UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Comparison of Satellite Tracking Techniques in Inclined Orbit.

Thiago Lima Sarmento

DM: 30/2019

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2019

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Thiago Lima Sarmento

Comparison of Satellite Tracking Techniques in Inclined Orbit.

DM: 30/2019

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2019

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Thiago Lima Sarmento

Comparison of Satellite Tracking Techniques in Inclined Orbit.

A dissertation submitted to the examination committee in the graduate department of Electrical Engineering at the Federal University of Pará in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering with emphasis in Telecommunications.

UFPA / ITEC / PPGEE

Campus Universitário do Guamá

Belém-Pará-Brasil

2019

Dados Internacionais de Catalogação - na – Publicação (CIP) Sistema de Bibliotecas da UFPA

S246c541e Sarmento, Thiago Lima, 1995-

Comparison of satellite tracking techniques in inclined orbit / Thiago Lima Sarmento.-2019.
Orientador: Aldebaro Barreto da Rocha Klautau Júnior
Dissertação (Mestrado) - Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2019.

1. Estações terrestres (Telecomunicação por satélite). 2. Satélites artificiais – Órbitas. 2. Satélites artificiais em telecomunicação. I.Título.

CDD 23. ed. 621.38853

Elaborada por Lucicléa S. de Oliveira – CRB -2/648

Comparison of Satellite Tracking Techniques in Inclined Orbit.

Dissertação de mestrado submetida à avaliação da banca examinadora aprovada pelo colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará e julgada adequada para obtenção do Grau de Mestre em Engenharia Elétrica na área de Telecomunicações.

Aprovada em _____ / _____ / _____

Prof. Dr. Aldebaro Barreto da Rocha Klautau Júnior

ORIENTADOR

Profa. Dra. Jasmine Priscyla Leite de Araujo

MEMBRO DA BANCA EXAMINADORA

Prof. Dr. Diogo Lobato Acatauassú Nunes

MEMBRO DA BANCA EXAMINADORA

Profa. Dra. Maria Emília de Lima Tostes

DIRETOR DA PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Acknowledgments

I thank God first for enlightening me all my way and giving me the strength to overcome all the challenges of this infinity war that is life.

Secondly, I appreciate all the support and understanding of my advisor Prof. Dr. Aldebaro Klautau, who shared valuable knowledge during my academic journey. I am grateful for the opportunity I was given to work at LASSE, which has brought about changes in my personality, both professionally and personally. There were many friendships cultivated in this one of the best laboratories of UFPA.

I would also like to thank the support received from my project team on LASSE. Thanks especially Kazuya Nakamura and Sofia Klautau, for the immense contribution you have made in my work. Thank you very much, and certainty my dissertation would not exist without your cooperation.

Last but not least, I would like to thank my girl and best friend Larissa Nogueira for the support and the right words for each moment. My family for the unique and endless support and understanding given to me during the writing of this paper. My parents Altenir and Ivone Sarmento, and my sister Samantha Sarmento, ”If I have seen further than others, it is by standing upon the shoulders of giants.”

Thiago Lima Sarmento

09/2019

We can only see a short distance ahead, but we can see plenty there that needs to be done.

Alan Turing

List of Acronyms

AESA Active Electrically Scanned Antenna

ACU Antenna Control Unit

ADU Antenna Drive Unit

APU Antenna Power Unit

ASU Antenna Sensor Unit

DL Deep Learning

DQN Deep Q Network

GPS Global Positioning System

INMET Using data from the National Institute of Meteorology

LASSE Telecommunications, Automation and Electronics Research and Development Center

ML Machine Learning

NASA National Aeronautics and Space Administration

NORAD North American Aerospace Defense Command

PPGEE Electrical Engineering Graduate Program

PCT Science and Technology Park

RL Reinforcement Learning

SDR Software Defined Radio

TLE Two-line Elements

UFPA Federal University of Pará

USRP Universal Software Radio Peripheral

UNOOSA United Nations Office for Outer Space Affairs

List of Figures

List of Tables

Contents

Abstract

This work implements satellite tracking techniques in inclined orbit and compares their performances with other techniques, describing operation and implementing in a simulation environment and in a real control system to generate the performance evaluation of each one. One of the techniques investigated is reinforcement learning. Tracking satellites in inclined orbit is of paramount importance for telecommunications using this type of link, allowing automatic communication maintenance and extending the service life of satellite services in this situation. Algorithms widely used in tracking, both in the literature and in commercial equipment, result in estimates or predictions of satellite position rather than actual position, and require a study of the specific characteristics of the region where the base station is located. The complexity and investments in the tracking technique vary according to the commitment made in the antenna installation and control systems, being necessary to compare the existing methods before their implementation. The work develops the environment necessary to simulate satellite communication, from reception to antenna movement, to analyze the performance of the technique in different controlled situations. Are also presented the actual implementation techniques results in a real antenna control system.

