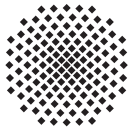


University of Stuttgart
Germany

Institute of
Robust Power
Semiconductor Systems

Student Research Project

Search Algorithm for the Ground Station Antenna of the EIVE Satellite

Mustafa Efe Erdogan

Supervisor: Laura Manoliu, M. Sc.
Prof. Dr.-Ing. Ingmar Kallfass

Period: 17.04.2023 – 17.9.2023

Stuttgart, 17.9.2023

Postanschrift: Institut für Robuste Leistungshalbleitersysteme
Pfaffenwaldring 47

D-70569 Stuttgart

Tel.: +49 (0) 711 685 68700
Fax.: +49 (0) 711 685 58700
E-Mail: sekretariat@ilh.uni-stuttgart.de
Web: www.ilh.uni-stuttgart.de

Declaration

I hereby declare that this thesis is my own work and effort and follows the regulations related to good scientific practice of the University of Stuttgart in its latest form. All sources cited or quoted are indicated and acknowledged by means of a comprehensive list of references.

Stuttgart, 17.9.2023

Mustafa Efe Erdogan

Executive Abstract

Satellite technologies have rapidly become integral to modern life, used for broadcasting, navigation, communication, and Earth observation. As technology advances, satellites generate increasingly large volumes of data, presenting hardware challenges in terms of data storage and transmission due to limited power sources. To meet the escalating demand for high-capacity channels with high data rates, the EIVE project was initiated to explore the feasibility of using E-band frequencies (71 −76 GHz) for satellite communication. EIVE, led by the University of Stuttgart in collaboration with various partners, aims to test data transfer capabilities in this uncharted frequency range. A key challenge is the establishment of a communication link between a ground station antenna and the LEO satellite in the EIVE project. This task is compounded by the ground station's Cassegrain antenna with a narrow HPBW of 0.23° and a low achieved scanning area because of the LEO.

To address these challenges, this research thesis introduces a CONSCAN based search algorithm, which expands the antenna's scanning area by executing conical patterns around the satellite's estimated trajectory. By using quaternion rotations and the Orekit library for trajectory estimation, this algorithm significantly enhances the search capabilities, increasing the scanning area in the sky.

Furthermore, the research highlights the importance of continuous signal acquisition from the satellite for the planned data transfers. To tackle this issue, the groundwork for a tracking algorithm based on the MMT method is introduced to provide precise measurements of the satellite's position, combining the MMT method with Kalman filters and GPS based methods. This novel method promises the improvement of the accuracy and stability of the system.

Zusammenfassung

Satellitentechnologien sind aus dem modernen Leben nicht mehr wegzudenken und werden für Rundfunk, Navigation, Kommunikation und Erdbeobachtung eingesetzt. Im Zuge des technologischen Fortschritts erzeugen Satelliten immer größere Datenmengen, was aufgrund begrenzter Energiequellen eine Herausforderung für die Hardware in Bezug auf die Datenspeicherung und -übertragung darstellt. Um die steigende Nachfrage nach Kanälen mit hoher Kapazität und hohen Datenraten zu befriedigen, wurde das Projekt EIVE initiiert, um die Machbarkeit der Nutzung von E-Band-Frequenzen (71 –76 GHz) für die Satellitenkommunikation zu untersuchen. EIVE, das von der Universität Stuttgart in Zusammenarbeit mit verschiedenen Partnern geleitet wird, zielt darauf ab, die Möglichkeiten der Datenübertragung in diesem unerforschten Frequenzbereich zu testen. Eine zentrale Herausforderung ist die Herstellung einer Kommunikationsverbindung zwischen einer Bodenstationsantenne und dem Satelliten im Projekt EIVE. Diese Aufgabe wird durch die Cassegrain-Antenne der Bodenstation mit einem schmalen HPBW von 0,23° und einem geringen erreichten Scanbereich aufgrund des LEO erschwert.

Um diese Herausforderungen zu bewältigen, wird in dieser Forschungsarbeit ein auf CONSCAN basierender Suchalgorithmus vorgestellt, der den Scanbereich der Antenne durch die Ausführung von Kegelmustern um die geschätzte Flugbahn des Satelliten vergrößert. Durch die Verwendung von Quaternion-Drehungen und der Orekit-Bibliothek für die Flugbahnschätzung verbessert dieser Algorithmus die Suchmöglichkeiten erheblich und vergrößert den Scanbereich am Himmel.

Darüber hinaus unterstreicht die Forschung die Bedeutung einer kontinuierlichen Signalerfassung vom Satelliten für die geplanten Datenübertragungen. Um dieses Problem anzugehen, werden die Grundlagen für einen auf der MMT Methode basierenden Verfolgungsalgorithmus eingeführt, der präzise Messungen der Satellitenposition ermöglicht und die MMT Methode mit Kalman Filtern und GPS basierten Methoden kombiniert.Diese neuartige Methode verspricht eine Verbesserung der Genauigkeit und Stabilität des Systems.

Contents

List of Figures

List of Tables

1. Introduction

Satellite technologies are just at the center of this fast-developing trend and therefore become integral parts of our daily lives. Satellites are used by both the public and commercial sectors for broadcasting, worldwide navigation, high-speed communication, Earth observation, and many more. Following this trend, the capability and the quality of the used equipment on the satellites developed quickly in recent years. For example, high-resolution infrared/optical cameras or new radar systems generate data at the rate of gigabits per second. The generated high data volumes result in hardware challenges in storing and transmitting the data considering that most satellites have a limited power source generally supplied by solar panels. In light of these technical challenges, it is unsurprising that there is a rapid escalation in the demand for high-capacity channels characterized by high data rates. Harati et al. [6] demonstrated the practicability of employing the E-band frequency range (71 – 76 GHz) for satellite communications, which extends beyond the conventional spectrum exploited by satellite communications, which typically encompasses the L-band (1 – 2 GHz), the S-band (2 – 4 GHz), the C-band (4 – 8 GHz), the X-band (8 – 12 GHz), the Ku-band (12 – 18 GHz), the Ka-band (26 – 40 GHz), the Q-band (33 – 50 GHz), and the V-band (50 – 75 GHz). Greater bandwidths are frequently attainable through higher frequency ranges, consequently inducing augmented data rates. Nonetheless, this gain in performance necessitates strengthened precision in aligning the antennas of both the satellite and the ground station. Additionally, this configuration is often susceptible to increased signal degradation [7]. In parallel, signal interference between the satellites has gained prominence, particularly in the lower frequency bands with the increasing number of satellites. The escalating population of satellites in the Earth's orbital space has prompted the exploration of novel technologies to mitigate such interference and enhance overall efficiency [8]. To address these new challenges, the Exploratory In-Orbit Verification of an E/W-Band Satellite Communication Link (EIVE) project was initiated to examine the feasibility of a communication link in the E-band. A model of the EIVE can be seen in figure 1.1.

Figure 1.1.: Model of the EIVE satellite [1]

The University of Stuttgart is in charge of the CubeSat project EIVE in cooperation with Fraunhofer Institute for Applied Solid State Physics, Radiometer Physics GmbH, TESAT, and Thales Alenia Space. The Institute of Space Systems and the Institute of Robust Power Semiconductor Systems are responsible for the design of the satellite bus and the payload. The 6U CubeSat developed by the Institute of Space Systems is launched to orbit in June 2023 and is active in the Low Earth Orbit (LEO). In an in-orbit verification, the data downlink from the CubeSat platform will first show the viability of broadband radio connections in a frequency range new to satellite communications. With the proposed payload, it then targets Earth observation applications and ultra-high throughput services. It also assesses the impact of the atmosphere on modulated data in this frequency range. Moreover, the payload consists of an E-band transmitter and a high-resolution camera to generate sufficient data to fully utilize the available bandwidth. Additionally, the satellite bus has an additional S-band (2 –4 GHz) transceiver to send telemetry to the ground station and receive telecommands. Visualization of the EIVE mission can be seen in figure 1.2.

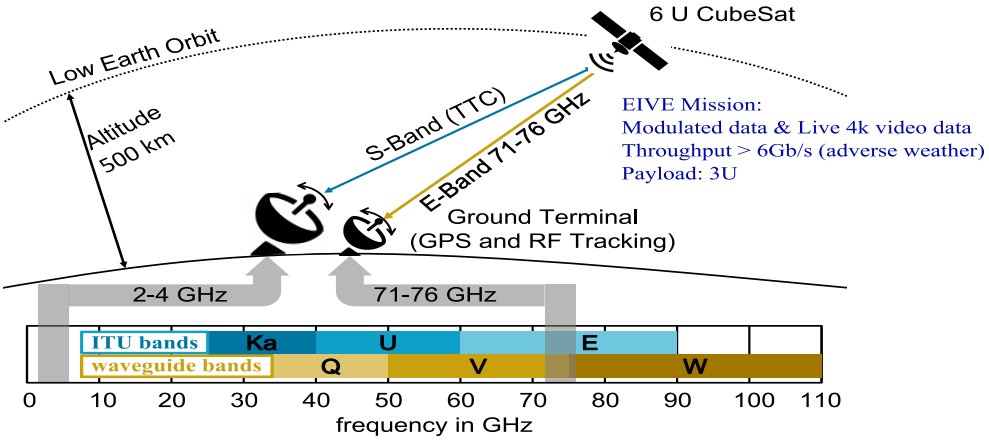


Figure 1.2.: Visualization of the EIVE mission [2]

A Cassegrain antenna with a diameter of 1200mm is installed at the university to establish the E-band communication link between the satellite's transmitter and the ground station antenna's receiver. The required high antenna gain leads to a consequential reduction in the Half Power Beamwidth (HPBW) at a mere 0.23° [2]. This characteristic outlines a narrow effective radiated field in the sky that hinges on the satellite's elevation and trajectory. Thus, a high precision system is required to find and track the satellite.

The phenomenon of a constrained effective radiated field, along with the intricacies of tracking a LEO satellite, are longstanding challenges within the domain of satellite communications. Over time, diverse methodologies have been devised to confront these issues. Some approaches involve the utilization of multiple antennas or/and multiple satellites, exemplified by techniques leveraging the Doppler effect as outlined by Pedrosa et al. [9] and Neinavaie et al. [10]. Others emphasize alternate strategies, such as amplitude-based signal analysis through electronic beam squinting or the sequential amplitude sensing method, as elaborated by Hawking et al. [11]. The most straightforward approach involves aligning the antenna with the satellite's projected orbital path based

on precomputed data, and subsequently attempting to capture any incoming signals. However, in this method, the antenna is directed blindly into the sky, lacking any confirmation of the correctness of its orientation until a signal is actually detected from the satellite. In contrast, more intricate systems use the received signal to determine the deviation between the antenna pointing and the satellite's actual position at the time of signal reception.

The selection of a tracking and searching methodology depends significantly on the specific equipment in use, the system's requirements, and the mission objectives. Considering EIVE only has one Cassegrain antenna available and a very narrow HPBW, it is very crucial that the search area is increased in the sky and a tracking system is utilized that requires only one antenna. Moreover, time plays a critical role as well since the satellite will be visible from the antenna for approximately 10 minutes in each pass. Given the intention to transmit gigabytes of data during each satellite pass, it is prudent to assign a sufficient duration to the data transfer period. Therefore, the detection and tracking of the satellite should be performed as quickly as possible. This research thesis focuses on the conical scanning algorithm (CONSCAN) and the novel multimode tracking (MMT) method to tackle the given problems in the EIVE project. A CONSCAN based algorithm is used to detect the satellite. The antenna's beam is rotated around the already predicted trajectory of the satellite in a conical motion, which increases the searching area and simultaneously decreases the effect of a possible error in the trajectory estimation. The MMT takes advantage of two radiation patterns, one with a maximum at the boresight of the antenna and one at zero, converting this information to the pointing error of the antenna in azimuth and elevation and then correcting the antenna's pointing accordingly. Additionally, Kalman filters (KF) will also be discussed as possible improvement to currently used methods.

CONSCAN and MMT were studied and addressed in this research thesis with a possible additional technique to enhance the tracking system using the KF. However, the focus lies on primarily the CONSCAN based search algorithm. First, the theoretical foundation for the searching and tracking system is elaborated in chapter 2. Afterward, the used search algorithm and its implementation is explained and provided in chapter 3, and the findings of the simulations and tests are displayed and examined in chapter 4. In chapter 5, potential improvements and future work are discussed. Lastly, the entire study is summarized and ended in chapter 6.

2. Theoretical Background and Literature Review

The theoretical background on the used search algorithm is given in this chapter. The used coordinate systems are defined in section 2.1, the basics of tracking and search algorithms are discussed in section 2.2, the method used to estimate the trajectory of the satellite is explained in section 2.3, and a basic understanding of the KF is given in section 2.4.

2.1. Coordinate Systems

The used antenna's control system accepts only azimuth and elevation values as input. Therefore, the horizontal coordinate system (HCS) is used as the main coordinate system to define the position of the antenna and satellite. The horizontal coordinate system is a topocentric celestial system that uses the observer's local horizon as the fundamental plane to define two angles [12]. The azimuth is the angle between an object's vector on the horizontal plane and its vector in the direction of the North Pole. It defines an object's horizontal motion in space. It is expressed in degrees, moving clockwise from north to south. Besides that, an object's vertical orientation in space is described by its elevation. It is the angle formed between the vector of the object on the vertical plane and the horizon plane. Thus, an object in the zenith has an elevation of 90°. An object with a negative elevation is not visible to an observer since it is below the horizon. The HCS is visualized in 2.1.

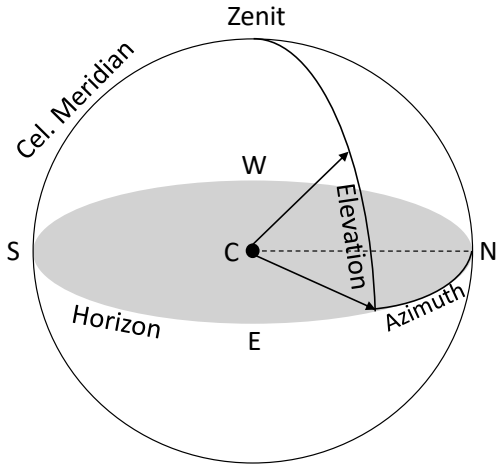


Figure 2.1.: Horizontal coordinate system [3]

The Earth-fixed coordinate system (ECEF) is also used for some of the computations in the algo-

rithm. It is a Geocentric coordinate system that depicts locations close to the Earth as X, Y, and Z (Cartesian Coordinates) measurements from its center of mass [12]. This includes the planet's surface, interior, atmosphere, and surrounding space. The North-South Polar axis is the Z axis, and it has only positive values increasing northward. The X axis runs through the origin in the plane of the equator, from the prime meridian (positive) to 180° of longitude (negative). Along with being on the equator's plane, the Y axis also runs through a region that stretches from 90°W (negative) to 90°E (positive) in longitude. The visualization of the ECEF is shown in figure 2.2.

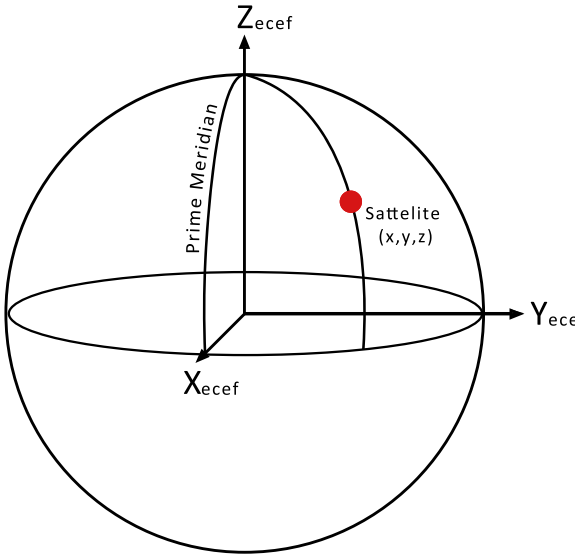


Figure 2.2.: Earth-fixed coordinate system

The main difference between the HCS and ECEF is the origin of interest. ECEF defines its origin as the earth's center; meanwhile, HCS defines its origin as the observer's position. They are both used for satellite tracking applications. During the implementation of the algorithm, translations between ECEF and HCS were conducted depending on the performed action. The ECEF is employed to compute the trajectory of the satellite, the antenna's pointing direction, and the beam's center position, whereas the HCS is used to translate the antenna's rotation in azimuth and elevation, and compute the input values for the antenna control system. The coordinate system transformations are conducted with the open source Python library 'Pymap3D' [13]. Pymap3D is employed by various projects worldwide, offering solutions to coordinate system transformation problems.

2.2. Satellite Tracking and Detection

In the case of the EIVE, establishing the E-band link between the antenna and the satellite consists of two dependent parts. First, the satellite should be detected when it enters the line of sight. Second, it should be tracked after its localization is completed. To achieve these two tasks, conscan and MMT are used. Before receiving any signals from the satellite, the antenna performs the blind search.

The antenna points a circular motion around the expected horizon position of the satellite during the blind search. After some time or the first detection, the antenna initiate the CONSCAN. During the CONSCAN, the antenna moves in conical motion along the predicted trajectory and tries to detect the satellite multiple times until the error between the satellite's position measurement and prediction is under a certain threshold. When the threshold is reached, MMT will start running and tracking the satellite in a control system. If the satellite is lost during the MMT, the CONSCAN will be activated again and search for the satellite. In summary, CONSCAN is used to conduct the localization, and MMT is used for tracking. The CONSCAN is explained in chapter 2.2.1, and the MMT is explained in chapter 2.2.2.

2.2.1. Conical Scanning Algorithm

Even though we predict and compute the possible trajectory of the satellite as explained in chapter 2.3, the antenna's pointing accuracy is affected by the environmental disturbances or antenna's mechanical limitations, such as its acceleration power [14]. Even wind or the weight of the payload can play a decisive role in the precision of the pointing. CONSCAN is a technique used to undermine these effects and detect the true position of the satellite. In the case of the EIVE, the CONSCAN primarily focuses on first founding the satellite and then performing a high accuracy localization during which the scanning area increases and the effect of the narrow beamwidht decreases. Furthermore, the performed conical movements during the CONSCAN causes variations in the received signal. These variations are then used to estimate the true position of the satellite [15]. The conical motion of the antenna around the trajectory is demonstrated in figure 2.3.

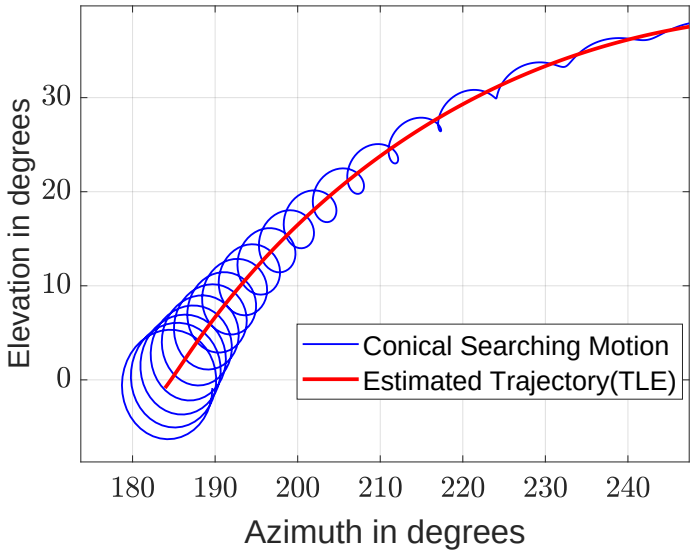


Figure 2.3.: Conical motion of the antenna around satellite's trajectory

The CONSCAN involves the addition of harmonic movements in the azimuth and elevation axes,

causing the antenna to follow the satellite in a circular pattern with an angular velocity of w and a radius of r. Depending on the radius and the antenna's capabilities, a sampling rate Δt with fixed frequency should be set, which also satisfies the Nyquist-Theorem [15]. The frame of reference for the circular motion is given in figure 2.4. The origin represents the original antenna direction, s_k denotes the location of the spacecraft, $\hat{s}_k$ denotes the estimated spacecraft position, a_k denotes the position of the antenna throughout the scan, and e_k is the vector indicating the difference between the spacecraft and antenna positions. The definition of the pointing error is given in equation 2.1 where the sub indexes e and a refer to elevation and azimuth axes.

$$e_k = s_k - a_k = \begin{bmatrix} e_{a,k} \\ e_{e,k} \end{bmatrix} = \begin{bmatrix} s_{a,k} - r\ cos(w\ t_k) \\ s_{e,k} - r\ sin(w\ t_k) \end{bmatrix} \tag{2.1}$$

The estimation of the true position with the help of the variations in the signal power can be done in different ways. The conventional way proposed by Gawronski and Craparo [15] with least mean square estimation (LMS) is explained in 2.2.1.1, and a different estimation method proposed by Souza Et al. [16] with KF is explained in 2.2.1.2. The method with the KF provides a strong improvement to the conventional way since it is not required to assume the position of the satellite and the carrier power constant. Additionally, it presents a model that allows one to estimate the position even without any measurements.

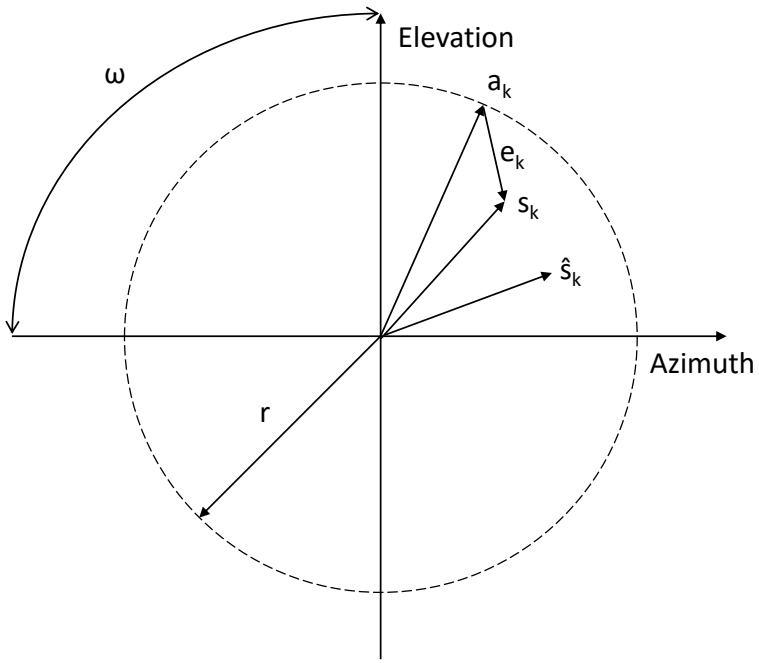


Figure 2.4.: Frame of reference

2.2.1.1. Least Mean Square Estimation

In Gawronski's and Craparo's work [15], the received signal power is defined as in equation 2.2

$$p_k = p_{0,k}\, exp(-\mu\, \frac{\epsilon_k^2}{h^2}) + v_k \tag{2.2}$$

, where $p_{0,k}$ is the carrier power, $\mu = 4ln2$ is a constant, h is the antenna beam width, v_k is an additive Gaussian white noise, and ϵ_k^2 is the squared total error. ϵ_k^2 is given in equation 2.3.

$$\epsilon_k^2 = e_k^T\, e_k = s_k^T\, s_k - 2a_k^T\, s_k + r^2 \tag{2.3}$$

The received power varies during the scan in a sinusoidal pattern as a result of the harmonic motions, and its amplitude is dependent on the position of the spacecraft. Using the first-order Taylor expansion of $exp(x)$, that is $exp(x) = 1 + x$, eq. 2.2 becomes:

$$p_k = p_0(1 - \mu\, \frac{\epsilon_k^2}{h^2}) + v_k \tag{2.4}$$

. If the equations 2.1 and 2.3 are combined, equation 2.4 can be written as

$$p_k = p_m + \frac{2p_0\mu r}{h^2}\, (s_a cos(wt_k) + s_e sin(wt_k)) + v_k \tag{2.5}$$

, where

$$p_m = p_0(1 - \frac{\mu}{h^2}(r^2 + s^T s)) \tag{2.6}$$

. Subsequently, the difference between the received power and the mean power can be written as in 2.7.

$$dp_k = p_k - p_m = gs_a cos(wt_k) + gs_e sin(wt_k) + v_k \tag{2.7}$$

Afterward, a new parameter g is defined as in 2.8.

$$g = \frac{2p_0\mu r}{h^2} \tag{2.8}$$

Finally, if $m_k = g\begin{bmatrix} cos(wt_k) & sin(wt_k) \end{bmatrix}$, equation 2.7 can be rewritten as in equation 2.9.

$$dp_k = m_k s + v_k \tag{2.9}$$

Gathering measurements during the scan period, the true position can be estimated using the LMS as given in equation 2.10.

$$\hat{s} = [(M^T M)^{-1} M^T] dp \tag{2.10}$$

LMS assumes that the spacecraft's position will remain constant throughout the scan duration with a constant carrier power. This assumption holds if the measurements are taken at regular intervals, and the distance between the spacecraft and the ground station is sufficient to allow for gradual changes in the position of the spacecraft. However, in LEO applications like the EIVE, the assumption of constant position and constant carrier power during the sampling period is not applicable since the satellite in LEO moves fast [16], [17]. For example, EIVE moves with a speed of 7 km/s, making the constant position assumption absolute. Therefore, a constant position cannot be assumed, and LMS cannot be employed for the EIVE.

