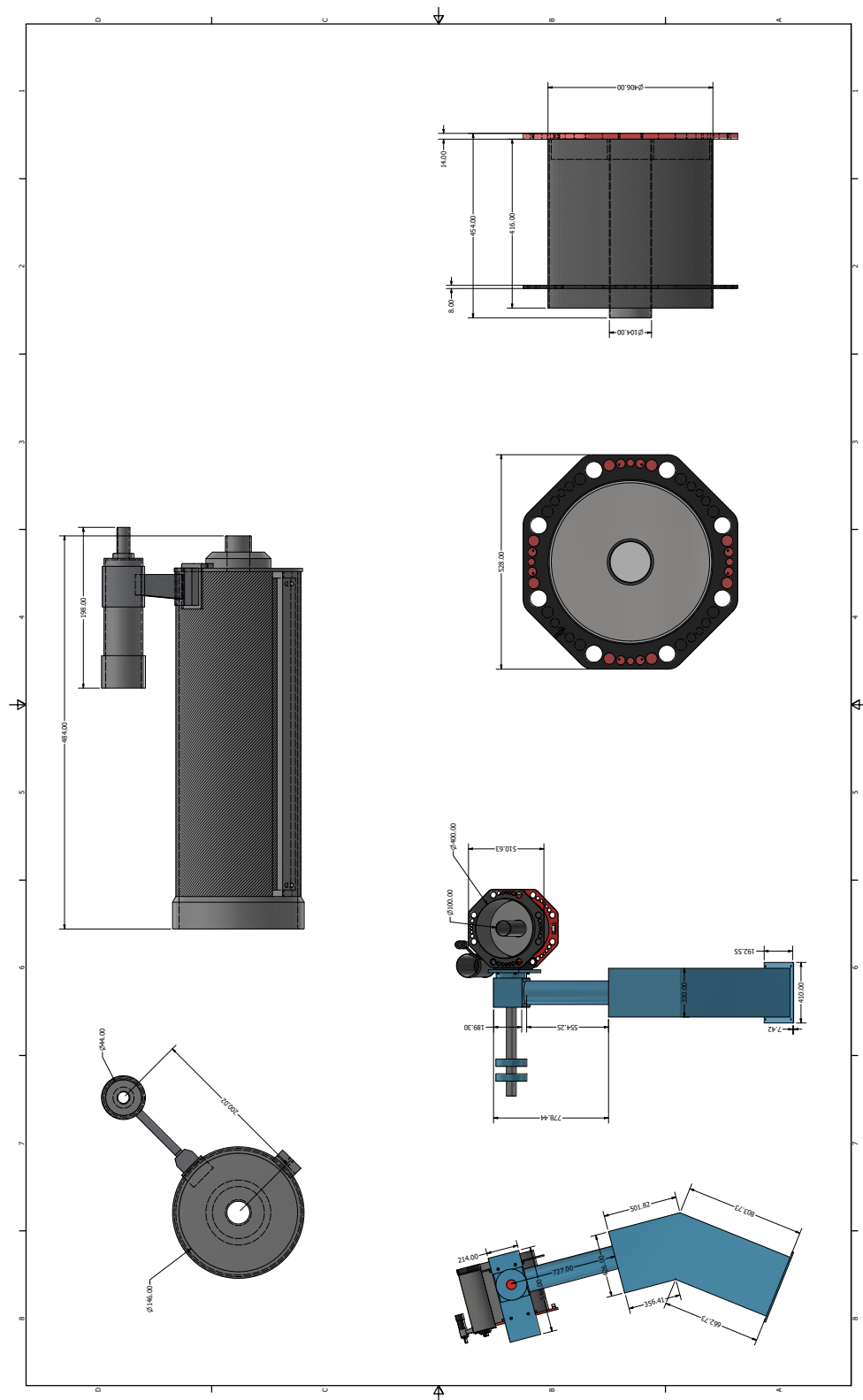


Fig. 30. Measurements of the Gratama Telescope model.