Resumo

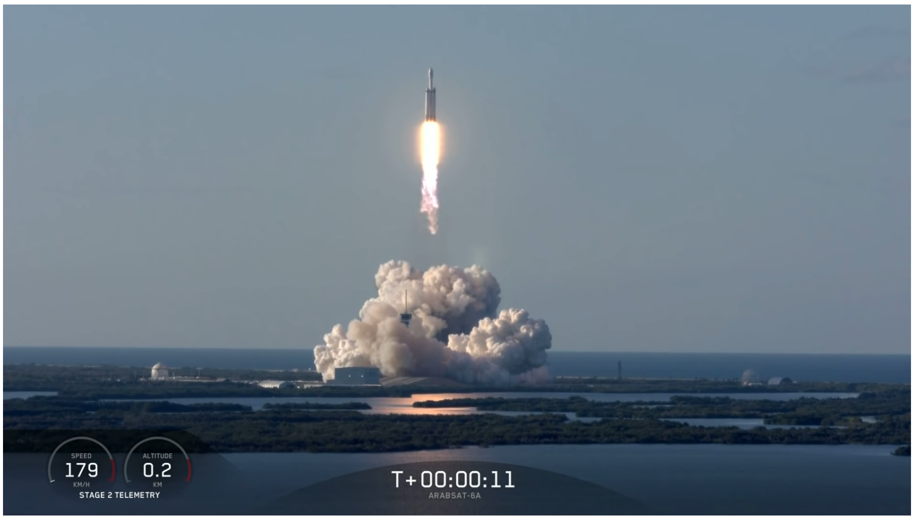
Este trabalho implementa técnicas de rastreio de satélites em órbita inclinada e compara seus desempenhos com outras técnicas, descrevendo funcionamento e implementando em ambiente de simulação e em um sistema de controle real para gerar a avaliação de desempenho de cada uma. Uma das técnicas investigadas é o aprendizado por reforço (”reinforcement learning”). O rastreio de satélites em órbita inclinada é de suma importância para as telecomunicações que utilizam este tipo de enlace, permitindo a manutenção da comunicação de forma automática e aumentando a vida útil dos serviços de satélites nesta situação. Os algoritmos amplamente utilizados em rastreio, tanto na literatura quanto em equipamentos comerciais, tem como resultado estimações ou predições da posição do satélite e não sua posição real, e requerem um estudo das características específicas da região onde a estação base está localizada. A complexidade e investimentos na técnica de rastreio variam de acordo com o compromisso assumido na instalação da antena e sistemas de controle, sendo necessária a comparação dos métodos existentes antes de sua implementação. O trabalho desenvolve o ambiente necessário para simular a comunicação por satélite, desde sua recepção até a movimentação da antena, para analisar o desempenho da técnica em diferentes situações controladas. Também são apresentados resultados reais da implementação das técnicas em um sistema de controle de antenas real.

Chapter 1

Introduction

Satellites are essential for global communications, enabling new possibilities and applications never imagined before. The possibility of establishing a long-range link without the need for channel infrastructure investments, in remote locations and devoid of other types of communications is among the reasons for choosing a satellite link. According to the Index of Objects Launched into Outer Space [1], maintained by the United Nations Office for Outer Space Affairs (UNOOSA), 4987 satellites are orbiting planet Earth providing services like television, telephone, radio, internet, GPS and military applications. With the emergence of new space launch technologies, such as the launching of the Arabsat satellite by SpaceX [2] using the first reusable rockets that landed safely on Earth, showed on Figure 1.1, the increase of the use of satellite links and the services it provides is expected in the coming years [3].

Figure 1.1: Arabsat-6A Satellite launch.



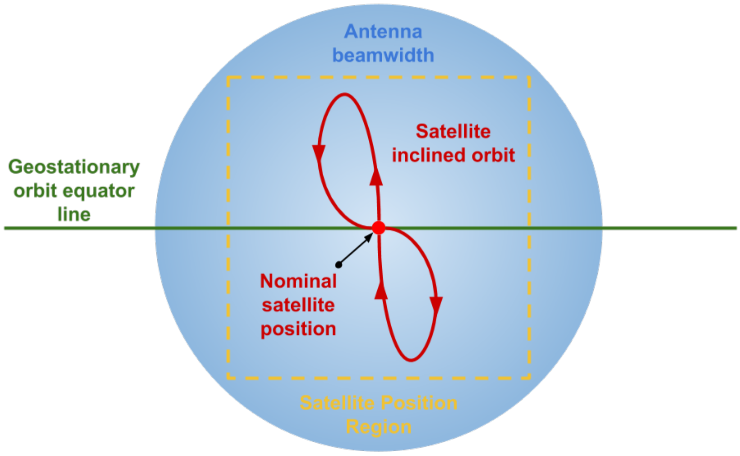
Source: Arabsat-6A Mission Webcast.

Long-distance satellite communication links have inherent peculiarities, and challenges arise because of these characteristics. One of these challenges is to keep the antenna pointing in the best satellite signal reception position, considering the possibility of either a geostationary, geosynchronous or non-stationary orbit. The object of study in this work is the satellite's geosynchronous inclined orbits, which have perturbations in its geostationary orbit. Geostationary satellites follow the earth's rotational motion, making them appear to be standing still to an observer at a fixed point on the Earth. Communications and observation applications of specific regions on the planet use this type of satellite because they are stationary for a terrestrial reference.

1.1 Inclined Orbit of Satellites

An inclined orbit, in celestial mechanics, is any orbit whose orbital plane does not coincide with the reference plane. The term is used in satellite literature [4] to define a satellite that does not maintain its strictly geostationary orbit. For a terrestrial observer, the satellite moves around its original position over time, as shown in the Figure 1.2.