2.2.1.2. Kalman Filter Estimation

The proposed technique by Souza [16], [17] employs the KF to estimate s_k. The basics to KF are explained chapter 2.4. This approach does not use the Taylor series to estimate the signal power p_k; instead, a mathematical adjustment is used to prevent linearization mistakes. The proposal monitors not only the position of the spacecraft, but also a variable linked to the carrier power. Furthermore, the state vector is expanded by its first-order time derivatives, making it more resistant to fluctuations in the state vector dynamics. Equation 2.2 may be transformed into the logarithmic function by neglecting v_k and treating it as a deterministic function, in which case it becomes:

$$ln(p_k) = ln(p_{o,k}) - \mu \frac{\epsilon_k^2}{h^2} \tag{2.11}$$

.It is necessary to attribute all of the model's uncertainty to the Gaussian white noise w_k with a zero mean and covariance $W = \sigma_p^2$. Following that, the equation 2.11 can be rewritten as in equation 2.12.

$$ln(p_k) = \Delta k + \frac{2\mu r}{h^2}(s_a cos(wt_k) + s_e sin(wt_k)) + w_k \tag{2.12}$$

The Δk is defined as in equation 2.13.

$$\Delta k = ln(p_{0,k}) - \frac{\mu}{h^2}(s_{a,k}^2 + s_{e,k}^2 + r^2) \tag{2.13}$$

The state vector X_k is composed of the state variables Δk , $s_{a,k}$, and $s_{e,k}$, and it is extended by the first-order time derivatives of the state variables. X_k is given in equation 2.14.

$$X_k = \begin{bmatrix} \Delta k & \dot{\Delta k} & s_{a,k} & \dot{s_{a,k}} & s_{e,k} & \dot{s_{e,k}} \end{bmatrix}^T \tag{2.14}$$

The evolution function is modeled as given in equation 2.15.

$$X_k = \begin{bmatrix} 1 & T_s & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T_s & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} X_{k-1} + q_{k-1} \tag{2.15}$$

A Gaussian white noise vector q_{k-1} with a covariance matrix Q is used in this equation. The time derivative variables are represented as constants disturbed by random noise. It should also be pointed out here that the standard deviation of the spacecraft position depends on the application, modeled as the maximum drift in one period, and the standard deviation of Δ depends on the sensor standard deviation σ_p. The measurement function is linear concerning the state variables, and it can be written as in equation 2.16.

$$ln(p_k) = C_k\, X_k + w_k \tag{2.16}$$

C_k is defined as given in equation 2.17.

$$C_k = \begin{bmatrix} 1 & 0 & \frac{2\mu r}{h^2} cos(wt_k) & 0 & \frac{2\mu r}{h^2} sin(wt_k) & 0 \end{bmatrix} \tag{2.17}$$

The state vector can be calculated using the KF in each sample period using the suggested formulation. The carrier power is not directly estimated by the filter, and only the spacecraft position is. The carrier power may be calculated using the predicted state using equation 2.13. This method allows the estimation of the satellite's position and carrier power even in the absence of any measurement. Moreover, there is no deterministic way to define the initial covariance matrix. Hence, it must indicate a high degree of uncertainty considering reasonable values.

2.2.1.3. Implementation Methods for Conical Scanning Algorithm

Like every algorithm, CONSCAN describes a problem and then tries solving it with different mathematical approaches. It is commonly used for adaptive tracking of an object whose position is not precisely known. During the first years of its development, fast Fourier transformation (FFT) was used as the base of this algorithm. However, Eldred [18] proved that a KF based algorithm outperforms the FFT-based one. KF based approach is proven to include better accuracy of the estimate, nearly continuous updates, and the ability to accumulate missing data, while it still has the same robustness as the FFT-based one. Consequently, FFT became irrelevant in the case of the CONSCAN.

KFs are a model predictive approach. The computed estimations are influenced by both the measurement and the prediction. Thus, the used model is highly influential in the behavior of the algorithm. The mathematical model in the CONSCAN describes a circular motion performed by

the antenna that compensates for its manufacturing imperfections and other disturbances. During the circular motion, the sinusoidal variations in the power of the received signal define the model's prediction method, which is used to estimate the true spacecraft position. This prediction model based on the power variations is given in equation 2.2 in chapter 2.2.1.1. This equation can be solved both in a linear and a nonlinear form. The linear solutions using LMS or the KF are shown in chapters 2.2.1.1 and 2.2.1.2. In the work of Chen Et al. [19], it is proven that the problem can be transformed into a linear one, and a general solution to this problem with the LMS can be given. It was also demonstrated that it is possible to define the problem as a nonlinear one with one "exact" solution based on Markov Chain Monte Carlo (MCMC) method. Their findings showed that linear approaches match the MCMC method in terms of performance when the amount of data gathered in practice is the same. The nonlinear approach only outperforms linear methods in terms of accuracy when the amount of data obtained is reduced, and the degree of noise is raised. The MCMC provides only 10 % error reduction compared to linear methods. In addition, this performance increase comes in the cost of computation power. For the EIVE, a linear solution to the problem is used to simplify the model since the nonlinear solution does not provide a significant performance increase, and it is not worth to increase the computation power.

2.2.2. Multimode Tracking

With a tracking speed of up to three degrees per second and pointing precision comparable to radio telescopes, tracking a fast-moving satellite (7 km/s) with a high-gain beam antenna is challenging. In this case, the standard solution to tracking problems, such as the GNSS-based tracking with TLE data, proves not sufficient because of its pointing error, explained in chapter 2.3. Therefore, a tracking method with higher accuracy and precision should be utilized for the EIVE. Multimode tracking is a simultaneous sensing method that allows the evaluation of the pointing error at any given time using only one antenna [3], making it a suitable candidate for LEO satellite tracking. Furthermore, the MMT requires at least two radiation patterns. The output magnitude of the communication signal is highest when the antenna is pointed directly at the point signal source and receives an incident wave [20]. On the contrary, elevation of the antenna feed's boresight axis from alignment with the point source results in the excitation of higher order modes within the waveguide [20]. The possible radiation patterns are demonstrated in figure 2.5.

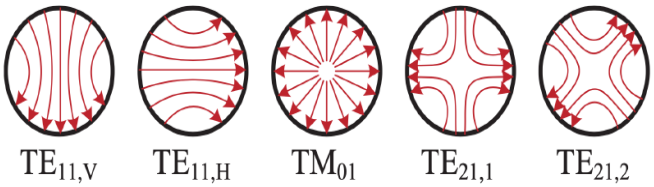


Figure 2.5.: Electrical field distribution of different normal modes within the waveguide [3]

Two modes are utilized in the employed tracking system, with the main mode having its main lobe

at the boresight and the other having a zero at the boresight. Consequently, only one waveguide is enough to generate the required radiation patterns. Moreover, the fundamental mode TE_{11} combining the $TE_{11,V}$ and $TE_{11,H}$ radiation patterns are employed to transmit the signal, whereas the second mode of higher order TE_{21} combining the $TE_{21,1}$ and $TE_{21,2}$ radiation patterns are employed to sense and detect the deviation. During the tracking, the signals from both modes are detected and fed into an IQ mixer. The offset and direction of the signal are defined by the amplitude and the phase shift between the two modes [21].

As a closed loop approach, MMT allows for constant evaluation and correction of pointing errors. The error correction process is done digitally with the help of a RaspberryPi and an analog digital converter (ADC). The ADC has a throughput rate of 1 MSa/sec and 14 bit resolution, allowing the discretization of the analog signals in $2^{14} - 1 = 16383$ steps. However, the closed-loop tracking mechanism confronts two principal challenges. Firstly, the transmitter antenna on the satellite has a gain of $33dBi$ and an HPBW of 3°, whereas the receiver Cassegrain antenna has a higher gain of $59dBi$ and an HPBW of only 0.23°. The beams of the satellite and the antenna are visualized in figure 2.6. Consequently, the satellite's beam is consistently positioned above the ground station's beam spot. Given the concurrent movement of both the satellite and the station, the smaller beam spot of the ground station antenna complicates continuous satellite tracking. Therefore, the MMT system needs to update itself continuously, not to lose the satellite. Secondly, a challenge is introduced by the pedestal to track the EIVE satellite near the zenith. Because the azimuth angle after the zenith is 180° different than the azimuth angle immediately previous to the zenith at this moment, the MMT system will not be able to track the satellite when it is directly overhead. Subsequently, the azimuth motor must quickly move the payload almost 180°, which is unrealistic, while the satellite is crossing the zenith position. These two challenges create the need for a MMT system with a predictive model that accounts for the delays created by the second challenge and can react if the satellite is lost at any time.

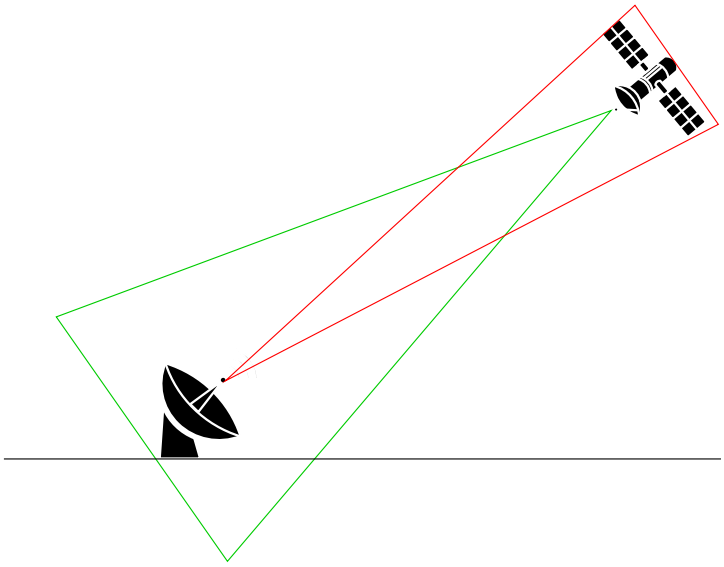


Figure 2.6.: Visualization of the beams

2.3. TLE and Trajectory Estimation

EIVE passes through the line of sight multiple times each day, with slight changes in its trajectory each time. Two of the possible EIVE orbits with elevation of 45° and 90°, the ground station, and the visible area from the antenna are shown in figure 2.7. Because of the changes in the trajectory, the ground station antenna should be adjusted each time since, in every pass, the position where it is expected to detect the satellite for the first time changes. This adjustment is done with the help of the so called two-line element (TLE). A list of an object's orbital elements for a specific period, or epoch, is encoded in the TLE data format [22]. The state (position and velocity) at any time in the past or future may be predicted using an analytical orbit propagator with the help of TLE. Furthermore, any method employing a TLE as a data source must implement one of the Simplified General Perturbations (SGP) models to accurately compute the state at a time of interest as the TLE data format is unique to the simplified perturbation models. The SGP4 model is used for LEO propagation, and it considers the main perturbation influences on a satellite: first four zonal Earth gravity field harmonics, atmospheric drag, and solar radiation pressure [22]. The open source Python library 'orekit' is used to read the TLE and then to compute the trajectories [23]. Orekit is employed worldwide, both by academics and industries, to realize space applications, studies, and operations.

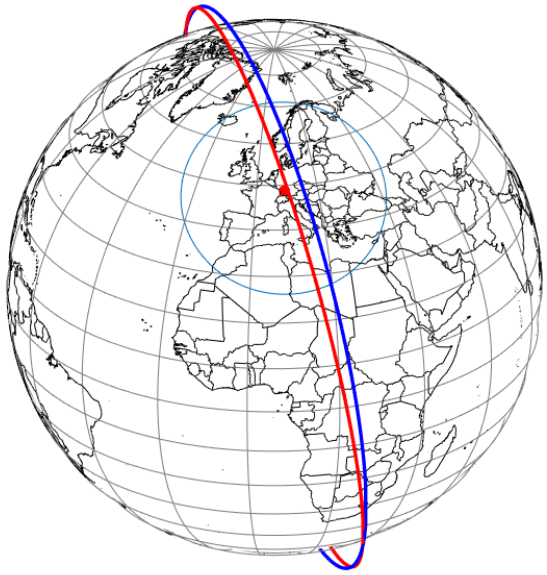


Figure 2.7.: Two possible EIVE orbits with 55 (red) and 90 (blue) degree maximum elevations, the ground station and the visible area

When the TLE data is read for orbit determination, the six orbital elements of the satellite can be seen: mean motion, eccentricity, inclination, longitude of the ascending node, argument of perigee, and mean anomaly. The orbital elements define the orbit and describe its Keplerian behavior [12]. Moreover, TLE data also provides the UTC, epoch, and revolution required for further calculations. The visual explanation of the TLE data can be seen in figure 2.8.

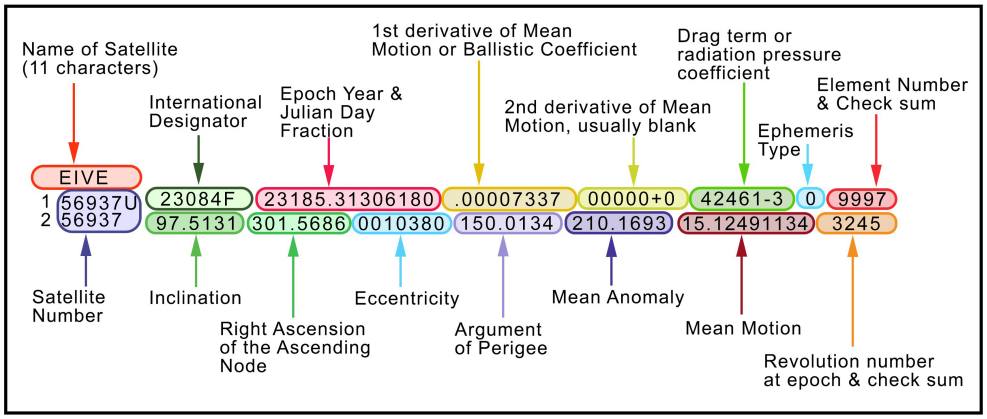


Figure 2.8.: TLE data

- Mean Motion (n): It is the angular speed needed for a body to complete one circle under the assumption of constant speed in a circular orbit that is equal in duration to the real body's variable speed in an elliptical orbit.
- Mean Anomaly (M): Despite not being a true geometric angle, it is a mathematical angle that changes linearly over time. It may be transformed into the true anomaly v, which depicts the angle between the orbiting object's position and its periapsis (closest approach to the central body) in the plane of the ellipse at any given time.
- Eccentricity (e): It describes how much the shape of the ellipse is elongated compared to a perfect circle.
- Argument of Perigee (ω): It is the angle between the ascending node and the periapsis, which defines as the ellipse's orientation in the orbital plane.
- Right Ascension of the Ascending Node (Ω): It horizontally aligns the ascending node of the ellipse with regard to the vernal point of the reference frame, which is when the orbit crosses through the reference plane from south to north.
- Inclination (i): It is the vertical tilt of the ellipse with respect to the reference plane, measured at the ascending node.

These values are then used in the SGP4 model to compute the predicted state vectors. When the predicted values are compared with the Global Navigation Satellite System (GNSS) measurements, an error of approximately (1 km) is found [22] , [24]. This error may be negligible, depending on the application and the system requirements. However, this is not acceptable for the EIVE precision and pointing accuracy requirements. To undermine this pointing error, the provided TLE data should be up to date. Despite that, TLE data is still used in the implementation of the algorithm to predict the trajectory and make the necessary adjustment to the antenna until the MMT takes over. Without the estimation of the trajectory, detection and localization of the satellite are not possible since MMT requires a very high accuracy pointing in the first place and a continuous connection with the satellite.

2.4. Kalman Filter

In this chapter, only a brief explanation of the KF is given, and it is discussed how the KF is used in tracking applications. The KF is a model based prediction method that can be described as a predict and correct loop. There are two inputs to the model:

- Measured Values: These inputs are the real measurements, such as the detected satellite position or the power variations in the power signal.
- Predicted Values: These inputs are the predictions done with the help of the used mathematical model.

Vital parameters are defined through the integration of both predicted and measured values, differentiated by the methodologies of their obtainment method. One set of values originates from real-time measurements captured through sensors or the antenna, while the other set is derived through predictive calculations using a mathematical model. Estimation of the system state parameters can be computed using these two inputs since there is no sole reliance on either measurements or predictions. Moreover, the alignment of the measured and predicted values through mathematical analysis serves to minimize error in the estimated state, considering sufficient data is available for both measurements and predictions. KF in satellite tracking applications is used explicitly for the reason that it allows projection of future movements, such as the future satellite position, and a simultaneous error minimization of the estimated state parameter. As a result, the satellite can be tracked even if the communication between the antenna and the satellite is interrupted. The low-level description of the Kalman filter is given in figure 2.9.

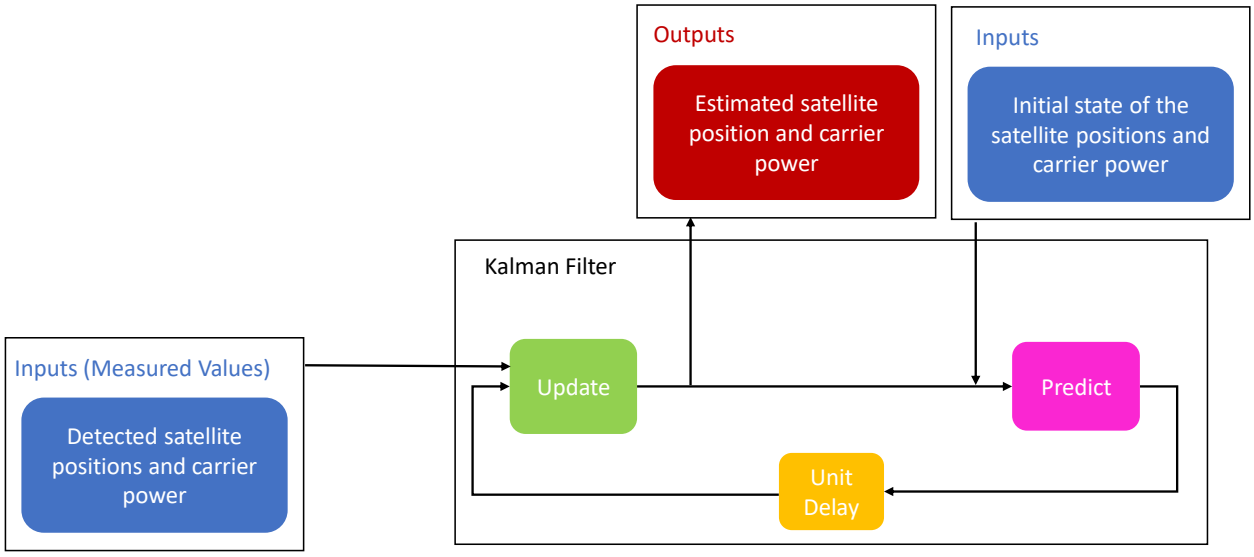


Figure 2.9.: Low level description of the Kalman filter

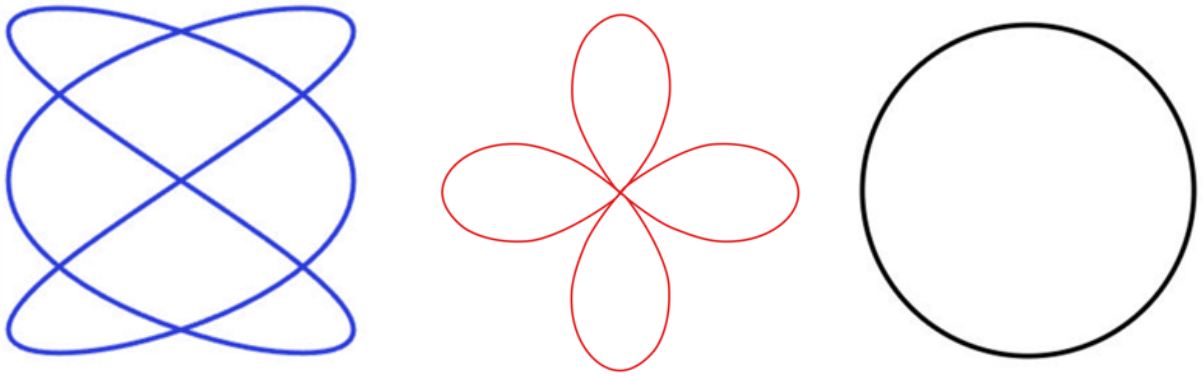
The predictive model used in the KF can be defined in a linear or nonlinear form, depending on the application. Tracking the satellite in its trajectory is a nonlinear problem. However, it is also possible to linearize the problem. Firstly, the system can be modeled as a direct nonlinear system, and an approximation method, such as the MCMC, can be used [19]. Nevertheless, this is proven to be not effective and costly, as explained in chapter 2.2.1.3. Secondly, linearization of the system can be performed. The linearization method with the standard KF is explained in chapter 2.2.1.2, which proves to be much more accurate and effective than the LMS linearization method in LEO applications, as explained in chapter 2.2.1.1. Additionally, it is also possible to use an extended Kalman filter (EKF) instead of standard KF, as shown in the work of Coelho Et al. [25]. The EKF is a recursive process with the ability to linearize the nonlinear model by using first-order Taylor series expansion, meaning that it has the ability to linearize around the current mean and covariance. It assumes that a local linearization is enough to cover the nonlinearity. This assumption leads the EKF to several limitations, such as poor convergence, erratic behaviors, or even inadequate linearization when applied to highly nonlinear systems. Moreover, to implement the KF in Python, the 'filterpy' library is used [26]. Filterpy library is a free library to implement any type of KF, which is documented in detail and updated frequently.

3. Search Algorithm and its Implementation

In this chapter, the implementation of the used scanning algorithm is discussed and explained. Firstly, the scanning patterns and the scanning area are explained in section 3.1. Secondly, advantages and disadvantages of different rotation matrices are discussed in section 3.2. Lastly, the implementation of the search algorithm is explained in section 3.5.

3.1. Scanning Patterns and Scanning Area

The search algorithm is required to detect and then localize the satellite each time it passes through the line of sight of the antenna. Considering each pass is approximately 8 to 15 minutes and the data transfer period should be maximized, the satellite should be located as fast as possible. The narrow beamwidth of the ground station antenna also creates an additional challenge since it translates into a small scanning area. As a result, the search algorithm should perform fast and increase the scanning area in the sky. There are different possible methods to increase the scanning in the sky. Three well known and used scanning patterns are the Lissajous, rosette, and conical movements pattern for the antenna. All these patterns use harmonic rotations of the antenna's x- and y-axis orthogonal to the boresight. The rotation in the x-axis is defined as the azimuth, and rotation in the the y-axis is defined as the elevation. The Lissajous, rosette, and conical scanning patterns from the perspective of the boresight are shown in figures 3.1a, 3.1b and 3.1c.