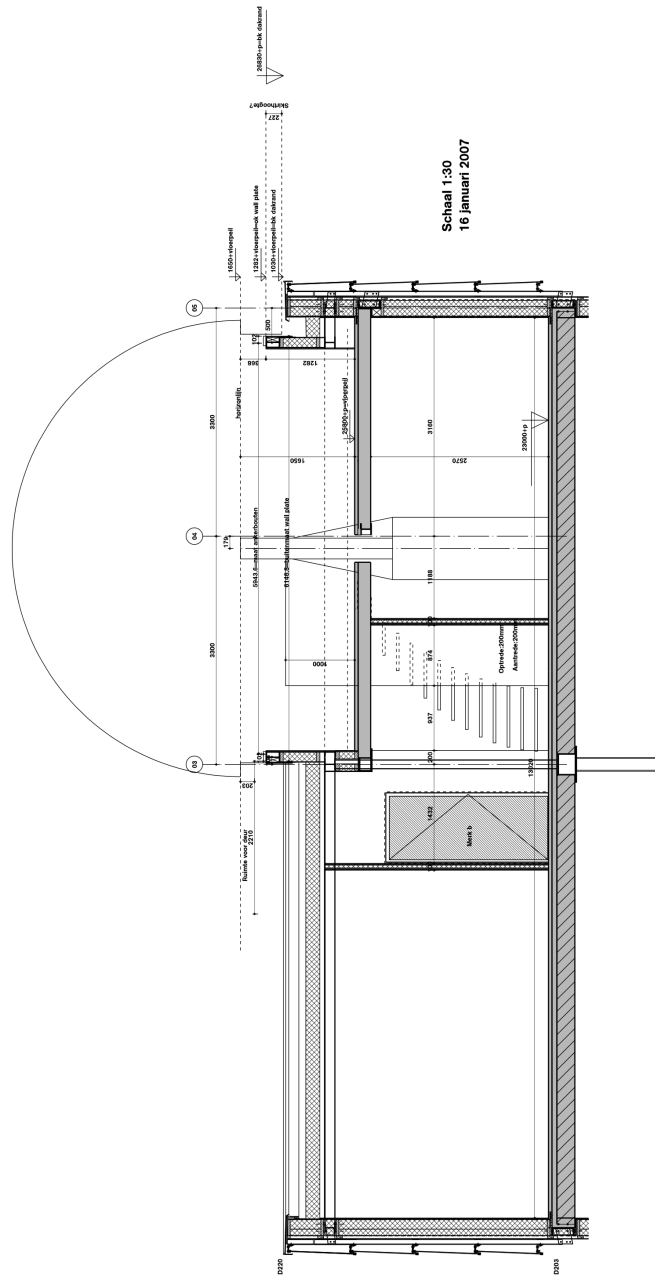


Fig. 31. Cross section of the dome of the Blaauw Observatory, with the control room below.

D 3D Models

For completeness, the 3D models of the Gratema telescope and Blaauw Observatory dome can be found in the [telescope-models](#) GitHub repository, in the `blaauw_3d` folder. The `Dome` and `Telescope` subfolders contain the individual parts of the dome and telescope and in the case of the telescope also the telescope assembly file, `Telescope.iam`. Most important is the observatory assembly file, `Observatory.iam`, containing the observatory as shown in Figures 7 and 8.

Note that the observatory assembly structure is divided into multiple folders, to be able to quickly change the visibility of specific parts of the model. These are the following folders, in the same order as they appear in the assembly:

- `Dome`: contains the wall and hemisphere of the dome. One can use the cylindrical joint `AzAxisRotation` to set the azimuth angle of the dome, note that in the model 0° degrees corresponds to the south.
- `TelescopeAxes`: contains the parts of the telescope with the rotational joints, controlling the hour angle and declination of the telescope. They are respectively called `RaAxisRotation` and `DecAxisRotation`, and the rotation direction is opposite of the regular conventions, as explained in Chapter 3.
- `TelescopeStructure`: contains the (many) parts of the telescope that do not control the rotation.
- `Beams`: contains the red cylindrical beams, as shown in Figure 8.