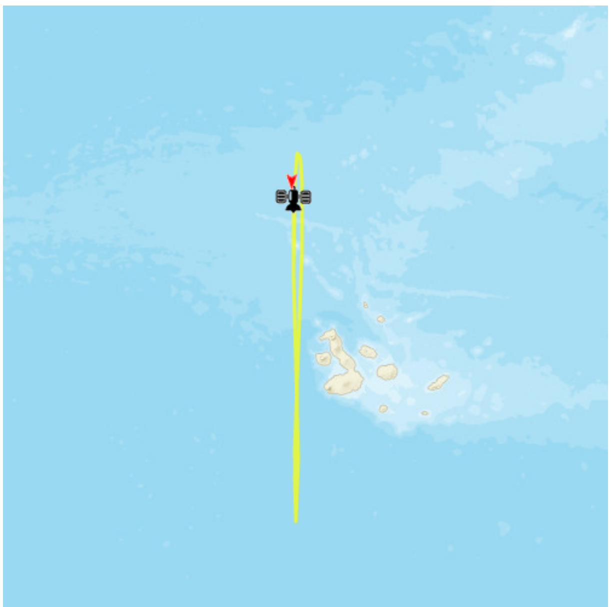
Figure 1.2: Inclined orbit example.



The Figure 1.3 exemplifies the inclined orbit, showing the current orbit of the Brazilsat B4 satellite.

Source: Author.

Figure 1.3: Brazilsat B4 orbit.



Source: N2YO [5].

The satellite is launched with fuel so that it can reposition itself in the proper position and correct the disturbances in its orbit. Since the satellite has limited fuel, near the end of its calculated useful life, it is no longer possible to correct its reference position, and the satellite goes into inclined orbit. These movements affect the reception of the satellite signal made by the antenna over time, decreasing the intensity as it moves away from the pointing and they may even make communication impossible. However, while the satellite can no longer reposition itself, it remains fully operational for telecommunications purposes, even when far from its reference position. Techniques to keep the antenna pointing to the correct position are used to work around the inclined orbit problem and extend the lifetime of satellite-associated services.

1.2 Satellite Tracking Techniques

Satellite tracking techniques are methods developed to position the antenna pointing to the spot of best signal reception, following the satellite's orbit over time. The commitment established in the satellite link is crucial for the choice of the composition of the tracking system that it will adopt, among the variants are the antenna construction and installation, the equipment involved and the chosen technique. This work classifies the satellite tracking techniques studied in three categories:

- Mathematical models of orbit prediction based onspecific characteristics of each satellite, such as the NORAD databases.

- Supervision of the satellite signal characteristics received by the antenna to estimate the best position of reception, searching continuously or with an established time interval.

- Satellite orbit position prediction based on accumulated data from the best pointing positions over time.

All the techniques presented result in approximations of the best pointing position, and it is not possible to measure or verify the actual position where the satellite is. Since it is not possible to determine the actual position of the satellite, the pointing error cannot be evaluated because there is no reference to calculate it. Another problem encountered in evaluating the performance of tracking techniques is the non-reproducibility of events in real cases. Assuming the channel that promotes this link is air, atmospheric phenomena cause significant interference in the received signal. These phenomena can be considered stochastic, and finding two same days to test different techniques under the same conditions is virtually impossible.

1.3 Dissertation Synthesis

This work implements satellite tracking techniques in a simulation environment and in a real control system, in order to generate the performance evaluation and compare the techniques in several proposed situations. The satellite tracking technique environment for Reinforcement Learning (RL) software agents is also presented. Chapter 2 explain the satellite tracking scenarios and the techniques described and used in this work. Chapter 3 presents the reinforcement learning approach, describing the construction of the agent and the environment to which it is inserted. Chapter 4 presents the results of the tests done with the tracking techniques in a real system. Chapter 5 presents the conclusions about the work and the next steps within this line of research.

Chapter 2

Satellite Tracking Techniques

Satellite communications provide long-distance transmission capability for applications in isolated places or where no other infrastructure can be used. With these new possibilities and due to the satellite communications characteristics, many challenges arise in maintaining the position and signal strength of the satellite. These challenges show up as characteristics in the transmission itself, such as channel, atmospheric and astronomical events, the type of satellite and its current state, the receiving station position, and antenna installation. All of these factors influence the maintenance of signal reception, and for this reason, satellite tracking techniques are essential in this scenario.

Satellite tracking techniques are implemented in control systems to promote antenna movement. The implemented technique's goal is to promote the tracking as efficiently as possible, within the limitations and compromises that the antenna installation offers. This work evaluates the efficiency of tracking through three aspects:

- Reaching the maximum satellite signal strength received or the pointing position with the smallest error. The tracking techniques have the same objective since the signal strength is directly proportional to the proximity of the antenna pointing direction to the satellite position.

- Perform satellite tracking in the shortest time possible. For searches performed within time intervals, a satellite is considered stopped when the search duration time is less than the time interval. For continuous searches, pointing is continuously moved depending on the supervised signal characteristics.

- Make minimum movement and spend as little energy as possible while repositioning the antenna. Energy expenditure is related to motor activation and applied power. The mechanical wear caused on the antenna is related to the number of movements, which may increase or decrease the interval between maintenance.

Efficiency varies depending on the technique implemented, which varies with the financial costs and infrastructure required for antenna and equipment. The scenario is decisive for the possible implemented technique choice and the installed infrastructure. Some techniques require specific infrastructure, and even if they are more efficient compared to other, techniques that require less infrastructure can be implemented without significant losses. This chapter discusses satellite tracking techniques and the implemented environment. Section 2.1 defines the satellite signal reception scenario, Section 2.2 explains the operation of techniques based on models of celestial orbits, Section 2.3 explains the techniques based on characteristics of the received signal, and Section 2.4 explains the prediction models of orbit.