(a) Lissajous pattern **(b)** Rosette pattern **(c)** Conical pattern

Figure 3.1.: Different scanning patterns shown from the perspective of the boresight

Rotation	Conical [mdeg]	Rosette [mdeg]	Lissajous [mdeg]
Elevation	0.089	0.080	0.083
Azimuth	0.089	0.088	0.080

Table 3.1.: Satellite position error with different scanning patterns [4]

The algorithm performs one complete pattern even if any signal is detected before it readjusts itself to the next position. It is trivial that the Lissajous and rosette patterns take significantly more time than the conical pattern. The quickness of the conical pattern comes from its simplistic form. The antenna only needs to move in a circular motion rather than a much more complicated motion like the rosette and Lissajous patterns. Additionally, the disturbances in the movement of the antenna should also be considered. The wind, unbalanced weight on the antenna or any problem with the motor might cause an error in the satellite detection. In the work of Gawronski and Craparo [4], these three patterns are simulated and tested with random and harmonic disturbances, and the errors in the detected satellite positions are analyzed. The summary of the results is given in table 3.1. The results show that the error of the detected satellite positions did not change significantly depending on the used pattern. That proves that the scanned area between the defined limits is sufficient, and the satellite was detected in all cases. Consequently, the conical scanning pattern is used for the implementation of the search algorithm.

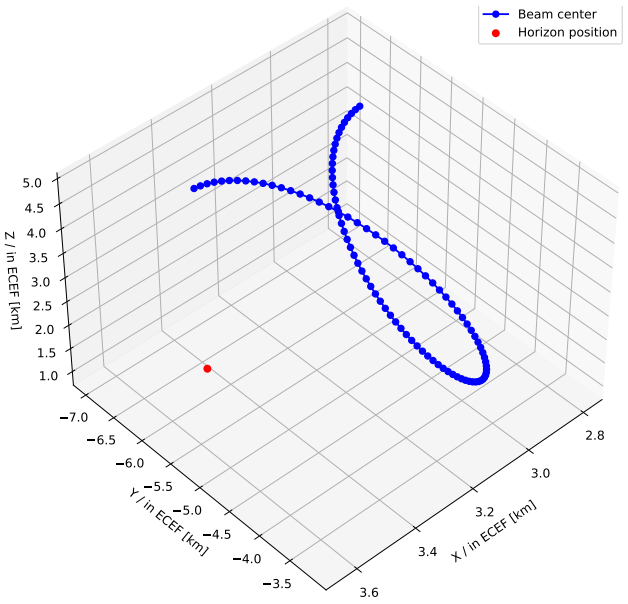


Figure 3.2.: Behavior of the antenna beam during CONSCAN

The behavior of the antenna's beam during the conical search is demonstrated in figure 3.2. The blue line represents the movement of the antenna beam's center in the sky. The red point is the so called horizon position. It is defined as the position in the sky where we expect to see the satellite for the first time. As expected, the beam moves around the horizon position in a conical pattern, searching for the satellite. If the same motion is observed from the boresight of the antenna, a

circular shape with a fixed radius will be observed, as in figure 3.3.

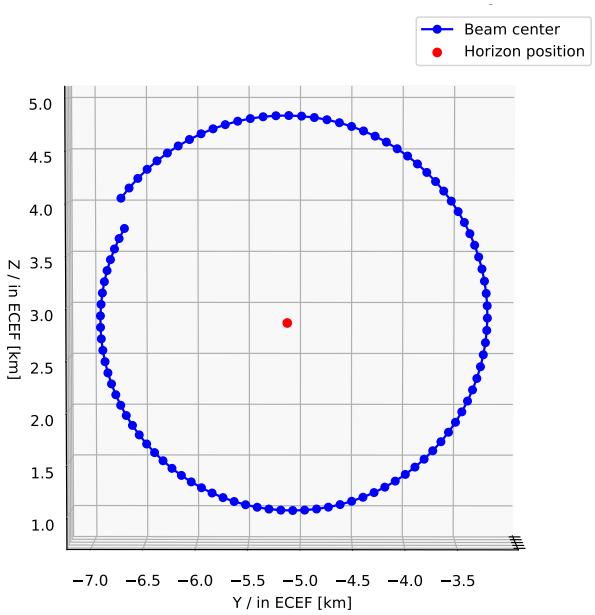


Figure 3.3.: Behavior of the antenna beam during CONSCAN seen from boresight

The visualization of the scanned area from the boresight perspective is shown in figure 3.4. The black point at the center represents the estimated position of the satellite. The four circles demonstrate the area covered by the beam during the conical search. The distance between the black point (possible satellite position) and the most outer side of beam's scanning area is defined as the radius of the conical scanning movement or $r_{conscan}$. If the radius is set to a very high value, the beam's area won't cover the entire search area around the estimated position of the satellite. That causes blank spaces in the search area. If the radius is set to a very low value, the search area might be too small, and the satellite might not be detected. Therefore, the conical scanning radius is a critical parameter that plays a key role in satellite detection.

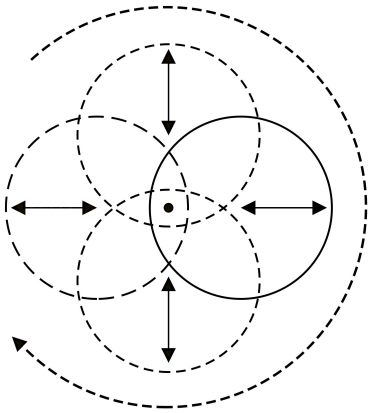


Figure 3.4.: Visualization of the scanned area

The scanning area of the antenna's beam is dependent on the slant range, HPBW and the radiation

pattern of the antenna.

- Slant Range: It is the distance between the antenna and tracked object when the tracked object is in the line of sight [27]. It is commonly given in kilometers. The slant range changes depending on the position of the satellite in the line of sight and the satellite's current trajectory.
- Half Power Beamwidht (HPBW)): HPBW is the angular separation in which the magnitude of the radiation pattern decreases by 50% (or −3 dB) from the peak of the main beam [27]. HPBW of the ground station antenna is 0.23°.
- Radiation Pattern: The graphical depiction of an antenna's radiation characteristics as a function of space can be described as an antenna pattern or radiation pattern. In other words, the antenna's pattern reveals how it radiates energy into space [27]. A circular radiation pattern is assumed for the ground station antenna.

The slant range or r_{slant} can be calculated using the law of cosine, as shown in equation 3.1 where r_a is the distance between the earth's center and the antenna position, r_s is the distance between the earth's center and satellite's position, and α is the angle from the observer to the target or the relative elevation in radians.

$$r_{slant} = \sqrt{r_s^2 + r_a^2 - 2 r_s r_a \cos(\alpha)} \tag{3.1}$$

The change in r_{slant} is demonstrated for a trajectory with a maximum elevation of 55° in figure 3.5. It can be seen from the plot that the r_{slant} decreases as the elevation increases when the satellite is in the line of sight. Therefore, the active trajectory of the satellite indirectly effects the scanning area of the antenna.

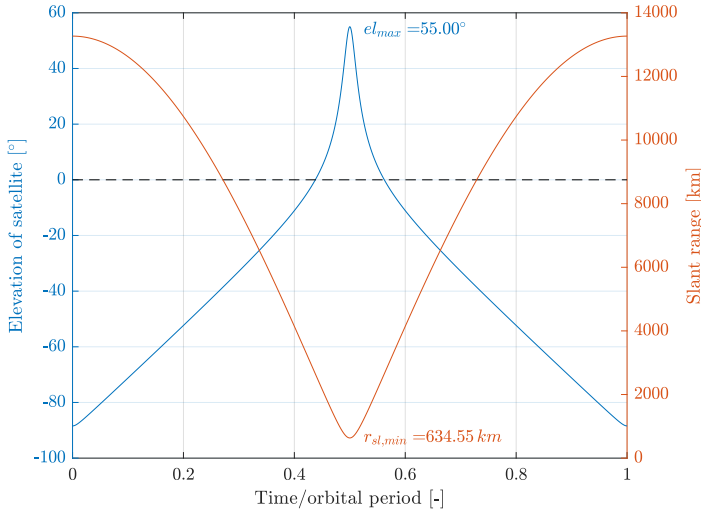


Figure 3.5.: Slant range relative to satellite's elevation

If the slant range and the HPBW are known, the radius antenna beam area r_{beam} can be calculated as given in equation 3.2.

$$r_{beam} = r_{slant} \cdot \tan(\frac{HPBW}{2}) \tag{3.2}$$

Because of the circular radiation pattern assumption, the antenna's scanning area or the beam area A_{beam} can be calculated as given in equation 3.3.

$$A_{beam} = \pi r_{beam}^2 \tag{3.3}$$

The A_{beam} is proportional to the r_{slant}. The behavior of A_{beam} against the r_{slant} with the same satellite trajectory (maximum elevation of 55°) is shown in figure 3.6. In the case of the EIVE, this means that the beam scanning area decreases when the satellite gets close to its highest possible elevation and increases when the satellite is at lower elevations. This behavior is useful for the search algorithm since the beam's scanning area is larger in lower elevations. The search for the satellite starts first from the horizon position (Elevation = 0°) and continues until the satellite is detected. As a result, the search starts with a higher scanning area, and the scanning area decreases with increasing elevation.

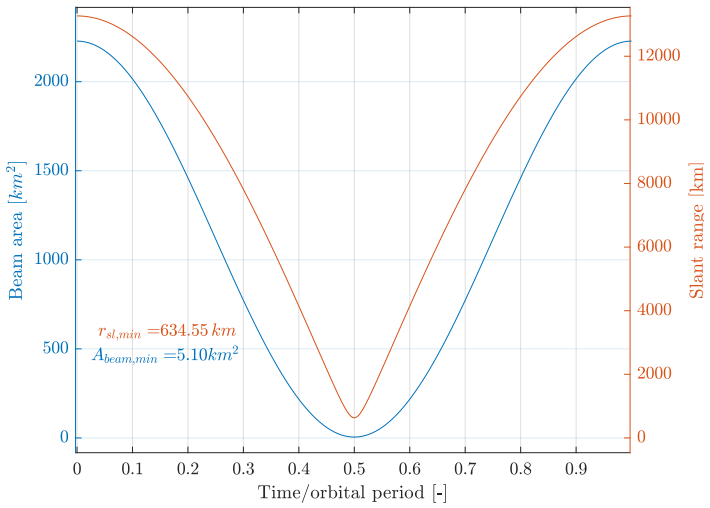


Figure 3.6.: Beam's scanning area relative to slant range

During the search, the antenna moves in a circular motion. This movement translates to the conical shape in the sky, as demonstrated in figures 3.2 and 3.3. Subsequently, the beam's circular scanning area also moves in a circle from the boresight perspective. The scanned area during the conical scanning movement is defined as the parameter $A_{conscan}$, and it can be calculated as the equation 3.4 where ω is the angular velocity of the antenna.

$$A_{conscan} = A_{beam} \cdot \omega \tag{3.4}$$

The ω defines the movement speed of the antenna. The antenna should move fast enough to perform the conical pattern in the sky and simultaneously catch up to the speed of the satellite. Therefore, the angular velocity of the satellite in ECEF can be defined as the minimum limit to the antenna's angular velocity. The satellite moves with a speed of 7 km/s. The angular velocity of the satellite ω_s in ECEF , so the lowest limit for the antenna's angular velocity $\omega_{a,min}$ can be calculated as in 3.5.

$$\omega_{a,min} = \omega_s = \frac{||r \times v||}{||r||} = 1.102 \, \frac{mdeg}{s} \tag{3.5}$$

The r and v vectors are the position and velocity vectors of the satellite's state vector, as explained in section 2.3. These two vectors define the satellite's trajectory and are read from TLE data using the Orekit library. It is crucial to point out that the changes in the r and v can be neglected. Therefore, the satellite's trajectory is assumed to have a perfect circular shape to simplify the calculation. Besides that, the $A_{conscan}$ can also be calculated using geometry and the $r_{conscan}$, as in equation 3.6.

$$A_{conscan} = \pi \cdot r_{conscan}^2 \tag{3.6}$$

The visualization of the used geometry is shown in figure 3.7. The darker blue area is the A_{beam}, and the lighter blue area is the $A_{conscan}$.

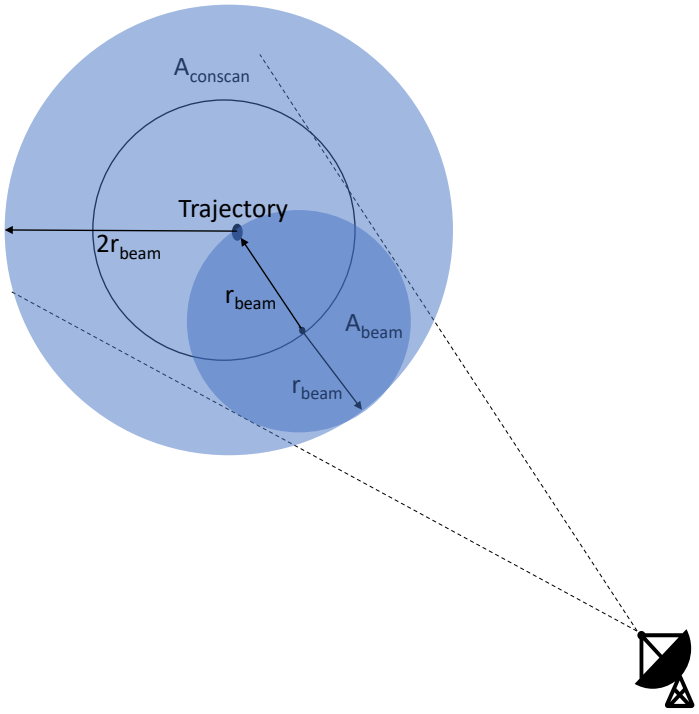


Figure 3.7.: Visualization of the scanning area of the conscan

If $r_{conscan}$ is equal to $2r_{beam}$, as in figure 3.7, then the maximum conscan scanning area is reached without any blank area in the conical shape. If $r_{conscan}$ is greater than $2r_{beam}$, then there will be blank spaces (unscanned area) near the point representing the trajectory. If $r_{conscan}$ is smaller than $2r_{beam}$, $A_{conscan}$ will be smaller, but there will be no blank spaces. Moreover, the $A_{conscan}$ will decrease with the increasing elevation of the satellite since r_{beam} will also decrease. Thus, $r_{conscan}$ should be dynamically decreased with the increasing elevation to avoid blank spaces in the scanning area during the conical search. The behavior of $A_{conscan}$ relative to the $r_{conscan}$ is given in table 3.2.

Condition	Blank space in $A_{conscan}$	$A_{conscan}$
$r_{conscan} > 2r_{beam}$	yes	$A_{conscan} = \omega \cdot A_{beam}$
$r_{conscan} = 2r_{beam}$	no	$A_{conscan} = \pi \cdot r_{conscan}^2$
$r_{conscan} < 2r_{beam}$	no	$A_{conscan} < \pi \cdot r_{conscan}^2$

Table 3.2.: Behavior of the scanning area relative to the $r_{conscan}$

In the employed search algorithm, the r_{slant} is computed using the Orekit library. It is possible to calculate the r_{slant} for each point(state vector) in the propagated trajectory of the satellite. Therefore, it is possible to calculate the $r_{conscan}$ for each point on the estimated trajectory, where the algorithm performs the conical search. Subsequently, the condition of $r_{conscan} = 2r_{beam}$ can be held onto for each performed conical pattern, and any blank space in $A_{conscan}$ can be prevented. However, it is also possible to set a constant $r_{conscan}$ and allow blank spaces or a smaller $A_{conscan}$ depending on the application.

If $r_{conscan}$ is larger than the $2r_{beam}$, the radius of the created blank space r_{blank} can be calculated, as given in equation 3.7. The o_{beam} and the $o_{conscan}$ represents the ECEF coordinates of the center position of the antenna beam and the expected position of the satellite, respectively. The Euclidean distance between these two points is calculated, and the r_{beam} is subtracted from the distance to find the r_{blank}.

$$r_{blank} = \sqrt{o_{beam}^2 - o_{conscan}^2} - r_{beam} \tag{3.7}$$

If the r_{blank} is known, the area of the blank space A_{blank} can be calculated, as given in equation 3.8. The blank area is assumed to have a circular shape since the A_{beam} and the $A_{conscan}$ are circular as well.

$$A_{blank} = \pi \cdot r_{blank}^2 \tag{3.8}$$

In some test during the experimental phase, a deliberate choice is made to establish the value of $r_{conscan}$ as 100 times its original value. This specific manipulation is undertaken to ensure a finer level of precision in observing the resultant behaviors exhibited by the conical patterns generated during the experiments. The underlying reason is that the antenna control system is only sensitive enough to capture changes in the movement of the antenna to three decimal places since the ADC employed in the control system has a sensitivity of 1 m deg/s.

3.2. Rotation Matrices

The efficacy of the employed 3D rotation matrix profoundly shapes the performance of the search algorithm. The elevation in the CONSCAN is characterized as a rotation relative to a horizontal axis orthogonal to the boresight, and the azimuth is characterized as a rotation relative to a vertical axis orthogonal to the boresight [15], as explained in chapter 2.2.1. Thus, when the search algorithm is executed, two distinct 3D rotations come into play. These rotations describe the antenna's circular movement and are used to compute the direction of the antenna. Subsequently, the beam center's position (in ECEF coordinates) is computed using the norm of the antenna's pointing at that specific time. These beam center positions draw the conical trajectory of the beam across the sky.

There are many different 3D rotations that might be utilized for this purpose. Therefore, four conditions are defined to compare the rotation matrices.

- Performance: The performance of a 3D rotation matrix can be defined as the number of mathematical calculations (computations) needed to perform n number of rotations to a vector. The number of computations directly affects the used memory and, thus, the fastness of the algorithm. Comparison of the performance is done using the performance metrics shown in the work of Soohwan and Minkyoung [5].
- Singularity: Singularity of a rotation matrix describes a point where the matrix has no inverse, and therefore cannot be used to uniquely determine the orientation of a rigid body. In special cases of Euler angles, one degree of freedom is lost, and the system is locked in a degenerate two-dimensional space [5]. It cannot be avoided by limiting ranges and the iteration does not converge near singularity. The Gimbal lock is a good example of this phenomenon.
- Complexity: This condition defines how complex it is to implement these rotation matrices using Python and preexisting libraries.
- Accuracy: This condition defines how well and accurate the used rotation matrix behaves when the algorithm is performed relative to the other options.

Considering the complexity condition, only three rotation matrices are tested in the search algorithm, which are the Euler rotation, Rodrigues rotation, and quaternion rotation. The Euler and Rodrigues rotations are implemented with the 'Rotations' class using the open source Python library SciPy [28], whereas the quaternion rotation is implemented using the open source Python library quaternion which extends the NumPy library [29].

3.2.1. Euler Angles and Rotation

The Euler angles (α, β, γ) describe the orientation of a rigid body relative to a fixed coordinate system [5]. If the fixed coordinate system is defined as a traditional x, y, and z coordinate system, any orientation of this rigid body can be reached utilizing the rotation matrices defined with Euler angles for each axis (x, y, z). The reference coordinate system with Euler rotation angles is shown in figure 3.8.

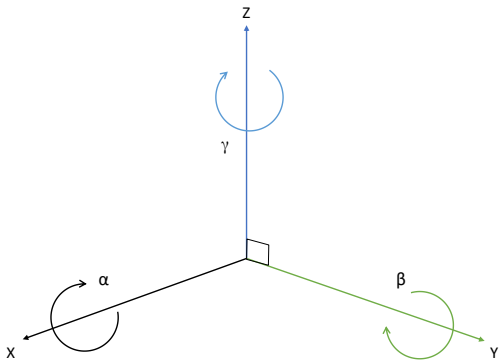


Figure 3.8.: The reference coordinate system with Euler rotation angles

The rotation of the x-axis by angle α is given in equation 3.9.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \tag{3.9}$$

Similarly, the rotation of the y-axis by angle β is given in equation 3.10.

$$R_y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{3.10}$$

Lastly, the rotation of the z-axis by angle γ is given in equation 3.11.

$$R_z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

Euler rotation is defined as a rotation with respect to a moving frame [30]. Therefore, the order of the axis rotations is critical in defining the 3D rotation. For a rotation where first the x-axis, then the y-axis, and lastly the z-axis is rotated, the 3D Euler rotation is given in equations 3.12 and 3.13.

$$R_{euler} = R_x(\alpha)R_y(\beta)R_z(\gamma) \tag{3.12}$$

$$R_{euler} = \begin{bmatrix} \cos\beta\cos\gamma & -\cos\beta\sin\gamma & \sin\beta \\ \cos\alpha\sin\gamma + \sin\alpha\sin\beta\cos\gamma & \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & -\sin\alpha\cos\beta \\ \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma & \sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma & \cos\alpha\cos\beta \end{bmatrix} \tag{3.13}$$

Moreover, even though Euler rotation is widely used, it suffers from the singularity problem known as the Gimbal lock. During the Gimbal lock, two rotation axes become parallel, and one

degree of freedom is lost [5]. The loss of degree of freedom locks the system into a degenerate two-dimensional space and causes unwanted behavior.

3.2.2. Rodrigues Rotation

Rodrigues' rotation formula is an algorithm for rotating a vector around a given axis with a fixed angle of rotation in space [31] [5]. If the Rodrigues' rotation formula is applied to all the three basis vectors, as defined in the 3D rotation group in Euclidean space [30], a rotation matrix can be created without computing the full matrix exponential. Not computing the full matrix exponential reduces the complexity of the implementation. Nevertheless, Rodrigues rotation has a singularity at 180° that needs to be avoided so that the rotation stays stable. The Rodrigues' formula is given in equation 3.14 for a single vector. A vector p is rotated about an axis $\hat{v}$ by an angle θ, and transformed to the vector P. Additionally, $\hat{v}$ is normalized and is a unit vector of one since only the direction is needed and not the magnitude.

$$P = p\cos\theta + (\hat{v}\times p)\sin\theta + (\hat{v}\cdot p)\hat{v}(1-\cos\theta) \tag{3.14}$$

If P is redefined as given in equation 3.15, a three dimensional rotation is reached where R is the rotation matrix.

$$P = Rp \in \mathbb{R}^3 \tag{3.15}$$

$R \in \mathbb{R}^{3x3}$ is given in equation 3.16, where I is the identity matrix, and $[.]_x \in \mathbb{R}^{3x3}$ denotes a skew-symmetric matrix which turns a cross product between two vectors into a matrix vector multiplication [5].

$$R = I + [\hat{v}]_x \sin\theta + [\hat{v}]_x^2(1-\cos\theta) \tag{3.16}$$

3.2.3. Quaternion Rotation

Unit quaternions, often referred to as versors, offer a concise mathematical framework to denote spatial orientations and rotations within three-dimensional space. Precisely, these quaternions encapsulate details of an axis-angle rotation, enabling representations of arbitrary rotational transformations. Computer graphics, computer vision, robotics, navigation, molecular dynamics, flight dynamics, and satellite orbital mechanics analysis benefits from the utility of rotation and orientation quaternions [31],[30]. In this chapter, only a brief explanation of the quaternions is given since this subject is highly complex and has many details. A quaternion $q \in \mathbb{H}$ is described as a summation of both imaginary and real numbers, as given in equation 3.17:

$$q = w + ix + jy + kz, \tag{3.17}$$

where the $x, y, z, w \in \mathbb{R}$. If $y = z = 0$, then q can be defined as a complex number. Therefore, the set of quaternions $\mathbb{H}$ includes the set of complex numbers $\mathbb{C}$. Moreover, a pure quaternion is defined as given in 3.18:

$$q = ix + jy + kz, \tag{3.18}$$

where $w = 0$. In this case, the pure quaternion can be interpreted as a 3D vector in space that allows the computation of rotations [5]. If a rotation axis is defined as unit vector $\hat{v} = [v_x, v_y, v_y]^T$, the corresponding unit quaternion for this rotation axis is given in 3.19:

$$\hat{v} = iv_x + jv_y + kv_z. \tag{3.19}$$

If the rotation of this axis $\hat{v}$ is described with an angle θ, it can be defined as a unit quaternion, as given in 3.20:

$$\hat{q} = \cos(\frac{\theta}{2}) + \hat{v}\sin(\frac{\theta}{2}). \tag{3.20}$$

Subsequently, the rotated vector p_{rot} can be calculated, as given in equation 3.21:

$$p_{rot} = \hat{q}p\hat{q}^* \tag{3.21}$$

, where p is a vector defined as a pure quaternion, as in equation 3.18. This principle of rotation is utilized for the 3D rotation. If a unit quaternion is defined as $\hat{q} = q_w + iq_x + jq_y + kq_z$, the 3D rotation matrix $R_{quaternion} \in \mathbb{R}^{3x3}$ can be computed as given in equation 3.22:

$$R_{quaternion} = \begin{bmatrix} 1-2(q_y^2+q_z^2) & 2(q_xq_y - q_wq_y) & 2(q_xq_z + q_wq_y) \\ 2(q_yq_x + q_wq_z) & 1-2(q_x^2+q_z^2) & 2(q_yq_z - q_wq_x) \\ 2(q_zq_x - q_wq_y) & 2(q_zq_y + q_wq_x) & 1-2(q_x^2+q_y^2) \end{bmatrix}. \tag{3.22}$$

In other words, equation 3.22 denotes the conversion from a unit quaternion to a rotation matrix. Besides that, the quaternions do not have any singularity. They are numerically stable and immune to problems such as the Gimbal lock.