2.1 Reception Scenario

The tracking techniques implemented in antenna control systems aim to find and monitor the position of best reception of the satellite signal, maintaining an acceptable signal level for the application. Depending on the application and type of tracked satellite, the best signal reception may vary over time, and in this situation, the antenna pointing correction is essential. The comprehension of the scenario is necessary for the implementation of satellite tracking techniques in the systems to ensure their operation in the real environment. Simulation environments are built based on the propagation of antennas and celestial orbits studies.

After understanding the scenario, the fundamental aspects of the problem that will be used to model the tracking techniques are defined, such as the antenna movement space, the satellite received signal and the agents that influence the signal reception. Subsection 2.1.1 explains the behavior of satellites in geosynchronous orbit, Subsection 2.1.2 presents the characteristics of satellite signal reception by the antenna, and Subsection 2.1.4 defines the operation of control systems and provides practical examples.

2.1.1 Satellite Orbit

The satellites that are the target of tracking in this work are the inclined orbit geostationary satellites. In celestial mechanics, an inclined orbit is any orbit whose orbital plane does not coincide with the reference plane. In the tracking literature, the term inclined orbit is used for satellites that do not have strictly geostationary orbit, having deviations in its orbit from its planetary observer referential. Geostationary satellites suffer various disturbances in their orbit, caused by gravitational effects of celestial bodies. These disturbances are corrected by the satellite's movements, moving to the correct position of its orbit. The problem is that the fuel needed for satellite correction runs out over time, making satellite orbit maintenance impossible.

The company or agency responsible for the satellite establishes an useful life for the satellite, which is calculated based on the amount of fuel it will have, among other factors. When the satellite's life is over, the position can no longer be corrected, and the satellite's orbit is disturbed. Although it no longer has strictly geostationary orbit, the satellite is still fully operational for telecommunications. Satellite tracking techniques make it possible to extend the life of the satellite by changing the antenna pointing direction instead of correcting the position of the satellite. There are several existing satellite tracking techniques, and this chapter will expose some techniques developed and adapted during the project.

According to [6], the satellite movement on azimuth M_{az} and elevation M_{el} axes in inclined orbit is described by the equation

$$M_{az} = a_1 t + a_2 sen(f_{az} t + p_{az}) + S_{az},$$
$$M_{el} = e_1 t + e_2 sen(f_{el} t + p_{el}) + S_{el}. \tag{2.1}$$

The constants a_1 and e_1 represent the linear coefficient of displacement on the route, a_2 and e_2 represent the amplitudes, f_{az} and f_{el} the frequencies, p_{az} and p_{el} the phases of the sine functions governing the characteristic variation of inclined orbit satellites and S_{az} and S_{el} represent the initial tracking position of the satellite on each axis.

2.1.2 Satellite Signal Reception

Signal reception is defined from antenna construction and installation characteristics. These characteristics influence the antenna search space and which satellites can be located for continuous signal reception and tracking. The search space S composed by the x and y axes (azimuth and elevation, respectively) can be described by

$$S = \{(x, y) \in \mathbb{R}^2 \mid \forall [0 \leq (x, y) < 2\pi], (x, y) \in S\}. \tag{2.2}$$

The antenna movement in the axes is delimited by its physical limits, characteristic of its construction and installation, and these boundaries are measured in degrees. The space L delimited by the configured physical limits: x_a and x_b for azimuth, y_a and y_b for elevation is valid only if it is contained in S:

$$L \subset S \leftrightarrow (x_a, y_a, x_b, y_b) \in S. \tag{2.3}$$

Thus, the space L is a subset of S and can also be described by x and y and be in the range formed by the physical limits as:

$$L = \{(x, y) \in \mathbb{R}^2 \mid \forall [x_a < x < x_b] \; e \; [y_a < y < y_b], \exists L\}. \tag{2.4}$$

The definition of the search space is used for all functionality involving antenna movement, serving as a reference for positioning and safety measures to prevent unauthorized positions of being reached.

The antenna used in this work has three movement axes (azimuth, elevation, and polarization), although this is considered for tracking purposes only azimuth and elevation axes. After being correctly positioned, the variation in the polarization axis is not sufficient to significantly interfere with signal reception [6]. For this reason, this axis may be disregarded for tracking purposes, and the search space may be formed by azimuth and elevation.

2.1.3 Beacon Signal

The beacon of a satellite is a reference signal to determine its position. The closer the antenna pointing is to the satellite, the stronger is the signal coming from the receiver. The peak represents the position of maximum reception of the signal and the satellite is located, this position must be contained in L for pointing and tracking to be possible.

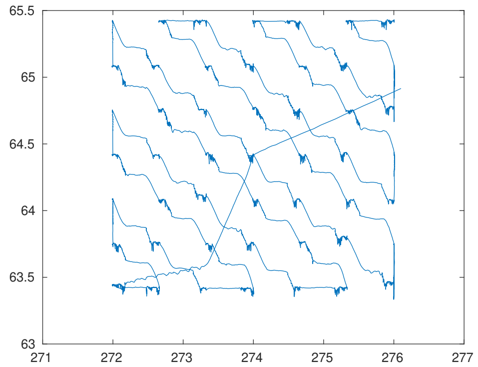
An function b is defined as the beacon signal strength, given in dBm, read in the system, and is dependent on the antenna position, varying its value as a function of azimuth, elevation, and polarization axes (For tracking purposes only azimuth and elevation). According to [6], the beacon signal has the behavior of a second-order function in the azimuth and elevation axes:

$$b(x, y) = C_x(x - P_x)^2 + C_y(y - P_y)^2 + I, \tag{2.5}$$

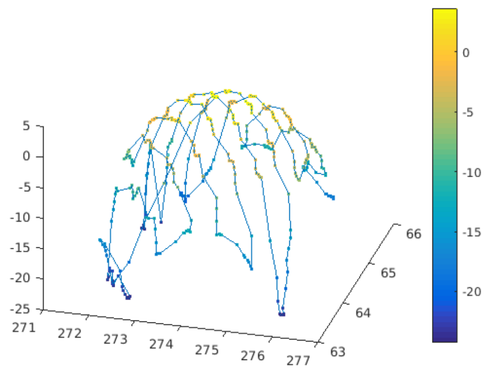
P_x and P_y are the satellite coordinates within the space L, C_x and C_y are the scalars of the functions, and I is the signal attenuation. A test was done using a real antenna to validate the model presented, scanning the antenna around a point to receive the beacon signal from a satellite, moving the antenna in a zigzag pattern delimiting the movement within a $4° \times 2°$

azimuth and elevation rectangle. Figure 2.1a shows the antenna movement during the scan, and Figure 2.1b the beacon scan result in relation to the movement space.

Figure 2.1: Beacon signal characteristics.



(a) Zigzag antenna movement.



(b) Beacon function.

Source: Author.

Although the proposed model is valid and proven with tests on a real system, the presence of noise and interference in signal reception distorts the function value. The noise R present in the signal reading is considered a random variable, representing all interferences in reception. Adding noise to the model we have

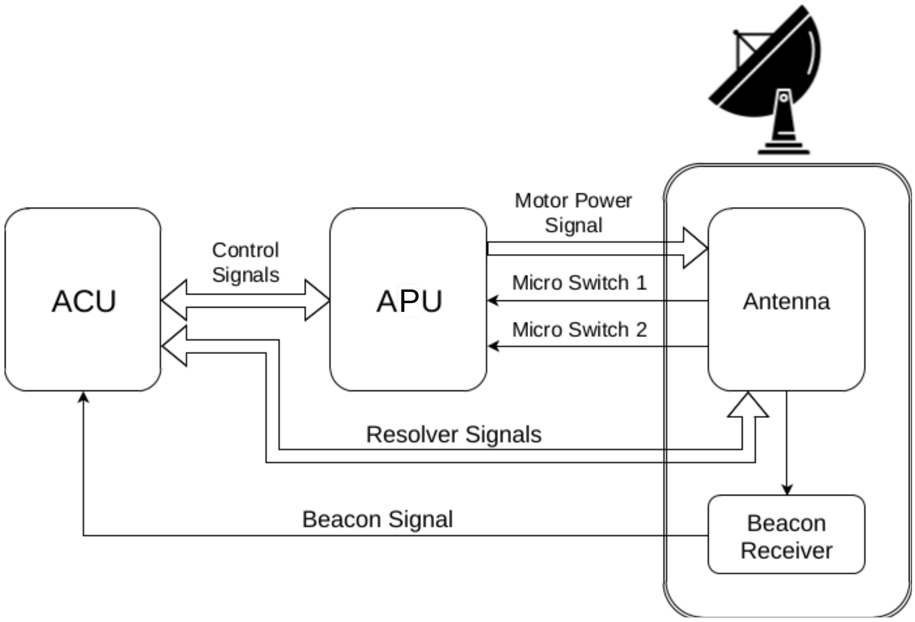
$$b(x, y) = C_x(x - P_x)^2 + C_y(y - P_y)^2 + I + R. \tag{2.6}$$

Interference in signal reception should be taken into account in the development and implementation of tracking techniques, as interference-contaminated signals may cause undesired behavior.

2.1.4 Antenna Control System

Antenna Control Systems are designed to control antennas on the movement axes by controlling the pointing position manually or automatically. These systems are also responsible for supervising satellite signal characteristics, correcting antenna pointing for optimal reception using the tracking techniques developed to control the antenna. The system consists of an Antenna Power Unit (APU) and Antenna Control Unit (ACU) as shown in Figure 2.2.

Figure 2.2: Antenna control system overview.



Source: Author.

The APU is responsible for driving and controlling the electrical components of the system, controlling antenna movement through azimuth, elevation and polarization axis motors. The ACU has the system control function, sending instructions to the APU, and performing available tracking techniques, as well as storing settings and other relevant information. Embedded systems were developed that perform specific activities and promote the communication of the members, transmitting information and instructions that will be used in operations with the antenna. The Antenna Drive Unit (ADU) controls the APU, including connections to the inverters and motors, limit switches and communication interface for connection to the other system unit. The Antenna Sensors Unit (ASU) reads sensors that are used as a reference for signal movement and reception, sends instructions to the ADU through existing connections, reviews system status, and receives instructions from a computer located at ACU.

The system is in charge of controlling the activities the antenna must perform to promote the tracking, which can be done by user supervision or automatically. The main activities are:

Antenna Movement: The system is capable of moving the antenna with the ACU, either manually or automatically.

Save Information: Saved information can be of configuration or operation type. The configuration information is the system's operational reference. The operating information saves satellite positions and parameters for the tracking techniques that will be used at the time of their execution.