3.2.4. Comparison of the Rotation Matrices

The Euler, Rodrigues, and quaternion rotations are compared in this chapter considering the four conditions defined in chapter 3.2. Firstly, the performance condition is analyzed. In the work of Soohwan and Minkyoung [5], the three methods are compared in respect to computations required to perform the n number of rotations to a vector. The results are summarized in table 3.3, where the performance is explained as the summation of multiplication/division operations (MD), addition/subtraction operation (AS), and number of math library function calls (MLFC). The

comparative analysis reveals that the Rodrigues rotation exhibits the least favorable performance, whereas the Euler and quaternion rotations demonstrate highly congruent levels of performance. Consequently, it is argued that Euler and unit quaternions needs less memory usage and less computation time in contrast to the Rodrigues rotation.

Rotation Method	MD	AS	MLFC	Total Operations
Euler	$9n+13$	$6n+4$	6	$15n+23$
Rodrigues	$9n+12$	$6n+13$	2	$15n+27$
Unit Quaternion	$9n+12$	$6n+12$	0	$15n+24$

Table 3.3.: Comparison of the rotation matrices for n number of rotations to a vector [5]

The singularity condition establishes the numerical stability characteristics of the applied rotation. As explained in chapters 3.2.1, 3.2.2 and 3.2.3, only the quaternion rotation is immune to singularity caused problems, including the Gimbal lock phenomenon. In contrast, the Rodrigues and Euler rotations can exhibit singularity problems, potentially inducing unanticipated behaviors if the singularity conditions are met.

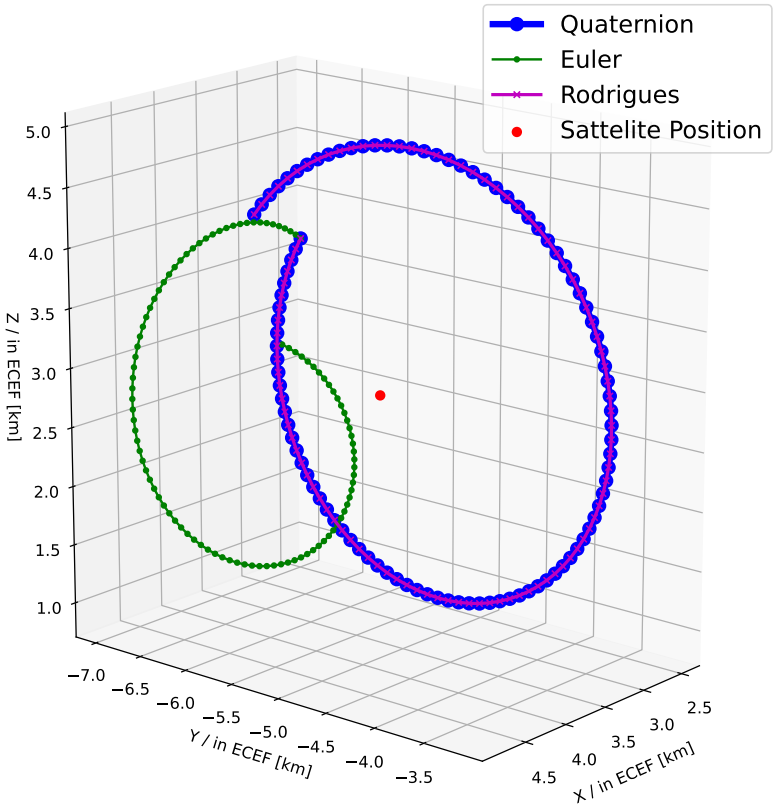


Figure 3.9.: Comparison of the accuracy condition of the rotation matrices in ECEF

The complexity condition is a rather subjective condition that characterizes the rotation matrices on their implementation hardship using preexisting libraries. Within this context, the quaternion

rotation manifests as the most straightforward to implement, succeeded by the Rodrigues rotation. Conversely, the Euler rotation emerges as the most intricate to implement.

The accuracy condition characterizes how accurately the rotation matrices behave when the search algorithm is conducted. The figure 3.9 visualizes the computed rotations for the antenna beam's center position in ECEF for Euler (green), Rodrigues (magenta), and quaternion (blue) rotations around the satellite's expected position for an exemplary trajectory. The graphical representation highlights that Rodrigues and quaternion rotations exhibit the anticipated conical motion centered around the expected satellite position. In contrast, the Euler rotation deviates unexpectedly. It not only fails to achieve a conical motion but also presents an incorrect perspective, implying a potential discrepancy in the rotation. The unexpected shape of the conducted Euler rotation may arise as a consequence of numerical instability attributable to singularity. Consequently, the potential numerical stability challenges associated with the Euler rotation render it an unfavorable choice for integration within the search algorithm. That is particularly relevant, considering the algorithm's step count increases proportionally with the number of circles performed, which causes more possible singularity points when performing the Euler rotation.

In figure 3.10, the behavior of the quaternion and Rodrigues rotations in HCS is shown for an exemplary trajectory. As can be seen, there is no numerical difference between the quaternion and Rodrigues rotation matrices. They both provide the same numerical and behavioral results. Therefore, it can be concluded that both Rodrigues and quaternion rotations have the same accuracy.

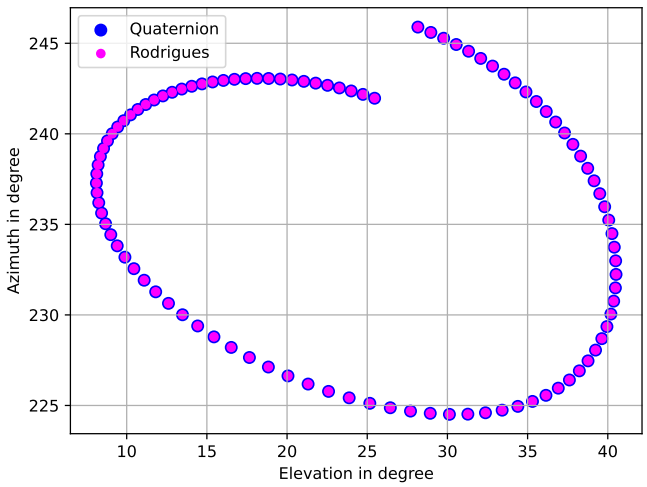


Figure 3.10.: Comparison of the accuracy condition of the quaternion and Rodrigues rotations in HCS

Upon comprehensive comparison and analysis of all conditions, two distinct factors emerge as paramount. Foremost is the presence of singularities, wielding a direct influence over the stability of the executed rotation. Subsequently, accuracy emerges as the ensuing critical factor, given that the algorithm's efficacy is compromised in the absence of precise rotations. In light of these

considerations, quaternion rotation inherently stands out as the compelling preference with low complexity, high performance, high accuracy, and no singularity. Thus, it logically follows that the performed rotations in the search algorithm are based on the quaternion rotations.

3.3. Characterization of the Conical Pattern

The nature of the conical pattern that unfolds is shaped by three principal parameters, with the influence of the applied rotation matrix set aside since only quaternion rotation is employed in the algorithm. These parameters are the number of steps in the conical pattern (n_{step}), the angular velocity (ω), and the radius of the conical shape $r_{conscan}$. The impact of the $r_{conscan}$ is already elaborated in chapter 3.1. Therefore, only the effects of n and ω are analyzed in this chapter.

The parameter n_{step} plays a pivotal role in determining the distribution of rotated points around the anticipated satellite location. In other words, it defines the number of steps executed to compute the positions of the beam centers which form the conical shape. It is important to note that n_{step} doesn't have a direct influence on the shape of the conical search pattern itself. It only determines how smooth the conical pattern is or if it has more edges. However, it dictates the time required to execute a single conical pattern with the antenna. The azimuth and elevation values of the antenna beam center positions are the input of the antenna control system. Subsequently, n_{step} times beam center positions in ECEF are needed to be converted into HCS. As the antenna initiates its motion, it cycles through these n_{step} points, with each cycle demanding at least 1 s per iteration. Consequently, the total time required to complete a full circle corresponds to the value of n.

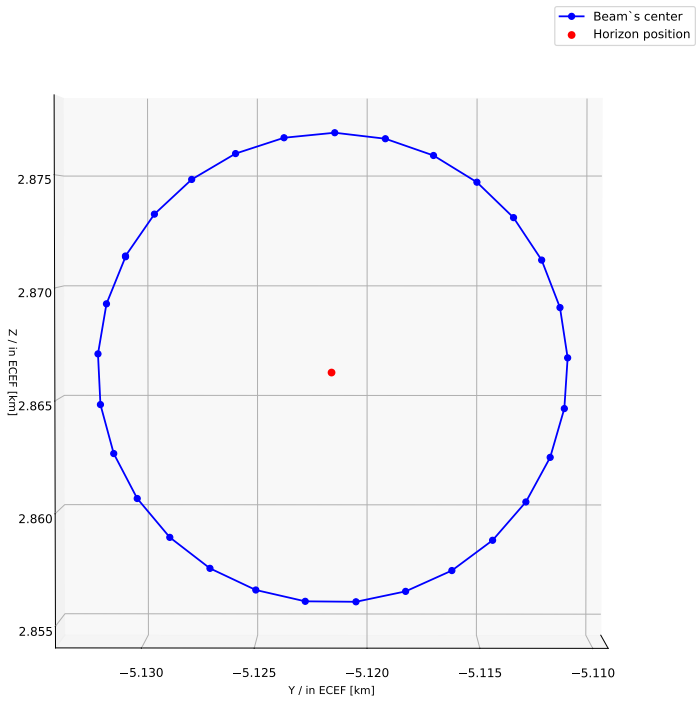


Figure 3.11.: Shape of the conical shape with ω equal to 0.001 102 m deg/sec

Theoretically, the step time of 1 s can be decreased to create a more precise circular shape, with less time between each step. However, this is not feasible since the antenna is not capable of accelerating and decelerating in such a short time.

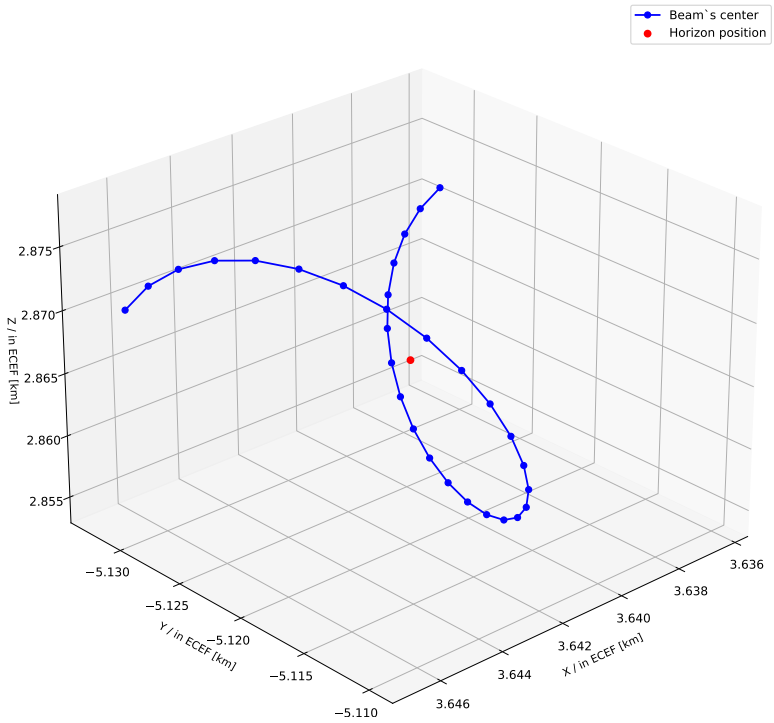


Figure 3.12.: Shape of the conical shape with ω equal to 15 m deg/sec

The parameter ω wields a direct influence over the configuration of the executed conical pattern. It precisely dictates the extent to which the pattern embodies a conical shape. Smaller ω values yield patterns that lean towards a circular form, while higher values lead to a more pronounced conical appearance. Furthermore, it is crucial to emphasize that the conical pattern's presentation is also contingent upon the satellite's position relative to the antenna. As the pattern takes form at different elevations and azimuth values, its structure undergoes alterations, consequently impacting the degree of influence exerted by ω. However, it's important to underline that, regardless of the elevation and azimuth, ω plays a direct role in shaping the conical pattern. The conducted conical shapes for an elevation of 25° and azimuth of 241° with ω equal to 0.001 102 m deg/sec, 15 m deg/sec are shown in figures 3.11, and 3.12. As expected, a higher value for ω corresponds to a more conical shape.

When ω is assigned to an excessively high value, it disrupts the intended conical shape, rendering the antenna incapable of executing the conical scan as desired. This outcome arises from the fundamental design of the conical scanning process, where the azimuth and elevation values are defined through sinusoidal rotations that are orthogonal to the antenna's boresight. This concept is elaborated upon in chapter 2.2.1. When ω is raised beyond a certain point, the sinusoidal behavior becomes overly stretched, which in turn causes unexpected and undesirable behaviors in the rotation mechanism. However, the exact value of ω where this issue arises depends on the

relative position of the satellite to the antenna. The conducted conical shape for an elevation of 25° and azimuth of 241° with ω equal to 100 m deg/sec is shown in figures 3.13.

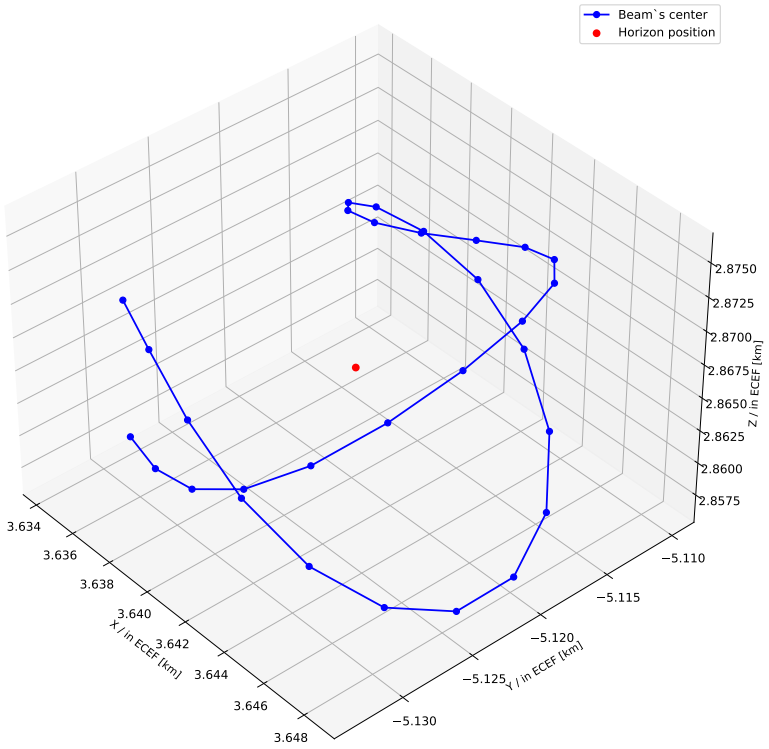


Figure 3.13.: Shape of the conical shape with ω equal to 100 m deg/sec

As can be seen in figure 3.13, the shape of the conical pattern deviates to an unwanted shape if the value for ω is chosen too high.

3.4. Synchronization of the Conical Search and the Satellite Pass

Another important aspect of the search algorithm is the time synchronization of the antenna's and the satellite's movement. The initial step to achieve the time synchronization is to compute both the satellite and the antenna movement time in the same relative time interval. The movement time of the antenna and the satellite is defined in seconds in UTC starting from 00.00 in UTC time. This computation is essential for orchestrating a harmonized alignment of their respective movements.

The temporal window during which the satellite remains within the antenna's line of sight can be determined by leveraging the trajectory propagation capabilities offered by the Orekit library. The process of trajectory propagation involves iterative computations, each executed with a predefined time increment. In the context of the search algorithm, this interval is set at 1 s. In other words, every state vector, which defines the satellite's path within the line of sight, maintains a one-second temporal gap between successive iterations. Consequently, the count of these state vectors corresponds to the overall time span required for the satellite to traverse the designated trajectory

within the antenna's line of sight.

Determining the overall time necessary for the antenna's operations involves a more intricate process. Initially, it's imperative to compute the duration required for executing a single conical pattern. As explained in chapter 3.3, this duration aligns with the number of steps in the conical pattern, given the one-second gap between each iteration. Subsequently, calculating the cumulative time for all conical patterns necessitates the multiplication of the total number of patterns and the count of steps (n_{step}). It is crucial to point out here that there are jumps from one pattern to the other during the conical searching process. These jumps occur since the conical search is performed iteratively around the trajectory. Furthermore, it is not feasible to perform the conical search for each state vector in the trajectory. Therefore, the number of positions to perform the conical search is downsampled. The downsampling is done with the downsampling factor (m). The number of positions ($n_{pos,conscan}$) to perform a conical search is calculated, as given in equation 3.23, where the n_{sv} defines the total number of state vectors.

$$n_{pos,conscan} = \frac{n_{sv}}{m} \tag{3.23}$$

Moreover, the time passed to jump from one conical pattern to the other is calculated with the angular separation method. The angular separation (θ) between the last step of the conducted conical pattern (p_{first}) to the first step of the next conical pattern (p_{next}) is calculated, as given in equation 3.24. The p_{first} and p_{next} are vectors containing the azimuth (az) and elevation (el) values for that specific step.

$$\theta = \arccos(\sin(el_{first}) \cdot \sin(el_{next}) + \cos(el_{first}) \cdot \cos(el_{next}) \cdot \cos(az_{first} - az_{next})) \tag{3.24}$$

Subsequently, the jump time (t_{jump}) between each conical pattern is calculated in seconds, as given in equation 3.25, where ω is the angular velocity of the antenna.

$$t_{jump} = \frac{\theta}{\omega} \tag{3.25}$$

Therefore, the total time passed for the antenna to perform its actions can be calculated, as given in equation 3.26.

$$t_{total,ant} = \sum_{k=1}^{n_{pos,conscan}-1} t_{jump} + n_{pos,conscan} \cdot n_{step} \tag{3.26}$$

Nevertheless, some delay can be expected since the antenna also moves in iterations. It accelerates and decelerates between each step to perform the conical pattern or the jumps, which causes a cumulative delay in the whole movement. If some environmental effects slow down the antenna movement, the cumulative delay can become too high, causing the synchronization to fail. Therefore, it is crucial to set the satellite angular speed sufficiently.

The time synchronization is implemented using the antenna's movement principles. Before conducting any conical patterns, the satellite position in time and the time passed since the antenna started moving are compared. The implementation method ensures that the antenna begins performing the conical search in each down sampled position before the satellite arrives, creating a time buffer between the satellite's arrival and the antenna's movement. The buffer can be adjusted depending on the application. The flowchart explaining how the synchronization works is shown in figure 3.14.

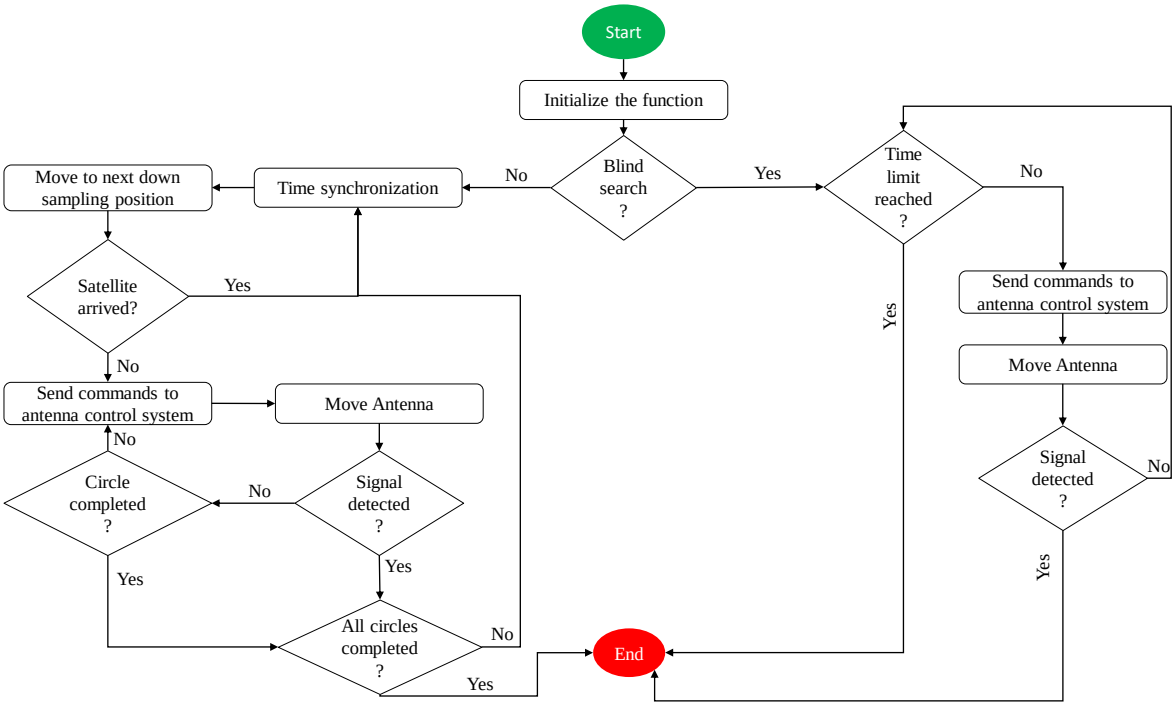


Figure 3.14.: 'Move antenna' flowchart

3.5. Search Algorithm

The search algorithm is structured into three key components. The initial segment referred to as 'Read TLE', explained in Chapter 3.5.1, entails the input of TLE data and the Coordinated Universal Time (UTC) for the projected satellite pass, achieved through a graphical user interface (GUI). Subsequently, pivotal data for subsequent algorithm steps, such as the estimation of the trajectory, angular velocity, state vectors, is extracted from the TLE. This is facilitated by the Orekit library [23] and its propagator. The ensuing phase harnesses the extracted data to initiate a blind search within the vicinity of the anticipated satellite position at the horizon, explained in chapter 3.5.2. The blind search persists until either initial contact with the satellite is established or a predefined time interval elapses. In the following phase, a conical search ensues along the projected trajectory of the satellite, aiming to detect the satellite multiple times across its orbital path, explained in chapter 3.5.3. Subsequent to the conical search's execution, the tracking system

transitions into operation, facilitated by the MMT.

3.5.1. Read TLE

Within the 'read TLE' phase of the algorithm, the requisite data is acquired from the TLE to propel subsequent algorithmic processes. As expounded upon in Chapter 2.3, Orekit [23] empowers users to establish a reference environment and engage its propagator to derive satellite or astrophysics related insights within this designated context.

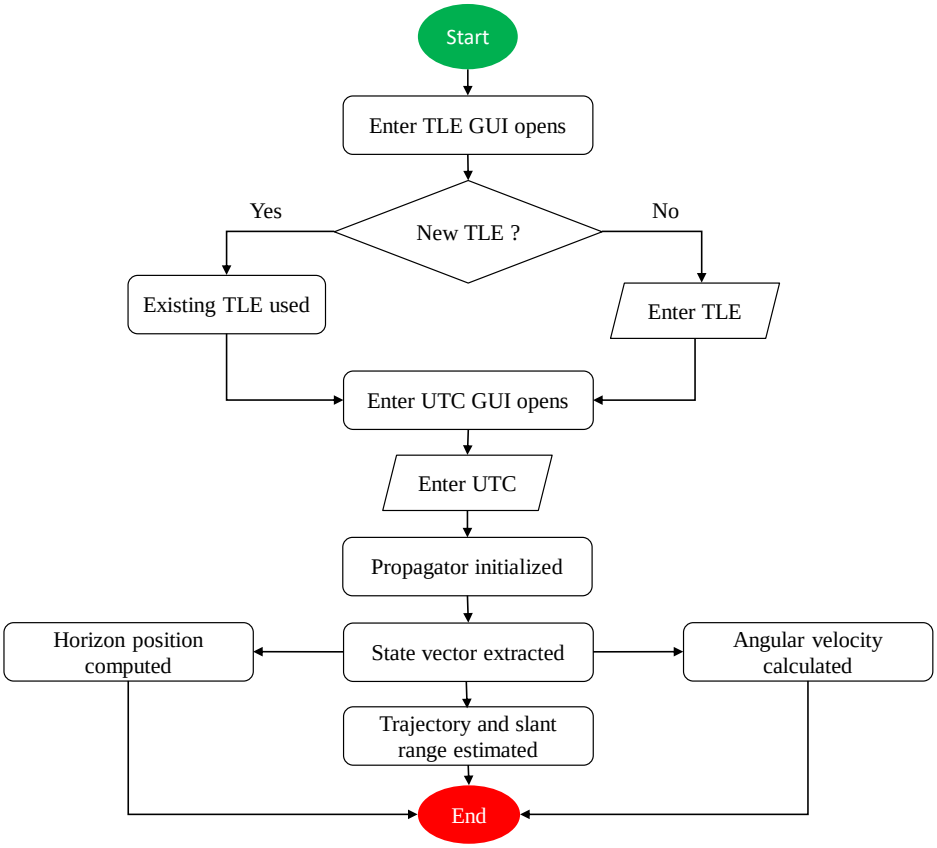


Figure 3.15.: 'Read TLE' flowchart

The flowchart outlining the intricate functioning of this phase is demonstrated in figure 3.15. In this algorithm, the Earth is modeled using the WGS84 model, and the SGP4 propagator, in combination with the Orekit library, is enlisted. Following comprehensive initialization of the environment, the state vector, encompassing both position and velocity vectors, is extracted from the TLE data for the predefined temporal span. This state vector serves as the foundation for calculating angular velocity, charting the satellite's trajectory over the designated time frame, and pinpointing the satellite's horizon position. In addition, the slant range for each position, where the conical search is conducted, is computed using the propagated trajectory. Subsequently, the beam radius, the CONSCAN radius, the beam area, and the CONSCAN area are computed for each position, as explained in chapter 3.1.

3.5.2. Blind Search

Within the blind search phase, the primary objective revolves around localizing the satellite during its initial passage near the horizon. Once the satellite's position is pinpointed or a predetermined time threshold is reached, the algorithm transitions into the subsequent phase, the conical search. The flowchart demonstrating the blind search phase of the algorithm is given in figure 3.16.

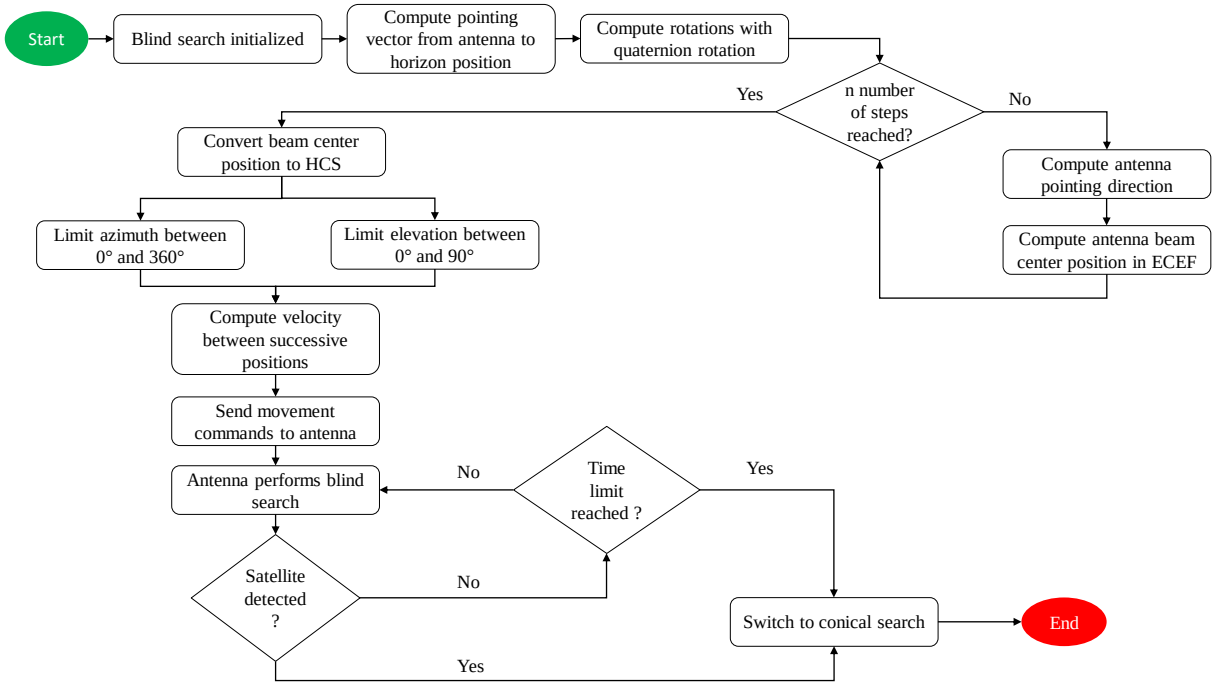


Figure 3.16.: Blind search flowchart

As seen in the in figure 3.16, the first step is to initialize the necessary variables for the blind search, which are the number of steps n, angular velocity ω, $r_{conscan}$, and time limit for the duration of the blind search. After that, the pointing vector between the antenna's position and the satellite horizon position is calculated. This pointing vector is called the horizon direction.

Upon obtaining the horizon direction, quaternion rotations are employed to derive rotation matrices for both the elevation and azimuth components of the antenna. These matrices serve as pivotal tools for the forthcoming computations. The pseudo code for the calculation of the rotations are presented in Algorithm 1. Following this, a conical pattern is created for the blind search. This pattern materializes through iterative rotations of the central point within the antenna beam around the satellite's fixed horizon position. This iterative process is executed n times, and the value of n governs the number of distinct points forming the conical pattern. In other words, n defines the number of steps in the conical pattern. It is essential to point out here that if the horizon position has a too low elevation (depending on the propagated trajectory), some of the computed beam center positions in the conical pattern may be under the horizon line (0° elevation). Consequently, that causes unwanted behaviors in the performed rotation and the movement of the antenna.

Prior to calculating the beam center positions within ECEF coordinates, it becomes essential to compute the antenna's pointing direction concerning the anticipated position of the satellite on the horizon. This pivotal step lays the groundwork for subsequent computations. The pseudo code for the calculation of the beam center positions are presented in Algorithm 2.

Algorithm 1 Compute the rotations

Compute the horizon direction:
    $horizon\_direction = horizon\_pos\_ECEF - antenna\_command\_pos^T$
    $horizon\_direction\_norm = \frac{horizon\_direction}{\|horizon\_direction\|}$
Compute the number of steps for each rotation:
    $\theta = \text{linspace}(0, 2\pi, num\_steps)$
Compute angles for elevation and cross-elevation:
    $angles\_elevation = (\theta)$
    $angles\_cross\_elevation = (\text{angular\_velocity} \cdot \theta)$
Quaternion-Based Rotations:
    $R\_azimuth = quaternion\_rotation([1,0,0], angles\_elevation)$
    $R\_elevation = quaternion\_rotation([0,1,0], angles\_cross\_elevation)$

Following the comprehensive calculation of all beam center positions, a conversion process takes place, transitioning these positions from ECEF coordinates to HCS. This conversion is imperative due to the requirements of the antenna control system, which exclusively accepts input in terms of azimuth and elevation values, or their corresponding velocities. Ultimately, the computed values in HCS are transmitted to the antenna control system.The antenna keeps performing the blind search until the satellite is detected or the predefined time limit is reached, and then the algorithm switches to the conical search part, ending the blind search.

Algorithm 2 Compute beam center positions

for each index i from 0 to $n-1$ **do**
    Compute antenna direction:
        $antenna\_direction_i = R_{azimuth[i]} \cdot R_{elevation[i]} \cdot horizon\_direction\_norm^T$
        $antenna\_direction\_norm_i = \frac{antenna\_direction_i}{\|antenna\_direction\|}$
        $antenna\_directions_i = antenna\_direction\_norm_i$
    Compute beam center position relative to the horizon position:
        $beam\_pos_i = horizon\_pos\_ECEF + (radius\_conscan \cdot antenna\_direction\_norm_i^T)$
        $beam\_positions_i = beam\_pos_i$

3.5.3. Conical Search

The conical search phase mirrors the structure of the blind search segment. The computations for the conical pattern and beam center positions are conducted, as detailed in algorithms 1 and 2. In other words, the fundamental concept remains unaltered. Nonetheless, there is a key distinction: while the blind search involves executing the conical pattern centered around a stationary position (horizon position), the conical search dynamically extends around the estimated trajectory, ensuring

a iterative search around the trajectory. The flowchart outlining functioning of the conical search is demonstrated in figure 3.17.

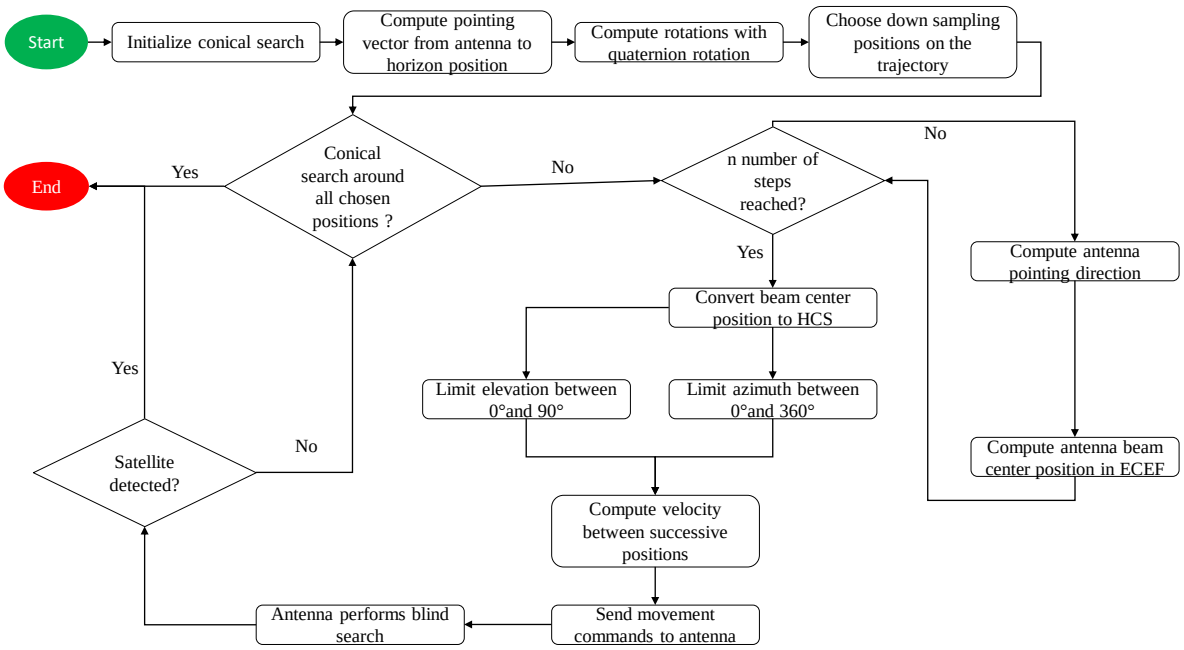


Figure 3.17.: Conical search flowchart

Within the conical search phase, the satellite's projected trajectory serves as the foundational reference. The estimation of the trajectory is propagated using the state vectors that are extracted from the TLE. Each of these state vectors corresponds to a distinct temporal point along the satellite's trajectory. However, performing a conical search for every individual state vector is not feasible due to the antenna's limited movement speed and physical capacities. Consequently, the state vectors are subjected to down sampling tailored to the desired number of circular motions to be executed. Moreover, it's important to note that each conical pattern corresponding to a temporal state vector is executed in an iterative manner. The antenna undertakes the conical pattern for one state vector before transitioning to the next, leading to jumps between iterations. The magnitude of these jumps hinges upon the down sampling parameter. A smaller number of circular motions results in longer jumps between consecutive iterations, effectively influencing the overall movement dynamics of the antenna. The consequent trade-off between the number of conducted circles and the jumps is a critical characterization of the conical search phase of the algorithm.

The slant range isn't a steady parameter; it fluctuates in relation to the elevation, as elaborated upon in Chapter 3.1. Consequently, both $r_{conscan}$ and $A_{conscan}$ undergo variations with each conical pattern. The algorithm integrates this dynamism, leading to nuanced alterations in the conical pattern shapes. An example of this phenomenon is given in figure 3.18 with $m = 3$, resulting

in three conical patterns. As anticipated, the conical patterns get smaller, and the scanning area decreases with the increasing elevation.

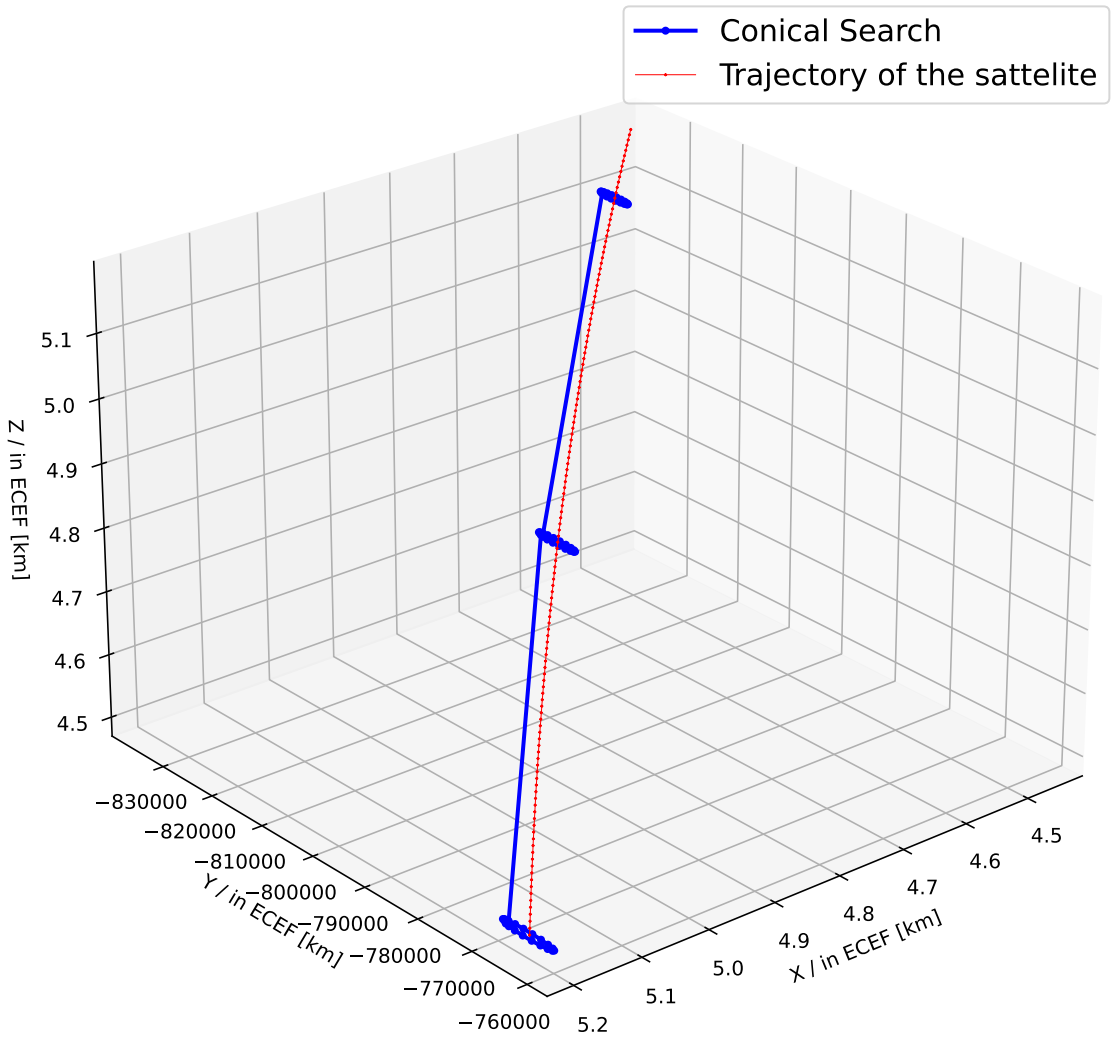


Figure 3.18.: Conical search flowchart

4. Simulations and Tests

The primary objective of these tests revolves around analyzing the dynamics of antenna movement while concurrently evaluating its agreement with the precomputed input parameters. As previously highlighted, the antenna exclusively accepts azimuth and elevation values as inputs. Consequently, the assessment pertains to the antenna's ability to execute the intended motion using the predetermined input values. The tests are conducted both for the case of the blind search and the conical search.

4.1. Blind Search

The executed algorithm should perform the blind search and conical search for any given TLE. Therefore, two different satellite passes of the EIVE are chosen to demonstrate the behavior of the blind search in different test cases. It should also be mentioned that the antenna performs the blind search for a time limit of 40 s with a maximum speed of 5 deg/sec so that the cumulative delay can be observed with consecutive circles. The first satellite pass was on 11.07.2023, starting from 23.48 UTC with a duration of 10 min. The second satellite pass was on 21.08.2023, starting from 00.09 UTC with a duration of 6 min. The respective TLEs are given below:

- Satellite pass 1 TLE: First line is '1 56937U 23084F 23185.31306180 .00007337 00000+0 42461-3 0 9997' and the second line is '2 56937 97.5131 301.5686 0010380 150.0134 210.1693 15.12491134 3245'.
- Satellite pass 2 TLE: First line is '1 56937U 23084F 23233.32797279 .00011238 00000-0 62996-3 0 9996' and the second line is '2 56937 97.5162 348.8709 0008765 357.1043 3.0136 15.13522526 10501'.

The blind search initiates at lower elevations, representing the initial phase of the search process. Consequently, our tests are carried out at these relatively lower elevations, where we anticipate the satellite's first detection. The defined test cases (TC) are given in table 4.1.

Test cases	Elevation	Satellite pass	$r_{conscan}$	ω	n_{step}
TC1	10°	pass 1	$= 2r_{beam}$	15 deg/s	20
TC2	10°	pass 2	$= 2r_{beam}$	15 deg/s	20
TC3	30°	pass 1	$= 2r_{beam}$	15 deg/s	20
TC4	30°	pass 2	$= 2r_{beam}$	15 deg/s	20

Table 4.1.: Blind search test cases

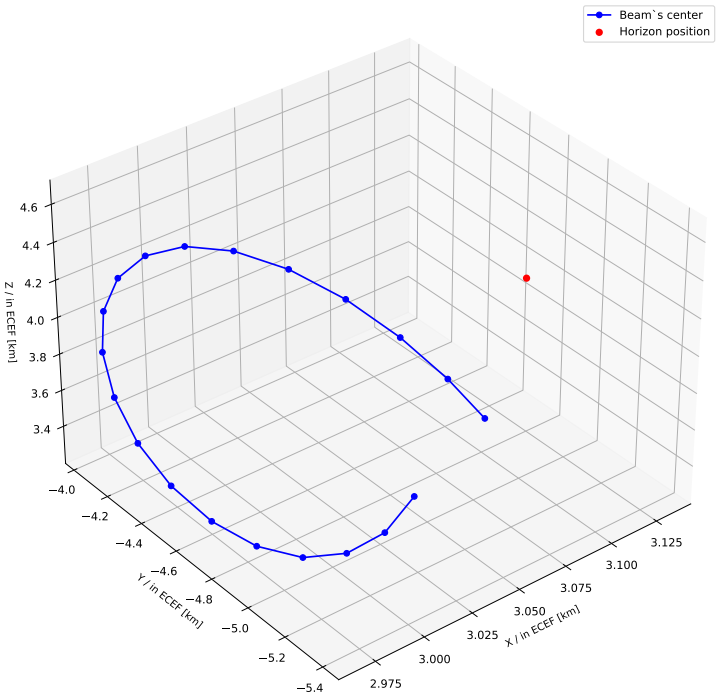


Figure 4.1.: Behavior of the conical pattern in TC1

The behavior of the conical pattern in TC1 is shown in figure 4.1. As evident, the conical pattern is notably offset from the horizon position, implying a requirement to augment ω to achieve a wider conical pattern. If ω is increased, the conical pattern will expand centering around the horizon position.

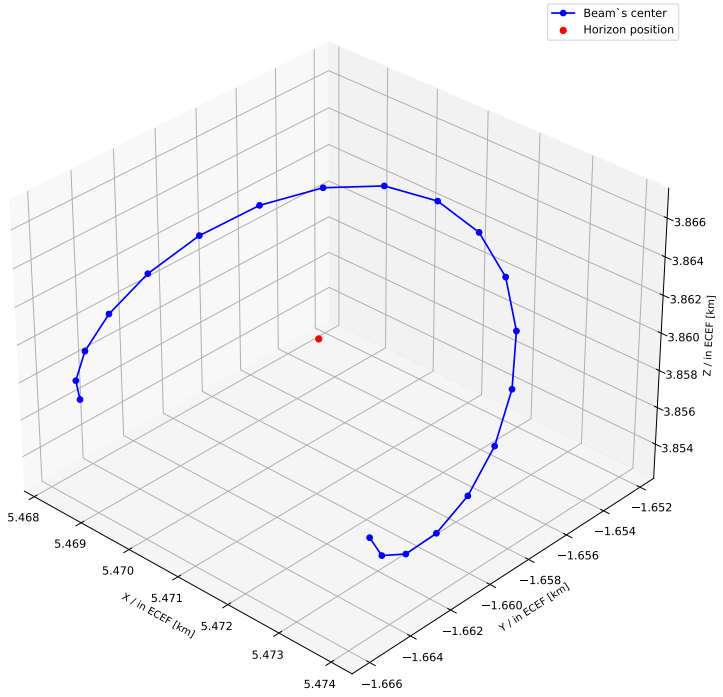


Figure 4.2.: Behavior of the conical pattern in TC2

The behavior of the conical pattern in TC2 is shown in figure 4.2. As evident, the conical pattern centers the horizon position and behaves as anticipated.

The contrast in how the conical pattern behaves in TC1 and TC2 underscores the potential necessity for algorithmic fine-tuning tailored to specific scenarios. Even if the start elevation is the same, the azimuth and the specifications of the TLE change for each trajectory and each position on the trajectory, resulting in varying behavior of the conical pattern. In other words, the essential parameters of the algorithm should be adjusted accordingly for each satellite pass before the search has begun.