Sensor Reading: The antenna has several sensors used as a reference for signal movement and reception. The ASU and ADU have the necessary interfaces to read the sensors, sharing the extracted data through the connections between them.

Safety Alerts: When emergency events such as the antenna reaching physical movement limits or the beacon signal dropping in intensity occur, the system shall alert the user and interrupt all functions currently being performed.

Satellite Tracking: Several tracking techniques have been developed and implemented to keep the antenna in the position of receiving a satellite signal. These techniques will be covered in sections 2.2, 2.3 and 2.4.

The control system is developed at the Federal University of Pará (UFPA) by the Telecommunications, Automation and Electronics Research and Development Center (LASSE). The antenna shown in the Figure 2.3 is installed in the Guamá Science and Technology Park (PCT Guamá) and it is part of a project in partnership with Brasilsat Harald. The system [7] is the reference for describing control systems and is used for the tests and implementations presented below.

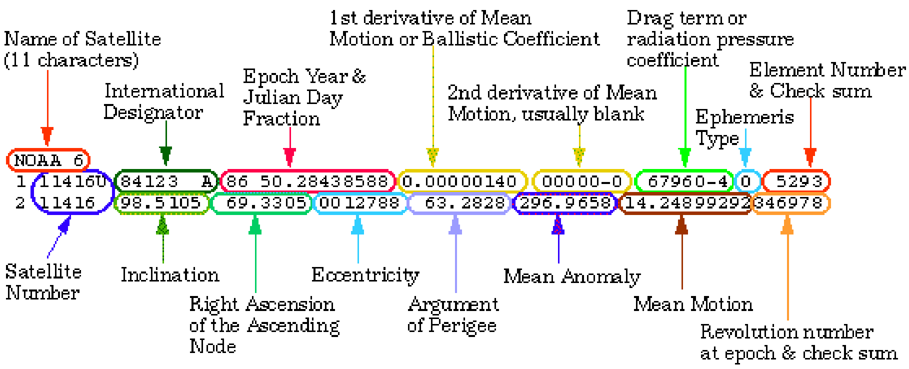
Figure 2.3: 3 axes antenna installed at PCT Guamá.

Source: Author.

2.2 Orbital Parameter Tracking

Mathematical models based on celestial mechanics studies are used to calculate the orbit of satellites, maintained by North American Aerospace Defense Command (NORAD) and National Aeronautics and Space Administration (NASA). They are called Simplified perturbations models, a group of mathematical models described in [8] (SGP, SGP4, SDP4, SGP8, and SDP8) and used to calculate orbital state vectors (which consist of position, velocity and time) of satellites relative to the Earth-centered inertial coordinate system. These models have as input the Two-line Elements (TLE), which is a data format to store a list of orbital elements of an object that orbits the Earth given an instant of time. The TLE is defined in [9], and the Figure 2.4 exemplifies the structure that it presents for an example satellite.

Figure 2.4: Definition of Two-line Element Set Coordinate System.



Source: NASA [9].

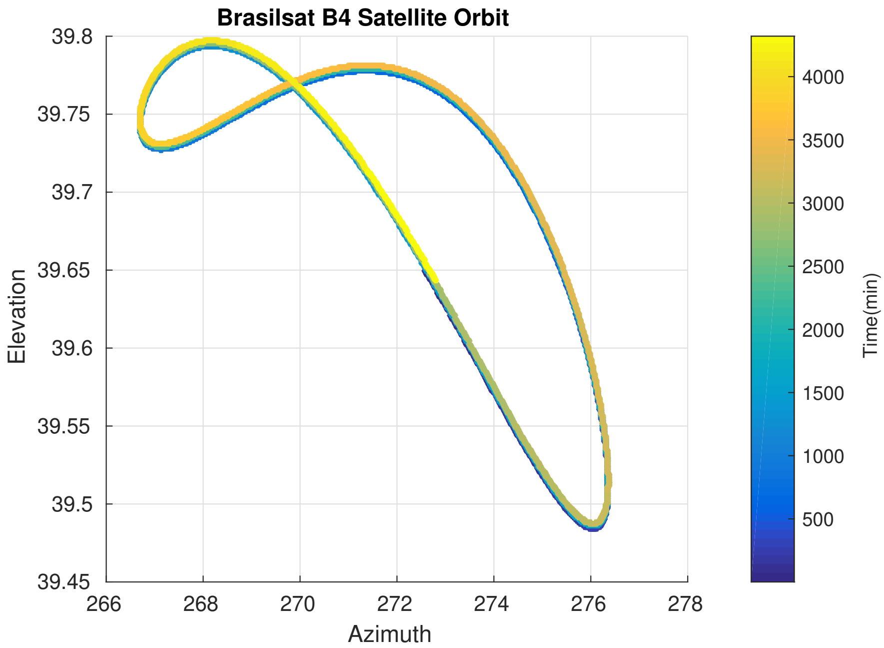
Using one of the models combined with the TLE, a table with the satellite orbit positions is obtained. The Figure 2.5 represents the satellite orbit obtained using an algorithm that implements SGP4 and TLE from satellite Brazilsat B4.

The resulting table is used to position the antenna pointing to receive the satellite signal over time, with the interval defined in the model.

2.3 Tracking by Signal Characteristics

Search algorithms use characteristics of the satellite beacon signal to be traced as a reference to find the best receiving antenna position, where reception quality is directly related to

Figure 2.5: Brasilsat B4 satellite orbit.