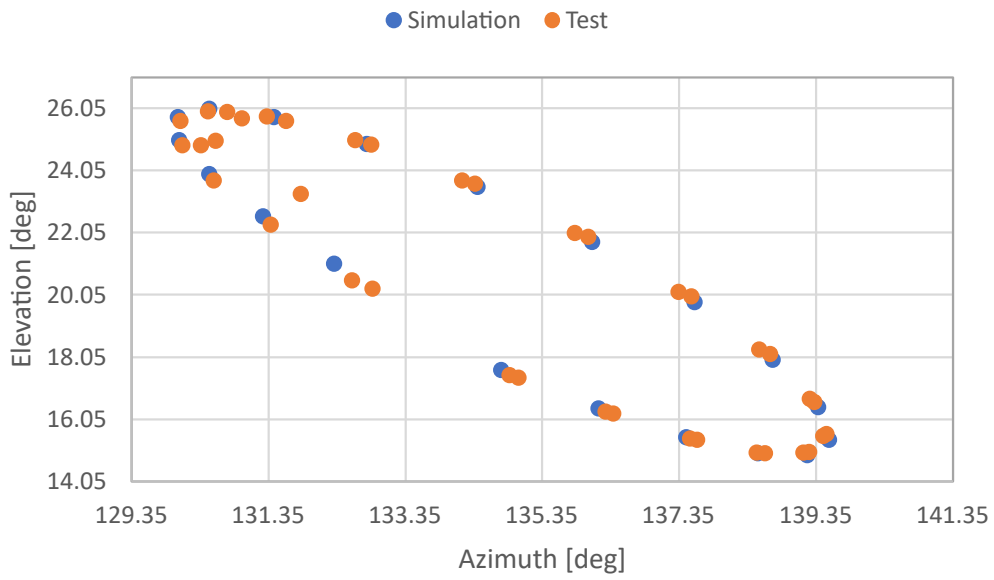


Figure 4.3.: Behavior of the conical pattern in simulation (blue) and tests (orange) in TC1 in HCS

The behavior of the conical pattern in the simulation (blue) and the tests (orange) in TC1 are given in figure 4.3. As can be seen, the antenna behaves as expected, performing the conical shaped pattern. However, an offset is evident when comparing the orange points (tests) to the blue ones (simulations). That can be explained by the anticipated cumulative delay caused by the acceleration and deceleration of the antenna and the environmental effects, such as the wind. This offset cannot be eliminated entirely. Nevertheless, it is possible to undermine its effect by increasing the antenna's acceleration speed. Additionally, a noticeable increase in the offset is observed, particularly in two scenarios. First, when the antenna transitions from lower to higher elevations, it contends with gravity, posing a challenge to acceleration. Second, when the distance between successive steps is large, the antenna must either accelerate more rapidly or attain a higher maximum speed. Nonetheless, it is only possible to change the maximum speed of the antenna. If the maximum speed is set to a higher value, the antenna will try to reach a higher maximum speed in the same time interval, resulting in an increased acceleration. Besides that, the elevation values seem not to be correct. The chosen horizon point has an elevation of 10°. As a result, it is anticipated that the antenna should move around the elevation of 10°. But the antenna performs

the conical search between 14° to 26° elevation. On the other hand, the ECEF coordinates of the performed conical patterns center around the horizon position and are correct, as seen in figure 4.1. Consequently, the mistake in the elevation values points towards a problem in the coordinate transformations from ECEF to HCS. The same problem is also observed for the azimuth values.

The behavior of the conical pattern in the simulation (blue) and the tests(orange) in TC2 are given in figure 4.4. Similar to TC1, the antenna executes the conical pattern as intended. However, a recurring issue persists. The elevation and azimuth values remain inaccurate. This discrepancy becomes evident when the elevation values are examined. As outlined in Table 4.1, the horizon position should possess an elevation of 10°. Yet, it is observed that the elevation value predominantly centers around 1.5°. That signifies a recurring challenge stemming from coordinate transformation. Additionally, the cumulative delay can also be observed here with the slight offset of the antenna values relative to the simulation values. The same affects explained for TC1 are also evident for TC2. Notably, the movement's shape differs compared to TC1, attributed to the parameter ω. While both cases share the same ω value, the distance between the satellite and antenna varies between them. Consequently, the identical ω value yields diverse behaviors in these scenarios.

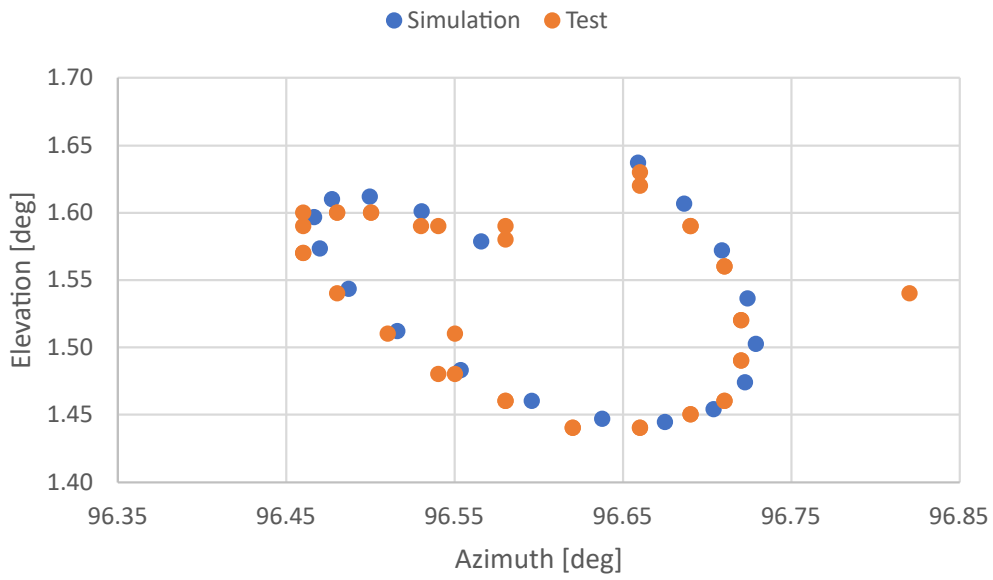


Figure 4.4.: Behavior of the conical pattern in simulation (blue) and tests (orange) in TC2 in HCS

The behavior of the conical pattern in TC3 is shown in figure 4.5. As evident, the conical pattern is slightly offset from the horizon position, implying a requirement to augment ω to achieve a wider conical pattern, just like the TC1. By increasing ω, the conical pattern can expand, aligning more closely with the horizon position. However, it's worth noting that the ideal ω value isn't a one-size-fits-all solution. For instance, in Satellite Pass 1, the trajectory's specific location demands a higher value of ω (in this case, 15 deg/m is insufficient). Conversely, in Satellite Pass 2, the situation differs. Each satellite pass or trajectory introduces variations that directly impact the required conical shape. Therefore, it's imperative to precompute the conical pattern before each pass and

analyze its structure to ensure optimal alignment.

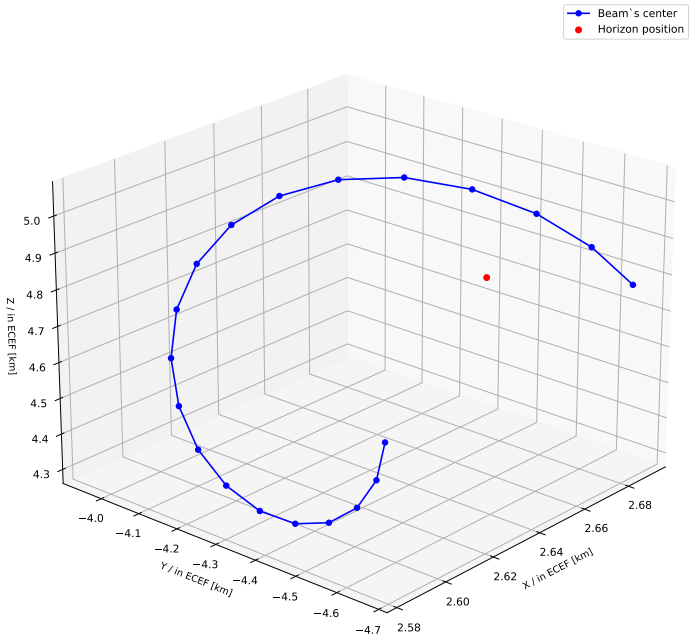


Figure 4.5.: Behavior of the conical pattern in TC3

The behavior of the conical pattern in TC4 is shown in figure 4.6. As evident, the conical pattern centers the horizon position and behaves as anticipated.

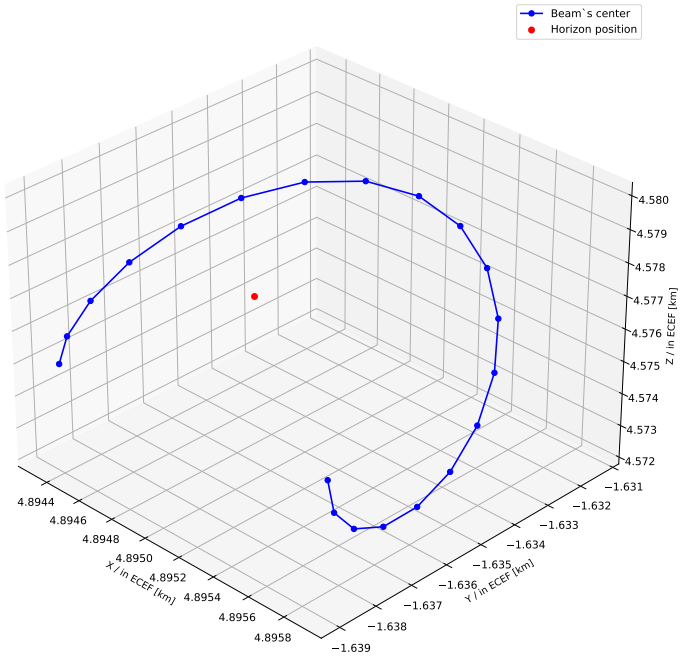


Figure 4.6.: Behavior of the conical pattern in TC4

The behavior of the conical pattern in the simulation (blue) and the tests (orange) in TC3 are given

in figure 4.7. As can be seen, the antenna behaves as expected, executing the conical movement. However, the anticipated cumulative delay caused by the acceleration and deceleration of the antenna and the environmental effects can be observed here as well. Besides that, the elevation of values seems again not to be correct. The chosen horizon point has an elevation of 30°. As a result, it is anticipated that the antenna should move around the elevation of 30°. But the antenna performs the conical search between 14° to 22°. On the other hand, the ECEF coordinates of the performed conical patterns center around the horizon position and are correct, as seen in figure 4.5. Consequently, the fact that the **ECEf!** values are correct but the HCS values are not points towards a transformation mistake from ECEF to HCS.

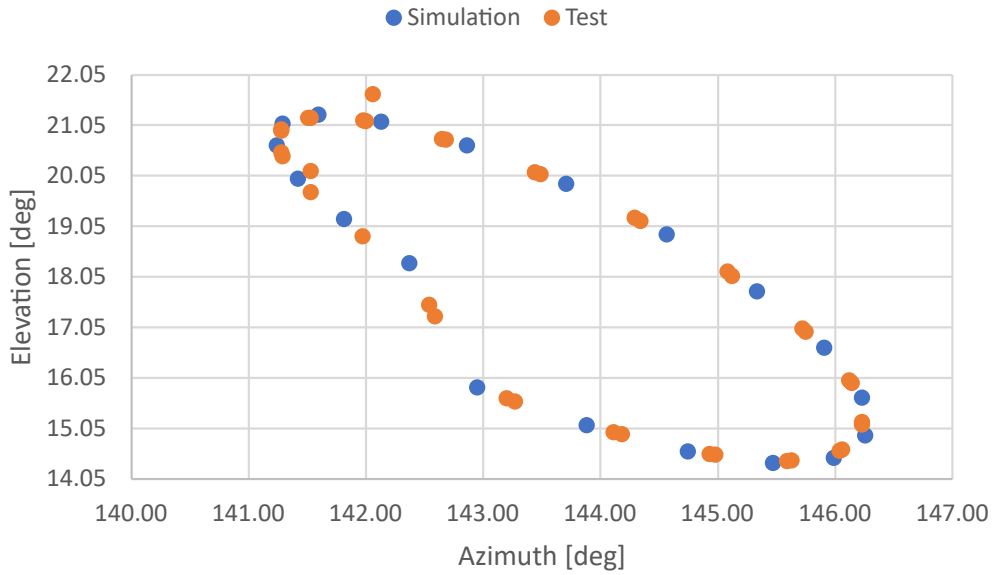


Figure 4.7.: Behavior of the conical pattern in simulation (blue) and tests (orange) in TC3 in HCS

The behavior of the conical pattern in the simulation (blue) and the tests (orange) in TC4 are given in figure 4.8. This visualization notably highlights the issue surrounding coordinate transformation. It is apparent that both the elevation azimuth values are wrongly transformed from ECEF to HCS since the behavior of the pattern was correct in ECEF, as shown in figure 4.6.

Furthermore, a cumulative delay is observed in this case as well. However, the positions logged during the test (orange) exhibit a more scattered distribution compared to the other TCs. Interestingly, the cumulative offset doesn't appear to increase uniformly during execution. This variation can be attributed to strong winds encountered during the test. During this particular test, winds were blowing at a relatively high pace of approximately 9 $\mathrm{km\,h^{-1}}$. These strong winds significantly influenced the antenna's movement, resulting in greater deviations from the calculated values.

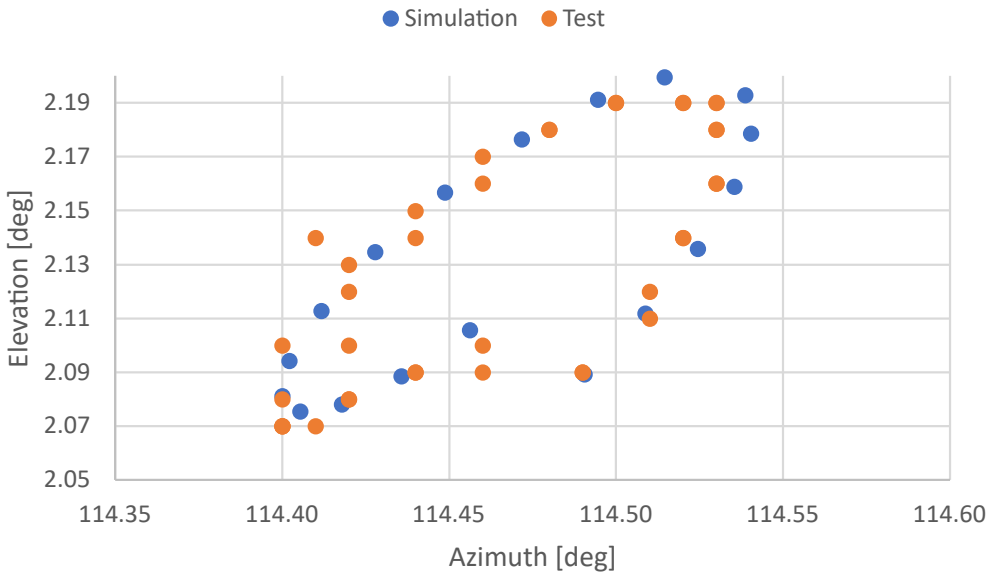


Figure 4.8.: Behavior of the conical pattern in simulation (blue) and tests (orange) in TC4 in HCS

For all the TCs, the mean absolute error (MAE) for both elevation and azimuth is calculated so that the error of the antenna's movement can be analyzed further in detail. The MAE for all the TCs are given in table 4.2. The total MAE, a combination of azimuth MAE and elevation MAE, increases when there's a wider angular gap between the maximum and minimum elevation and azimuth. That is anticipated, as the antenna needs to cover a greater angular distance in the same timeframe, causing higher total MAE. In other words, a larger scanning area during a single conical pattern execution leads to a higher total MAE.

Test cases	TC1	TC2	TC3	TC4	All TCs
Azimuth MAE [deg]	0.15907	0.14149	0.37373	0.07241	0.18668
Elevation MAE [deg]	0.17733	0.09295	0.49844	0.06598	0.20868
Total MAE [deg]	0.33640	0.23444	0.87217	0.13839	0.39536

Table 4.2.: Antenna movement MAE for all TCs

The A_{beam}, $A_{conscan}$ and the increased area for dynamic slant range with the condition $r_{conscan} = 2r_{beam}$ fulfilled for all the TCs are given in table 4.3. The calculations are based on the ECEF coordinates, disregarding the evident coordinate transformation problem in the algorithm since the ECEF coordinates are proven to be correct.

Test cases	TC1	TC2	TC3	TC4
A_{beam} $[km^2]$	37520.54	38739.99	10791.70	11549.33
$A_{conscan}$ $[km^2]$	150082.16	154959.95	43166.79	46197.33
Achieved scanning area	x4	x4	x4	x4

Table 4.3.: Comparison of achieved A_{beam} and $A_{conscan}$

It can be seen that the conical search algorithm ($A_{conscan}$) increases the scanning area four times compared to just pointing the antenna in the estimated position of the satellite (A_{beam}) for all the TCs, achieving the primary objective of the search algorithm. Besides that, it can be seen that both the A_{beam} and $A_{conscan}$ are lower for TCs with higher elevation. That is anticipated since the slant range is smaller for higher elevations, resulting in a narrower scanning area in the sky, as explained in chapter 3.1.

In summary, the antenna executed the desired conical patterns for all the TCs, increasing the scanning area four times for each of the TCs. However, two pivotal issues have surfaced. Firstly, although the computed rotations appear accurate in ECEF coordinates, the conversion from ECEF to HCS is wrong, leading to correct behavior but incorrect coordinates. Secondly, an observable delay cumulatively compounds with the increasing number of conical pattern executions, primarily attributed to the antenna's physical limitations. To address the second issue, increasing the antenna's angular velocity or introducing optimization strategies to refine its motion is essential. One such approach involves programming the antenna to maintain a near continuous acceleration and deceleration between steps, thereby preserving its acceleration/deceleration reached in the preceding step and leveraging it for smoother motion.

4.2. Conical Search

The executed algorithm should perform the blind search and conical search for any given TLE. Therefore, two different satellite passes of the EIVE are chosen to demonstrate the behavior of the conical search in different test cases. For both of the satellite passes, the algorithm starts its search at an elevation of 20° to avoid any negative elevations in the executed conical patterns and to avoid damaging the antenna. It should also be mentioned that the antenna performs the conical search with a maximum speed of 5 deg/sec between the steps, considering the equipment placed on the antenna. The first chosen satellite pass was on 13.09.2023, starting from 23.56 UTC with a duration of 12 min and a maximum elevation of 79.91°. The second satellite pass was on 12.09.2023, starting from 00.21 UTC with a duration of 12 min and a maximum elevation of 53.21°. During both of the satellite passes, the satellite reaches the maximum elevation in 6 min. Furthermore, the TLEs for both the passes are the same since they are only one day apart from each other. The TLE is given below:

- The first TLE line is '1 56937U 23084F 23249.06111443 .00011380 00000-0 63201-3 0 9995'.
- The second TLE line is '2 56937 97.5169 4.3915 0008005 297.0840 62.9574 15.13859529 12884'.

The first satellite pass is shown in azimuth and elevation in the figure 4.9. Noticeably, there is a change of 45° azimuth between the 300. sample and the 360. sample. In other words, the satellite moves 45° azimuth just in 60 s, as explained in chapter 3.4. This sharp shift induces additional errors (cumulative delay) since the antenna must traverse a greater distance during this period between steps with a fixed time frame between each step, leading to higher delays.

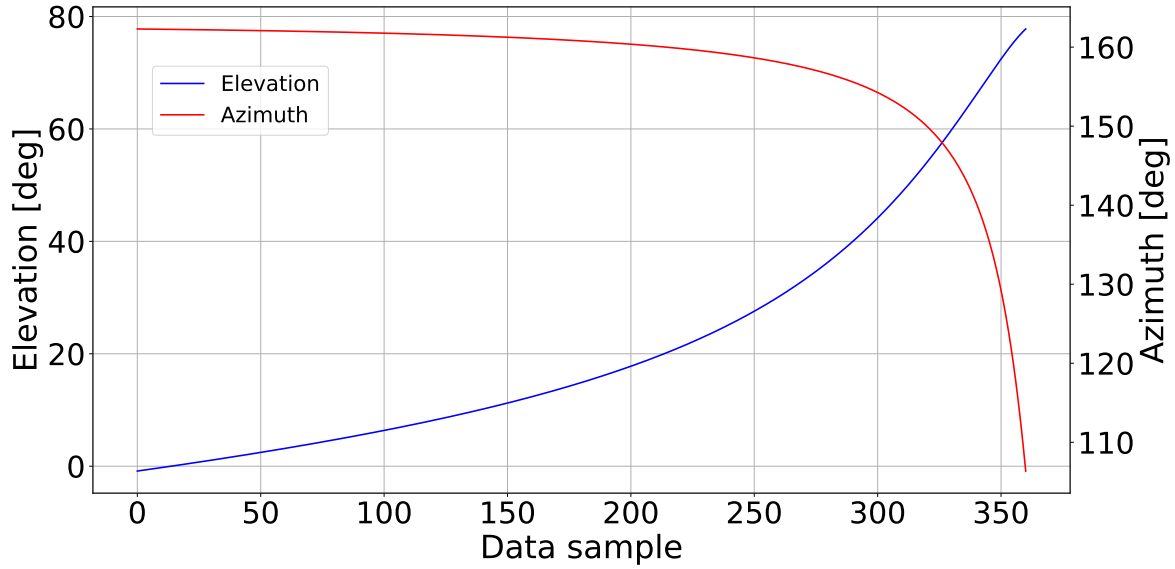


Figure 4.9.: First satellite pass in azimuth (red) and elevation (blue) until the maximum elevation is reached

The second satellite pass is shown in azimuth and elevation in the figure 4.10. The second pass

shows a different characteristic. As can be seen, the azimuth increases with the increasing elevation, and there is also a dramatic shift between the 30th sample and 360th sample, as in pass 1. During this period the satellite moves 60° azimuth just in 60 s.

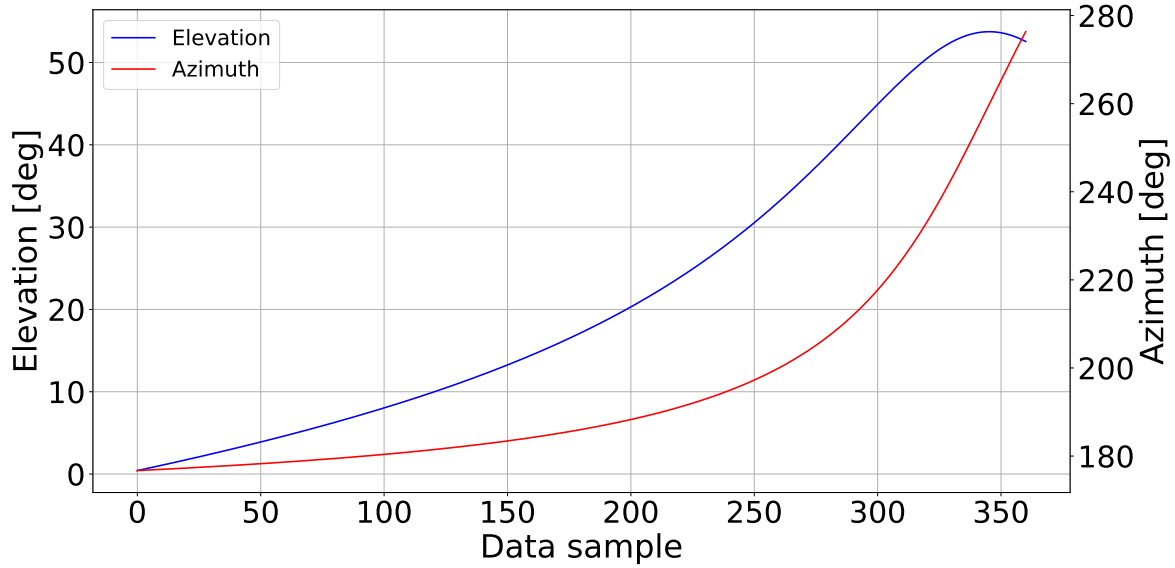


Figure 4.10.: Second satellite pass in azimuth (red) and elevation (blue) until the maximum elevation is reached

The downsampling factor m is the decisive parameter for the conical search since it directly decides how many conical patterns will be executed. Additionally, the number of propagated state vectors for all trajectories is different. Therefore, the number of executed conical patterns (N_{cp}) for different satellite passes with the same m changes depending on the satellite pass. The defined TCs are given in table 4.4 with the critical parameters.