Source: Author.

signal strength. The search techniques that are applied to the algorithms alter their performance in maintaining signal strength, the number of movements and energy efficiency, optimizing for better reception and position correction time.

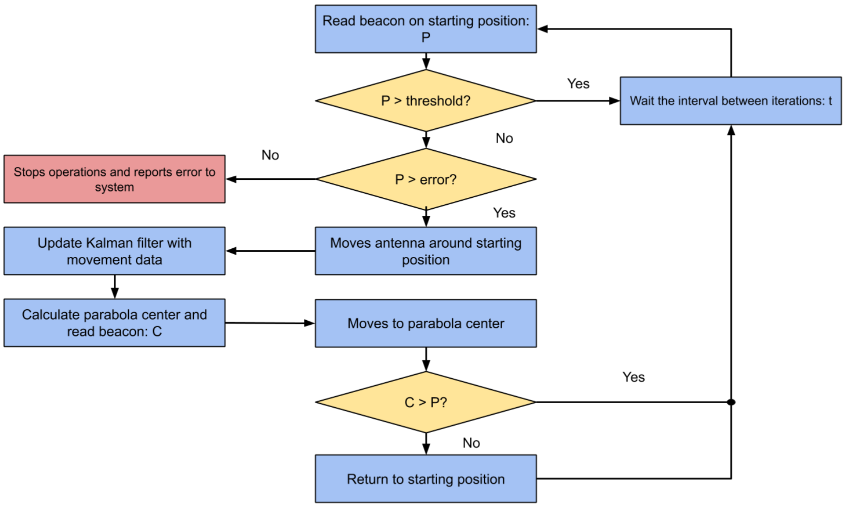
2.3.1 Kalman Track

The Kalman Track algorithm [10] relies on the model described in 2.1.3 for maximum beacon signal reception. The Kalman filter estimates the coefficients of the (2.6) function and thus obtains a possible center of the parabola that corresponds to the position of maximum intensity. This filter is fed with the antenna movement data around the satellite location region, relating the position of the azimuth and elevation axes with the beacon intensity.

2.3.1.1 Algorithm Operation

The algorithm works by calculating the function coefficients that represent the beacon signal along with the search space. The Figure 2.6 shows the flowchart of the Kalman track algorithm in execution.

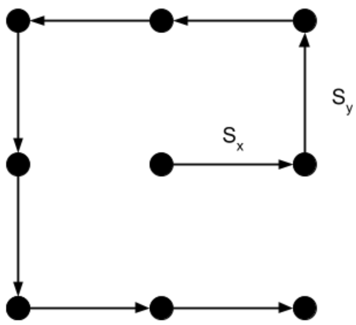
Figure 2.6: Kalman Track execution flowchart.



Source: Author.

These calculations are repeated at each interaction of the algorithm in the system, updating the antenna position based on them. The movement is based on circumventing the current position by recording the position of the axes with the corresponding beacon value. The outline is made in a rectangular shape, as shown in Figure 2.7.

Figure 2.7: Kalman Track movement.



Source: Author.

The S_X and S_Y parameters form the contour rectangle used in the center search. The Kalman filter is fed with records that relate the position of the axes to the beacon intensity value at that location, estimates the position of the center of the parabola based on the data and mathematical model expressed in (2.6), and moves the antenna to the position. After the first algorithm execution, it is already possible to estimate the position of maximum reception at that moment, and this position will be updated with each interaction of the algorithm.

2.3.1.2 Algorithm Configuration

The Kalman Track search algorithm has operating parameters that are configured for the best satellite tracking performance to be tracked. Table 2.1 shows the configurable parameters of the Kalman Track.

Table 2.1: Kalman Track parameters.

Parameter	Type	Description
S_x	Value in degrees	Step moving the azimuth axis
S_y	Value in degrees	Step moving the elevation axis
timer	Time in minutes	Time interval between algorithm execution
reset	On or Off	Reset Kalman filter with each algorithm execution

Source: Author.

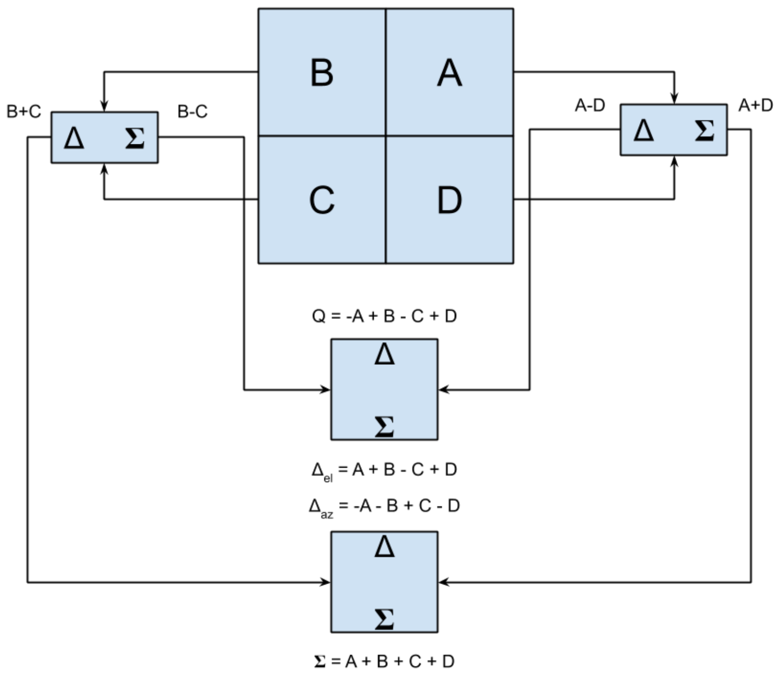
The S_x and S_y parameters form the contour rectangle used in the center search. If the set values are too small, the Kalman filter may not learn enough to estimate the center of the parabola correctly. If the set values are too large and generate unnecessary movement, they can lead to wrong learning about the area covered, invalidating the data generated during the travel time. The timer is the time interval between the algorithm search executions, and this value must be set according to the satellite orbit movement. The further the satellite moves, the shorter the time interval must be, as the satellite can move sufficiently between runs to cause loss of signal and tracking route. The filter reset parameter is set according to the satellite orbit. The best position estimation has errors when the movement between interactions is significant, and the filter reset is off. Because the filter tends to generate results close to the previous position, it was observed during the actual tests that resetting the Kalman filter before the search movement generates better estimation performance when the large satellite moves in its orbit.