Test cases	m	Satellite pass	$r_{conscan}$	ω	n_{step}	N_{cp}
TC1	10	pass 1	$= 2r_{beam}$	15 deg/s	20	15
TC2	10	pass 2	$= 2r_{beam}$	15 deg/s	20	17
TC3	30	pass 1	$= 2r_{beam}$	15 deg/s	20	5
TC4	30	pass 2	$= 2r_{beam}$	15 deg/s	20	6

Table 4.4.: Conical search test cases during the blind search

The behavior of the conical search until the satellite reaches its maximum elevation for the TC1 in ECEF can be seen in figure 4.11. As evident, the conical search is executed every tenth state vector, resulting in a total of 15 conical patterns executed. These patterns are centered around the estimated satellite trajectory, showing the desired behavior. Moreover, there's a noticeable reduction in the scanning area as elevation increases. That is the result of the decrease in the slant range with the increasing elevation, as discussed in chapter 3.1. Fortunately, this behavior aligns with the algorithm's objectives. Initially, the algorithm aims for a large scanning area to detect and locate the satellite at the first executed conical patterns, as explained in chapter 3.5.

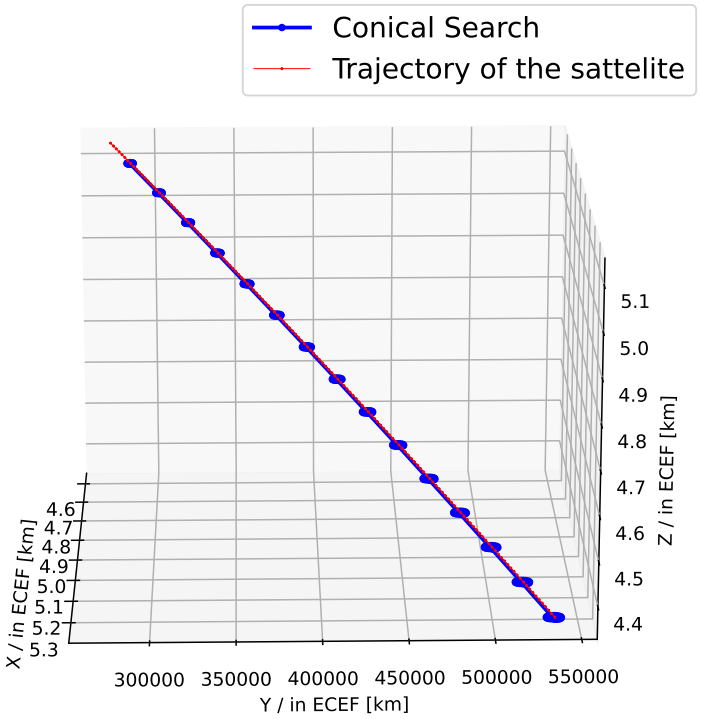


Figure 4.11.: Behavior of the conical search in TC1 in ECEF

The behavior of the conical search until the satellite reaches its maximum elevation for the TC2 in ECEF can be seen in figure 4.12. As evident, the conical search is executed every tenth state vector, resulting in a total of 17 conical patterns executed. The algorithm shows the desired behavior.

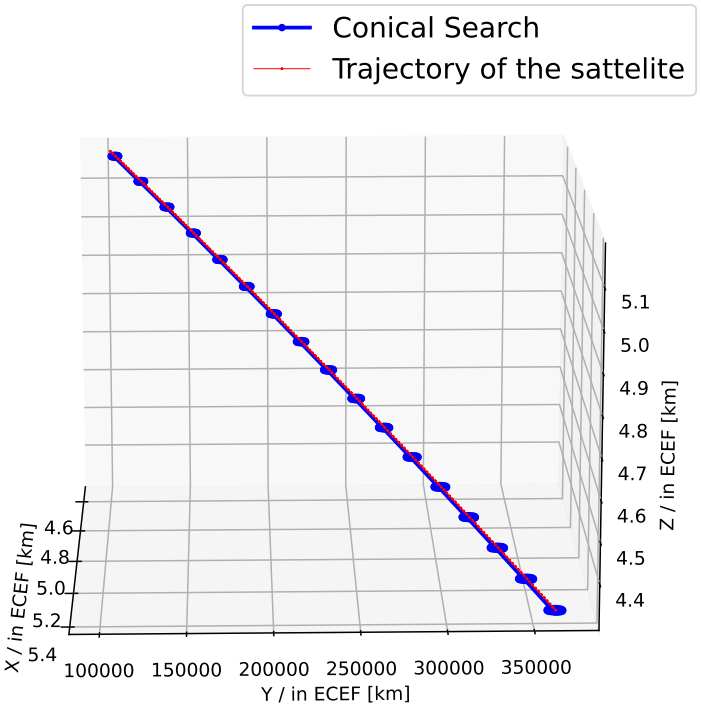


Figure 4.12.: Behavior of the conical search in TC2 in ECEF

In addition, TC2 also exhibits the trend of decreasing scanning areas as elevation increases, aligning with the intended algorithm behavior. Besides that, the variation in the number of conical patterns executed between TC1 and TC2 is attributed to the distance the satellite covers both in azimuth and elevation in the line of sight of the antenna. The satellite covers a higher distance in pass 2, resulting in more propagated state vectors and, therefore, more executed conical patterns.

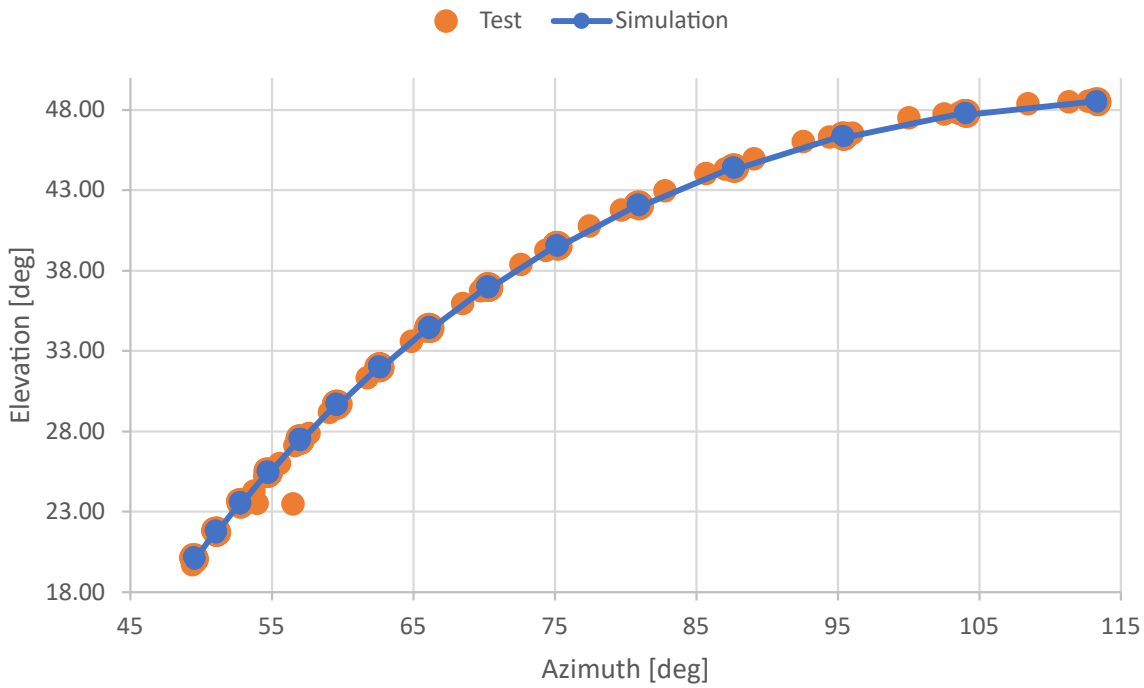


Figure 4.13.: Conical search in simulation (blue) and tests (orange) in TC1 in HCS

The behavior of the conical search until the satellite reaches its maximum elevation in both simulation (blue) and tests (orange) for the TC1 can be seen in figure 4.13. The blue points in the simulation correspond to the point where the conical patterns are conducted. As evident, the concentration of the orange points around the blue ones proves that the antenna executes the conical patterns at the correct positions. Examining the region where only a single conical pattern is executed reveals a behavior consistent with that discussed in Chapter 4.1. An example can be found in Appendix A. On the other hand, two prominent issues are observed during the conical search, similar to what we encountered in the blind search tests. Firstly, there's the issue of cumulative delay caused by the iterative movement of the antenna. This delay grows as the gaps between conical patterns increase. The antenna's movement sequence, first in azimuth and then in elevation, can lead to significant errors, especially when rapid changes in azimuth or elevation are required, such as when the antenna jumps from 88° azimuth to 95° degree azimuth. Secondly, the persistent problem of ECEF to HCS coordinate transformation becomes apparent again. Notably, the antenna fails to reach the expected maximum elevation of 79.91° in both the simulation and the tests despite successfully achieving the desired behavior in ECEF. This inconsistency highlights a potential error in the transformation process from ECEF to HCS coordinates.

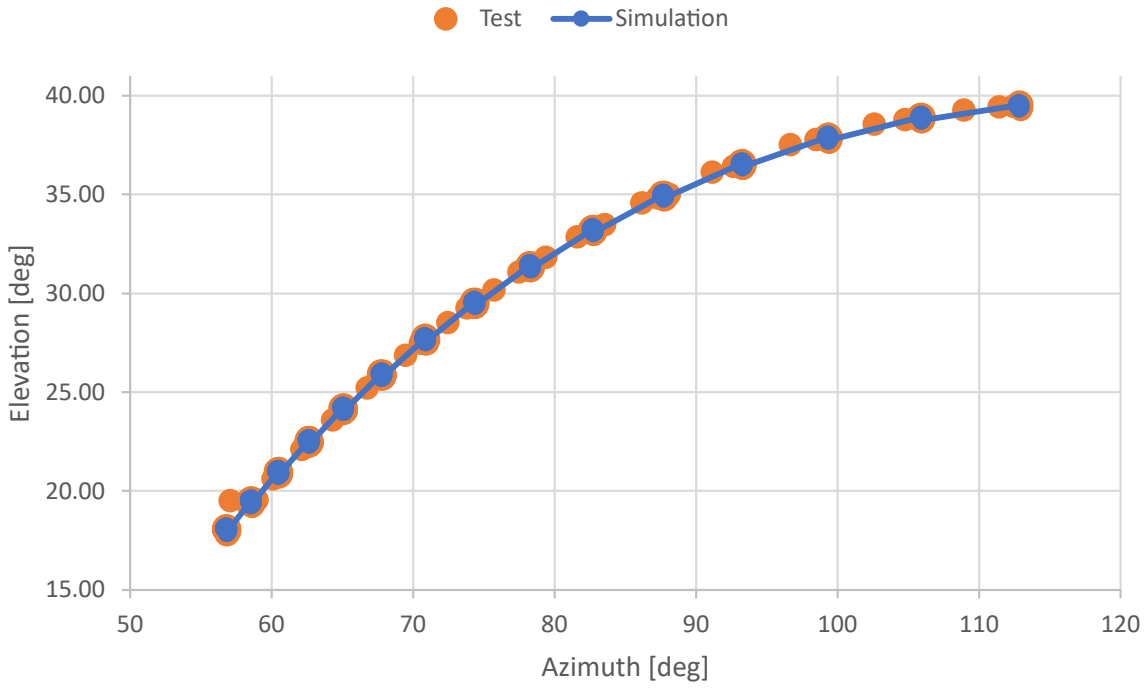


Figure 4.14.: Behavior of the conical search in simulation (blue) and tests (orange) in TC2 in HCS

The behavior of the conical search until the satellite reaches its maximum elevation in both simulation (blue) and tests (orange) for the TC2 can be seen in figure 4.14. The results are similar to the results from TC1. Even though the antenna moves as intended, executing 17 conical patterns correctly, the cumulative delay can still be observed. Moreover, the same coordinate transformation error happens, resulting in the correct behavior but false coordinates. The antenna did not reach the maximum elevation in this TC as well. Interestingly, the elevation and azimuth range for both the TC1 and TC2 are very similar, even though the satellite passes were happening in different elevations and azimuths. This is interesting since the results in ECEF are correct, as can be seen in figure 4.11 and 4.12, whereas the transformed HCS coordinates seem to be wrong.

The behavior of the conical search until the satellite reaches its maximum elevation for the TC3 can be seen in figure 4.15. As evident, the conical search is executed every 30th state vector, resulting in a total of 5 conical patterns executed. These patterns are centered around the estimated satellite trajectory, just like the TC1 and TC2. Moreover, there's a noticeable reduction in the scanning area as elevation increases, as expected. Therefore, it is argued that the algorithm shows the expected behavior in ECEF coordinates for TC3. Besides, one intriguing observation is that the last conical pattern doesn't reach maximum elevation. That is due to the employed downsampling factor. If the downsampling factor is set to 40 instead of 30, that final pattern would get much closer to the maximum elevation. This phenomenon is related to the varying number of state vectors in each propagated trajectory. Consequently, the effect of the downsampling factor varies depending on the specific orbit and the propagated trajectory.

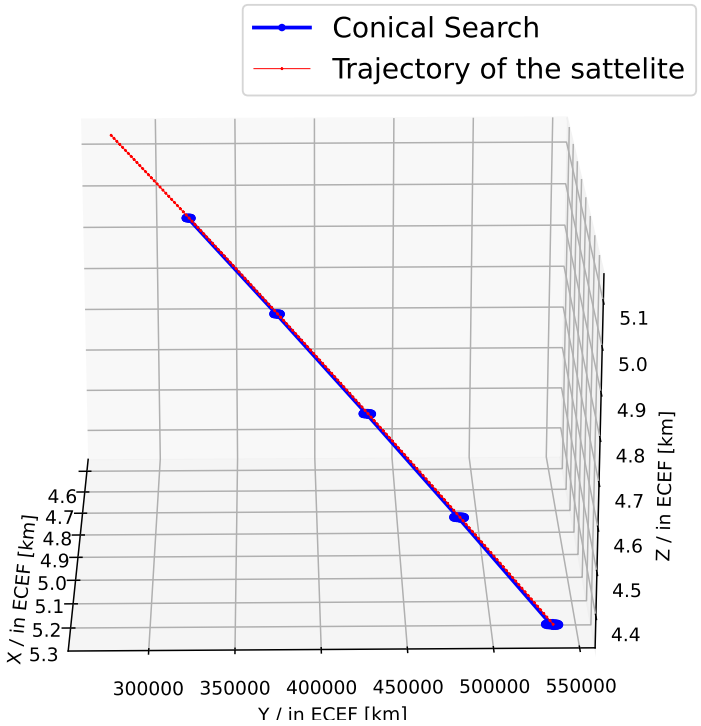


Figure 4.15.: Behavior of the conical search in TC3 in ECEF

The behavior of the conical search until the satellite reaches its maximum elevation for the TC4 can be seen in figure 4.16. As evident, the conical search is executed every tenth state vector correctly, similar to other TC3, resulting in 6 conical patterns executed.

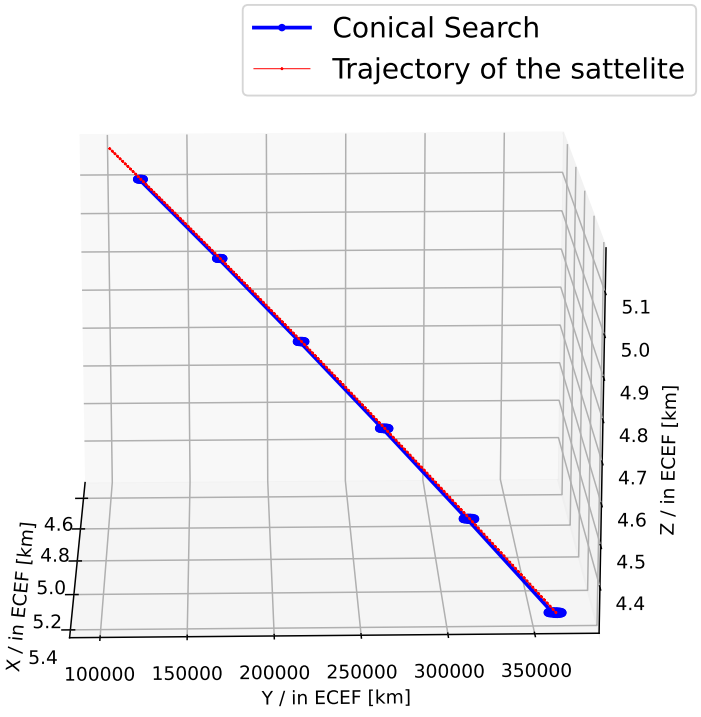


Figure 4.16.: Behavior of the conical search in TC4 in ECEF

The behavior of the conical search until the satellite reaches its maximum elevation in both simulation (blue) and tests (orange) for the TC3 can be seen in figure 4.17. The results are similar to the results from TC1 and TC2. Even though the antenna moves as intended, executing five conical patterns, the cumulative delay is stronger in this TC since there is more distance between each step because of the higher downsampling factor. However, the concentration of the points proves that the conical patterns are executed at the correct positions and behave as expected, similar to the shown example for the TC1 in appendix A. Moreover, the same coordinate transformation error observed in other tests occurs, even though the results in ECEF are correct, as can be seen in figure 4.15. The maximum elevation is also not reached in this TC, even though the number of executed conical patterns match with the results of the ECEF simulations. This situation further proves that the ECEF to HCS transformation is not happening as expected.

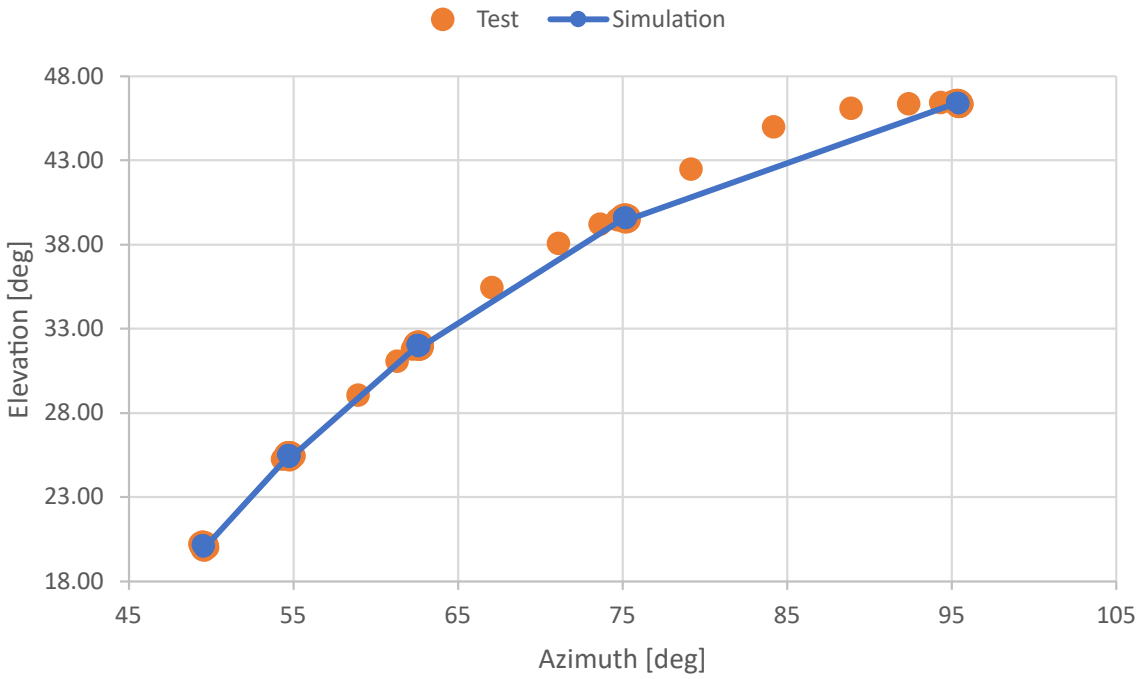


Figure 4.17.: Behavior of the conical search in simulation (blue) and tests (orange) in TC3 in HCS

The behavior of the conical search until the satellite reaches its maximum elevation in both simulation (blue) and tests (orange) for the TC4 can be seen in figure 4.18. The results are similar to the results from other TCs. While the antenna executes its movements as intended, executing six conical patterns, in this TC, it's worth noting that the cumulative delay is more pronounced relative to the TC1 and TC2. This heightened delay occurs because there is a greater distance between each step, a consequence of the higher downsampling factor. Nevertheless, the clustering of points provides strong evidence that the conical patterns are being executed at the correct positions with the correct behavior, similar to the example in appendix A. Furthermore, the coordinate transformation error, as observed in previous tests, still occurs despite the results being accurate in ECEF, as shown in figure 4.16.

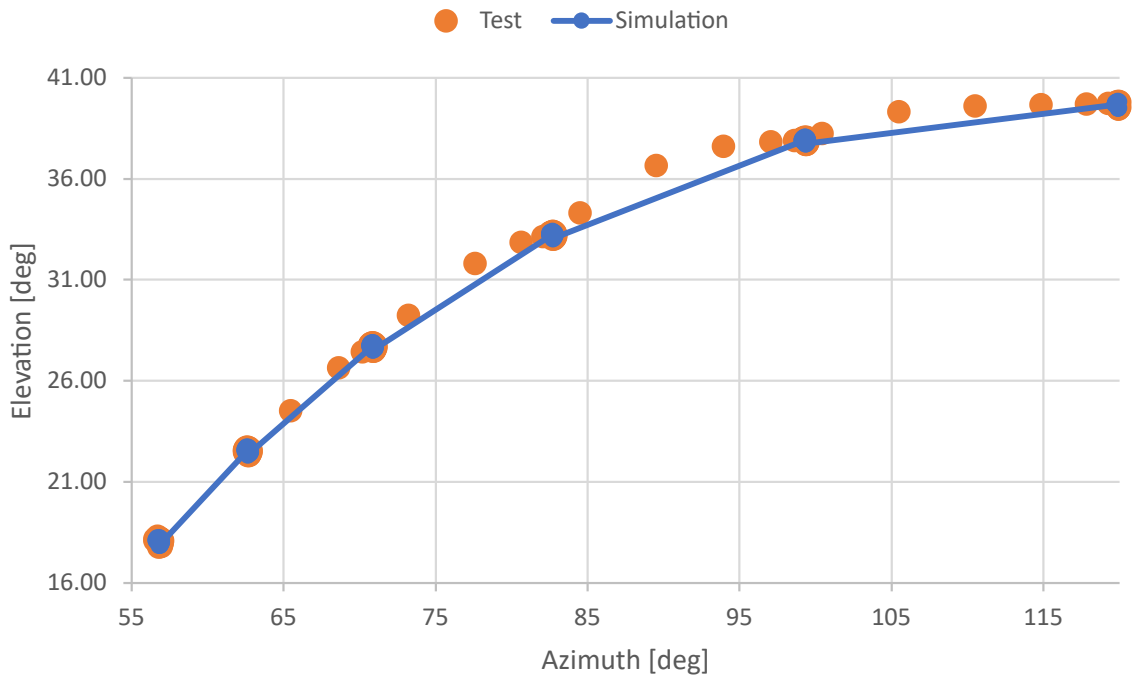


Figure 4.18.: Behavior of the conical search in simulation (blue) and tests (orange) in TC4 in HCS

For all the TCs, the MAE for both elevation and azimuth is calculated so that the error of the antenna's movement can be analyzed further in detail. The MAE for all the TCs are given in table 4.5. The MAE rises during the jumps between the executed conical patterns. This is a predictable outcome because the antenna must traverse a greater distance within the same time frame, leading to a higher MAE.

Test cases	TC1	TC2	TC3	TC4	All TCs
Azimuth MAE [deg]	0.44349	0.71719	0.58704	0.59529	0.58575
Elevation MAE [deg]	0.15806	0.25098	0.16847	0.11203	0.17239
Total MAE [deg]	0.60155	0.96817	0.75551	0.70762	0.75814

Table 4.5.: Antenna movement MAE for all TCs during the conical search

Remarkably, MAE consistently exhibits higher values for azimuth across all the TCs. This outcome is unsurprising, given that azimuth for both satellite passes involves covering a greater angular distance between the minimum and maximum values compared to elevation. Additionally, the total MAE, which is azimuth MAE added to elevation MAE, is highest for TC2 and lowest for TC1, whereas the average total MAE is 0.75814° for all the TCs. It's crucial to emphasize that the MAE values are influenced by environmental factors, such as wind conditions, during the testing phase. Consequently, generalizing these results across all scenarios can be challenging. Nonetheless, it's noteworthy that the total MAE consistently stays below 1° for all TCs.

The average MAE for both satellite passes and downsampling factors is given in table 4.6. If the total MAE for satellite pass 1 and 2 are analyzed, it is seen that the total MAE is higher for pass 2,

as anticipated. That can be explained by the higher angular distance the antenna needs to cover when following the pass 2 compared to pass 1. Moreover, if the results are compared depending on the downsampling factor, it is seen that the total MAE for the case where $m = 10$ is higher than the case where $m = 30$. This outcome aligns with expectations since when $m = 10$, the antenna executes much more conical patterns in the same time period, resulting in a higher cumulative error.

Test cases	Pass 1	Pass 2	$m = 10$	$m = 30$
Azimuth MAE [deg]	0.51527	0.65624	0.58034	0.59117
Elevation MAE [deg]	0.16327	0.18151	0.20452	0.14025
Total MAE [deg]	0.67854	0.83775	0.78486	0.73142

Table 4.6.: Antenna movement MAE for the two satellite passes and the two downsampling factors during the conical search

In table 4.7, the total area scanned in the sky with ($A_{conscan}$) and without (A_{beam}) using the search algorithm is given for all TCs with the condition $r_{conscan} = 2r_{beam}$ fulfilled. The $A_{conscan}$ and A_{beam} are calculated as explained in chapter 3.1, and the calculations are based on the ECEF coordinates, disregarding the evident coordinate transformation problem in the algorithm since the ECEF coordinates are proven to be correct. As apparent, the searched area in the sky increases four times compared to only pointing the antenna to the estimated trajectory. Therefore, it is argued that the algorithm achieves its primary objective and increases the scanning area in the sky, offering a solution to the very low HPBW limitation.

Test cases	TC1	TC2	TC3	TC4
Total A_{beam} $[km^2]$	146477.82	173476.78	54592.38	65012.10
Total $A_{conscan}$ $[km^2]$	585911.29	693907.11	218369.51	260048.41
Achieved scanning area	x4	x4	x4	x4

Table 4.7.: Comparison of the total achieved A_{beam} and $A_{conscan}$

In summary, the antenna effectively executed the desired conical patterns in all four scenarios, achieving the primary objective of increasing the scanning area. However, two critical issues have come to light, like in the blind search tests. Firstly, while the computed rotations are accurate in ECEF coordinates, the conversion to HCS appears flawed, resulting in correct behavior but incorrect coordinates. Secondly, a noticeable delay accumulates as the number of conical pattern executions increases, mainly due to the antenna's physical limitations. The suggested solutions in chapter 4.1 can also improve the behavior of the conical search as a whole.

5. Improvements and Future Work

In this chapter, further possible improvements to the scanning algorithm and the tracking system are discussed and explained. Firstly, the required optimization on the search algorithm is elaborated in section 5.1. Secondly, possible improvements to increase the accuracy of the tracking system are explained and discussed in chapter 5.2.

5.1. Improvements in the search algorithm

The search algorithm serves two main task objectives next to its primary goal of increasing the scanning area. The first involves computing the elevation and azimuth input parameters for the antenna control system, specifically tailored for the CONSCAN before the satellite's arrival. A key challenge arises when translating the computed conical pattern coordinates from ECEF to HCS. As expounded upon in chapter 4, although the desired conical patterns manifest accurately in ECEF coordinates, the conversion to HCS coordinates, concerning the antenna's position, exhibits inaccuracies.

This translation process relies on the utilization of two libraries: the pymap3D library and the orekit library. The orekit library plays a dual role, defining the Earth model and propagating the satellite's potential trajectory, extracted from the TLE. Meanwhile, the pymap3D library is tasked with converting the computed ECEF coordinates, sticking to the specified Earth model, into HCS coordinates relative to the antenna's position.

It's probable that compatibility issues may arise between the orekit and pymap3D libraries. An alternative approach could involve azimuth and elevation conversion, avoiding the pymap3D library. That could entail either manually implementing the intricate ECEF to HCS conversion or seeking assistance from the orekit library.

The second objective is to set the antenna in motion. In this regard, achieving precise time synchronization between the antenna and the satellite takes center stage, a subject discussed in chapter 3.4. It's essential to fine-tune the current time synchronization method and subject it to comprehensive testing to detect and rectify any potential issues.

GPS data can be leveraged to enhance the accuracy of time synchronization, a well-established method employed in orbit propagation and time synchronization for satellites in LEO [32], such as the EIVE. The real-time data acquired from GPS serves as valuable input into the system and can be cross-referenced with the calculated time, as elucidated in Chapter 3.4. Furthermore, there's the intriguing possibility of incorporating this information into the KF as input to further refine the

tracking system's precision [33]. In the work Hauschild and Montenbruck [34], a KF based GPS clock estimation algorithm is demonstrated, and positioning accuracies at the decimeter level and below are reached. Alternatively, in the work of Yang et al. [35], a consider KF is suggested to determine the orbit of the satellite in real-time that can be used to increase the accuracy of both the time synchronization and the tracking accuracy.

5.2. Improvements in the tracking system

Tracking the satellite represents a distinct challenge compared to the initial search and detection phase. It necessitates a heightened level of precision in antenna positioning. Despite the CONSCAN effectively addressing the issue of a narrow HPBW, the employed tracking method, MMT, mandates a continuous signal from the satellite. Relying solely on an iterative CONSCAN approach falls short of fulfilling this requirement, as it solely hinges on the TLE data (which inherently carries some degree of error) and does not dynamically adjust its conical patterns and movement trajectory in response to received signals during its motion.

When a signal is detected from the satellite, MMT provides precise satellite position information for that specific moment. However, this information proves less beneficial in this version of the system since the antenna doesn't capitalize on it. What's imperative is the integration of an error minimization method that recalibrates and continuously updates itself using this data. Such an approach is necessary to leverage the entirety of the available information, significantly enhancing precision and enabling effective satellite tracking with MMT.

Two critical pieces of information offer a pathway for the required error minimization. Firstly, we have the satellite's predicted position, courtesy of the orekit library. Secondly, real-time data on the satellite's actual position is acquired through the MMT whenever a signal is detected. These two data sources essentially lay the groundwork for a KF. As elucidated in Chapter 2.4, the KF seamlessly combines model predictions with actual measurements to estimate the subsequent satellite position. The effectiveness of the KF in error minimization has been well-established, as evidenced by the work of Gawronski and Craparo [15], Souza et al. [16], and Karol et al. [36]. This method has proven to be both stable and exceptionally accurate. However, the KF approach comes with its own problems. The traditional conical scanning estimator works very well under conditions where the radius of the scan is comparable to or larger than the motion caused by tracking or external noises [36]. However, for small $r_{conscan}$, the step response of the estimator highly increases. Considering that the EIVE is in LEO, and consequently the $r_{conscan}$ is relatively small, with a high step response. In the work of Karol et al. [36], a new step response estimator is suggested which overcomes this issue.

Another approach would be the incorporation of the GPS data into the KF system to increase the precision. The GPS data provides real time orbit information which can be used to undermine the error of prediction model (TLE based) that is used in the KF. In the work of Kavitha et al. [37], a method is proposed for fast and precise estimation on ground, of orbits (position and velocity) of

LEO satellites, based on EKF estimation technique using GPS precise data products. They reached an accuracy of up to 1 m. A different approach using the unscented KF and information fusion is proposed in the work of Ning and Fang [38]. Their results show a further improvement in the orbit estimation compared to the EKF based methods. In the work of Mahmut et al. [39], an algorithm based on robust unscented KF is shown. In their specific case the robust unscented KF showed higher precision compared to a standard unscented KF.

In summary, enhancing system precision is imperative for achieving stable satellite tracking and, by extension, reliable information transfer. Various KF applications, encompassing both nonlinear and linear methods, offer viable solutions that can be effectively deployed in the context of the EIVE project.

6. Conclusion

This research thesis primarily focuses on the CONSCAN based search algorithm designed for the ground station antenna within the EIVE project. The overarching objective of the EIVE project is to explore and establish data transfer capabilities using E-band frequency channels. A communication link must be established between the ground based antenna and the satellite to achieve the required communication link. However, there are notable challenges and limitations to overcome. The satellite resides in LEO, and the ground station possesses only one antenna. Consequently, the available scanning area in the sky is inherently limited and relatively small due to the lower slant range, as explained in chapter 3.1. On top of that, the employed antenna has a narrow HPBW of 0.23°, further constraining the scanning area in the sky. Given these constraints, the development of an algorithm that effectively expands the search area in the sky becomes compulsory to precisely localize the satellite.

The CONSCAN based algorithm tackles these limitations and challenges by increasing the scanning area by executing a conical search pattern around the estimated trajectory/position of the satellite. The algorithm implements the desired conical patterns by utilizing the quaternion rotations, as explained in chapter 3.2. Quaternion rotations allow a numerically stable 3D rotation process, achieving the required conical shapes without encountering singularity problems. Furthermore, the search algorithm employs the Orekit library on Python to be able to propagate the estimated trajectory of the EIVE, as detailed in chapters 2.3 and 3.5.1. This trajectory estimation is essential as it is the sole source of information regarding the satellite's position for this system version. It essentially defines the area in the sky that needs to be scanned by the antenna to detect and localize the satellite.

The conducted tests proved that the scanning area was increased four times, as shown in chapter 4, compared to only pointing the antenna at the estimated position, achieving the primary goal of this thesis. Nonetheless, two main problems surfaced during the tests. Firstly, there is an inconsistency in the coordinate transformation process from ECEF to HCS, demanding a rectification. This issue might originate from a potential incompatibility between the utilized libraries. It is likely that the model generated using the Orekit library does not perfectly align with the transformation functions within the Pymap3D library. Secondly, a cumulative positioning error has emerged in the antenna's movements due to its physical limitations and iterative motion. A promising approach to mitigate this effect involves capitalizing on the antenna's acceleration. This issue can be addressed effectively by ensuring that the antenna never comes to a complete halt and utilizing its existing acceleration to minimize cumulative errors.

Solving the coordination transformation problem and fine-tuning the time synchronization method,

as outlined in Chapter 3.4, stand as crucial steps for further refinement of the search algorithm. However, it's important to note that testing the implemented time synchronization method remains pending due to the coordination transformation issue, which currently obstructs the detection of any satellite signals. Consequently, a conclusive determination of the time synchronization's functionality has yet to be made.

Another significant challenge on the horizon involves securing a continuous signal from the satellite to facilitate the planned data transfer, during which the tracking algorithm takes center stage. The tracking algorithm relies on the MMT method, as outlined in Chapter 2.2.2, and it excels at providing precise measurements, pinpointing the satellite's position upon signal detection. However, it does not fully harness this information's potential. An error minimization method is essential to fully capitalize on the potential of the MMT method, as discussed in chapter 5.2. A KF approach can help the system establish the required continuous data transfer by utilizing both the trajectory estimation of the Orekit library and the measurement of MMT system, increasing the tracking precision. In addition to that, a combination of both the KF and the GPS based methods can help to refine the possible time synchronization problems between the antenna and the satellite, further improving the accuracy and stability of the entire system.

In conclusion, this research thesis achieved its principal goal by increasing the scanning area of the antenna, allowing the system to overcome its limitations. Despite the challenges mentioned in the previous sections, the work described in this research thesis can be considered as a starting point for the development of a tracking system, laying the groundwork for further improvement of the search algorithm itself and the whole tracking system.

Bibliography

[1] B. Schoch. Projekt eive. [Online]. Available: https://www.ilh.uni-stuttgart.de/forschung/mmw/EIVE/

[2] L. Manoliu, B. Schoch, S. Haussmann, A. Tessmann, R. Henneberger, J. Freese, F. Steinmetz, D. Wrana, J. Wörmann, M. Koller, and et al., "The technology platform of the eive cubesat mission for high throughput downlinks at w-band," *Acta Astronautica*, vol. 205, p. 80–93, 2023.

[3] I. Verspohl, "Development of a control concept and implementation of the e-band multimode tracking for the eive ground station," 04 2021.

[4] W. Gawronski and E. Craparo, "Three scanning techniques for deep space network antennas to estimate spacecraft position," *Interplanetary Network Progress Report*, 07 2001.

[5] S. Kim and M. Kim, "Rotation representations and their conversions," *IEEE Access*, vol. 11, pp. 6682–6699, 2023.

[6] P. Harati, B. Schoch, A. Tessmann, D. Schwantuschke, R. Henneberger, H. Czekala, T. Zwick, and I. Kallfass, "Is e-band satellite communication viable?: Advances in modern solid-state technology open up the next frequency band for satcom," *IEEE Microwave Magazine*, vol. 18, no. 7, p. 64–76, 2017.

[7] U. Lewark, J. Antes, J. Walheim, J. Timmermann, T. Zwick, and I. Kallfass, "Link budget analysis for future e-band gigabit satellite communication links (71–76 and 81–84 ghz)," *CEAS Space Journal*, vol. 4, 06 2013.

[8] M. Berioli, A. Molinaro, S. Morosi, and S. Scalise, "Aerospace communications for emergency applications," *Proceedings of the IEEE*, vol. 99, no. 11, p. 1922–1938, 2011.

[9] P. Pedrosa, D. Castanheira, A. Silva, R. Dinis, and A. Gameiro, "A state-space approach for tracking doppler shifts in radio inter-satellite links," *IEEE Access*, vol. 9, pp. 102 378–102 386, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:236481332

[10] M. Neinavaie, J. Khalife, and Z. M. Kassas, "Acquisition, doppler tracking, and positioning with starlink leo satellites: First results," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 3, pp. 2606–2610, 2022.

[11] G. Hawkins, D. . Edwards, and J. McGeehan, "Tracking systems for satellite communications," *Proceedings of the IEEE*, vol. 135, no. 5, pp. 393–407, 1988.

[12] H. D. Curtis, *Orbital Mechanics for Engineering Students: Revised fourth edition*. Elsevier, 2021.

[13] M. Hirsch, "Pymap3d: 3-d coordinate conversions for terrestrial and geospace environments," *Journal of Open Source Software*, vol. 3, no. 23, p. 580, 2018. [Online]. Available:

https://doi.org/10.21105/joss.00580

[14] R. Dadkhah Tehrani, H. Givi, D.-E. Crunteanu, and G. Cican, "Adaptive predictive functional control of x-y pedestal for leo satellite tracking using laguerre functions," *Applied Sciences*, vol. 11, no. 21, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/21/9794

[15] W. Gawronski and E. Craparo, "Antenna scanning techniques for estimation of spacecraft position," *IEEE Antennas and Propagation Magazine*, vol. 44, no. 6, pp. 38–45, 2002.

[16] A. L. Souza, J. Y. Ishihara, H. C. Ferreira, R. A. Borges, and G. A. Borges, "Antenna pointing system for satellite tracking based on kalman filtering and model predictive control techniques," *Advances in Space Research*, vol. 58, no. 11, pp. 2328–2340, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117716304343

[17] A. L. G. Souza, G. de Araujo Borges, Joao, Y. Ishihara, H. C. Ferreira, A. Renato, Borges, V. A. Larin, and V. V. Belikov, "Antenna pointing error estimation using conical scan technique and kalman filter," 2013.

[18] D. B. Eldred, "An improved conscan algorithm based on a kalman filter," 1994.

[19] L. Chen, N. Fathpour, and R. Mehra, "New algorithms for estimating spacecraft position using scanning techniques for deep space network antennas," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5809, 05 2005.

[20] A. K. Pandey, "Design of multimode tracking system for earth station antenna," pp. 1–4, 2016.

[21] H. Bayer, A. Krauss, R. Stephan, and M. Hein, "A dual-band multimode monopulse tracking antenna for land-mobile satellite communications in ka-band," pp. 2357–2361, 03 2012.

[22] T. Paulet and B. Cazabonne, "An open-source solution for tle based orbit determination," 04 2021.

[23] L. Maisonobe, V. Pommier, and P. Parraud, "Orekit: An open source library for operational flight dynamics applications," 04 2010.

[24] J. Bayard, B. Cazabonne, M. Journot, and P. Cefola, "Accurate orbit determination using semi-analytical satellite theory and recursive filter," 09 2021.

[25] M. d. F. Coelho, K. Bousson, and K. Ahmed, "An improved extended kalman filter for radar tracking of satellite trajectories," *Designs*, vol. 5, no. 3, 2021. [Online]. Available: https://www.mdpi.com/2411-9660/5/3/54

[26] R. Roger and J. Labbe, *Kalman and Bayesian Filters in Python*. Ebook, 2020.

[27] C. A. Balanis, *Antenna theory: Analysis and design*. Wiley, 2016.

[28] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "Scipy 1.0: Fundamental algorithms for scientific computing in python," *Nature*

Methods, vol. 17, pp. 261–272, 2020.

[29] M. Boyle, B. Bonnett, J. Long, M. Ling, L. C. Stein, E. Wieser, D. A. B. Iozzo, H. Haglid, J. Belmonte, J. Long, M. Wiebe, Y. Li, Z. Vinícius, and J. Macfarlane, "moble/quaternion: Release v2022.4.3," Feb 2023.

[30] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. John Wiley Sons, 2020.

[31] G. Taubin, "3d rotations," *IEEE Computer Graphics and Applications*, vol. 31, 11 2011.

[32] T. P. Yunck, W. G. Melbourne, and C. L. Thoenton, "Gps-based satellite tracking system for precise positioning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-23, no. 4, pp. 450–457, 1985.

[33] O. Montenbruck and P. Ramos-Bosch, "Precision real-time navigation of leo satellites using global positioning system measurements," *GPS Solutions*, vol. 12, pp. 187–198, 07 2007.

[34] A. Hauschild and O. Montenbruck, "Kalman-filter-based gps clock estimation for near real-time positioning," *GPS Solutions*, vol. 13, no. 3, p. 173–182, 2008.

[35] Y. Yang, X. Yue, and A. G. Dempster, "Gps-based onboard real-time orbit determination for leo satellites using consider kalman filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 2, pp. 769–777, 2016.

[36] R. Karol and G. Ristroph, "Improving conical scanning for satellite tracking on-the-move," 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:15636037

[37] S. Kavitha, P. Mula, M. Kamat, S. Nirmala, and J. G. Manathara, "Extended kalman filter-based precise orbit estimation of leo satellites using gps range measurements," *IFAC-PapersOnLine*, vol. 55, no. 1, pp. 235–240, 2022, 7th International Conference on Advances in Control and Optimization of Dynamical Systems ACODS 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896322000398

[38] X. Ning and J. Fang, "An autonomous celestial navigation method for leo satellite based on unscented kalman filter and information fusion," *Aerospace Science and Technology*, vol. 11, no. 2, pp. 222–228, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963807000028

[39] M. O. Karslioglu, E. Erdogan, and O. Pamuk, "Gps-based real-time orbit determination of low earth orbit satellites using robust unscented kalman filter," *Journal of Aerospace Engineering*, vol. 30, no. 6, p. 04017063, 2017. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29AS.1943-5525.0000779

List of Abbreviations

EIVE	Exploratory In-Orbit Verification of an E/W-Band Satellite Communication Link
LEO	Low Earth Orbit
CubeSat	Square-shaped Miniature Satellite
HPBW	Half Power Beamwidth
CONSCAN	Conical Scanning Algorithm
MMT	Multimode Tracking
KF	Kalman Filter
HCS	Horizontal Coordinate System
ECEF	Earth-fixed Coordinate System
LMS	Least Mean Square Estimation
FFT	Fast Fourier Transformation
MCMC	Markov Chain Monte Carlo
TLE	Two Line Element
ADC	Analog Digital Converter
SGP	Simplified General Perturbations
GNSS	Global Navigation Satellite System
EKF	Extended Kalman Filter
MD	Multipication/Division Operation
AS	Addition/Subtraction Operation
MLFC	Math Library Function Call
GUI	Graphical User Interface
UTC	Coordinated Universal Time
TC	Test Case
GPS	Global Positioning System
MAE	Mean Absolute Error

A. Conical Search Tests

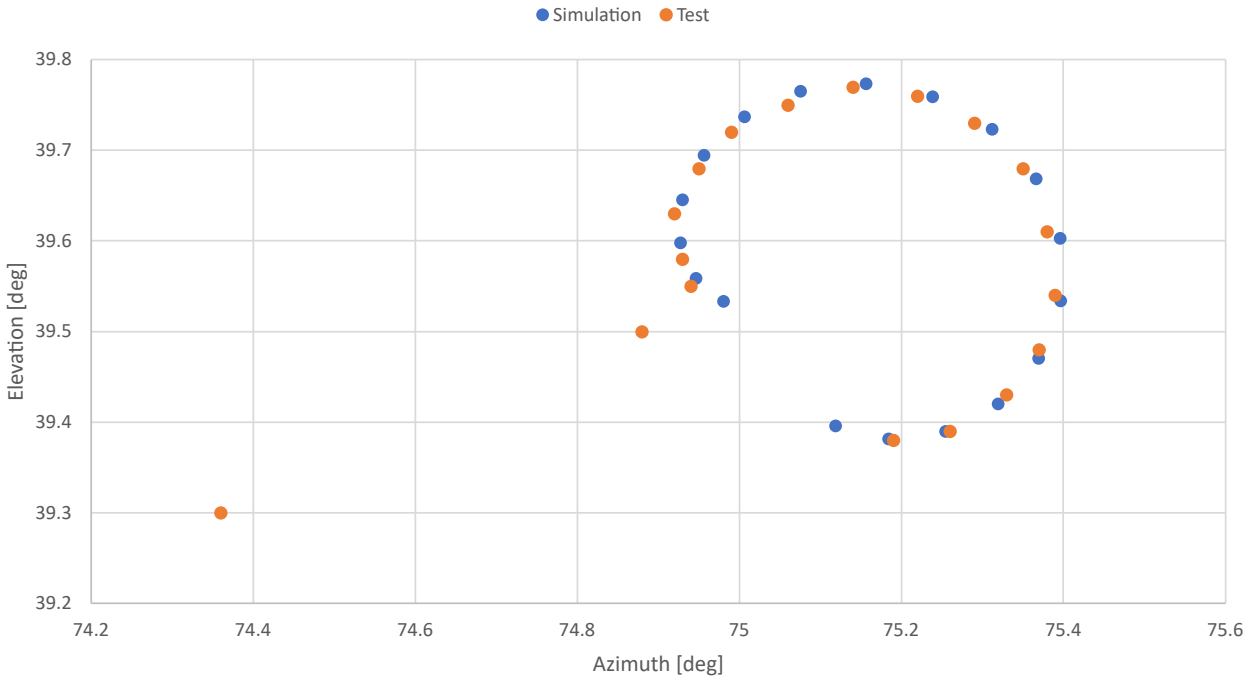


Figure A.1.: Conical search TC1 in HCS, the conical pattern around 75° azimuth and 39° elevation

The executed conical pattern in TC1 at 75° azimuth and 39° elevation. The anticipated behavior is achieved. However, the cumulative delay can be observed. Additionally, the two points at 74.4° azimuth and 39.3° elevation and 74.9° azimuth and 39.5° elevation demonstrates how the antenna conducts the jump between the conical patterns. This behavior can be observed for all the conical patterns in each TC in the conical search tests.