2.3.2 Satellite Tracking with Monopulse

A monopulse antenna is a tracking radar method. The term ”monopulse” signifies that with a single pulse, the antenna can infer angle data. Monopulse radar techniques can be used passively, when the antenna only receives the beacon signal and uses it for positioning.

The monopulse uses four antennas (or quadrants of a single antenna). They can be horns or sections of a flat plate array of radiators, or even subarrays of an active electrically scanned antenna (AESA) phased array. The elements are all steered together mechanically (on a gimbal) or electrically (using phase shifters in the case of AESA). The target is illuminated by all four quadrants equally. A comparator network is used to ”calculate” four return signals, as shown in Figure 2.8.

Figure 2.8: Monopulse Feedhorn.



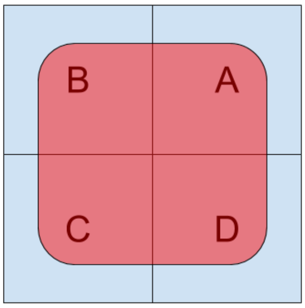
Source: Author.

The sum signal has the same pattern in receive as transmit, a broad beam with highest gain at boresight; the sum signal is used to track target distance and perhaps velocity. The calculation of the signal is given by

$$\Sigma = A + B + C + D, \tag{2.7}$$

shown in the Figure 2.9.

Figure 2.9: Monopulse sum signal.



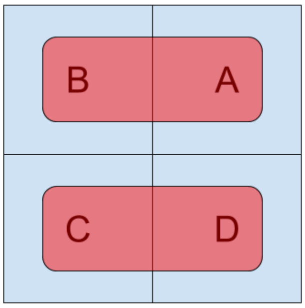
Source: Author.

The elevation difference signal is formed by subtracting the two upper quadrants from the two lower quadrants and it is used to calculate the target's position relative to the horizon. The calculation of the signal is given by

$$\Delta_{el} = (A + B) - (C + D), \tag{2.8}$$

shown in the Figure 2.10.

Figure 2.10: Monopulse elevation difference signal.



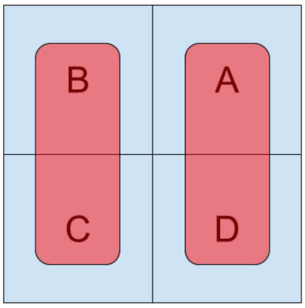
Source: Author.

The azimuth difference signal is formed by subtracting the left quadrants from the right quadrants and is used to calculate the target's position to the left or right. The calculation of the signal is given by

$$\Delta_{az} = (A + D) - (B + C), \tag{2.9}$$

shown in the Figure 2.11.

Figure 2.11: Monopulse azimuth difference signal.



Source: Author.

A fourth signal, called “Q difference” is the diagonal difference of the quadrants. This signal is often not used, so the typical monopulse receiver needs only three channels. Sometimes only a two-channel receiver is used, as the two differences signals are multiplexed into one with a switching arrangement.

After the sum signal generation and the difference signals on each axis are calculated, they are used to infer the best signal reception position. The pointing is corrected from the processing of the obtained signals, and the unit responsible for this processing within a monopulse tracking system is the monopulse processor. The monopulse processor is a portion of a monopulse radar that operates on the voltages derived from the simultaneous antenna patterns to produce the monopulse outputs.

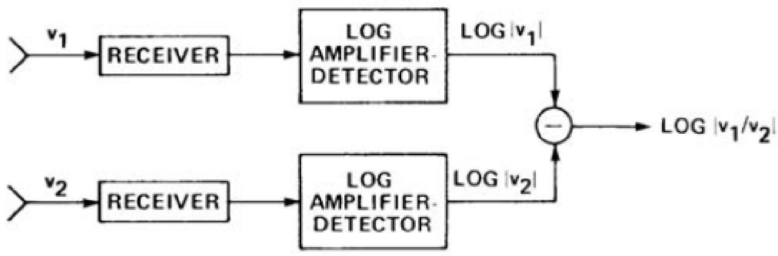
Every processor has a different commitment to its implementation, from the robustness and accuracy to the ease of implementation and the associated cost. As in [11], tracking techniques used on satellite link antennas are presented, and it is exposed that monopulse systems are more expensive compared to other techniques. In Chapter 8 of [12], the processors are described, and one of them is detailed in 2.3.2.1. The monopulse processors chosen are implemented using GNU Radio and Universal Software Radio Peripheral (USRP).

The universal software radio peripheral (USRP) [13] Software Defined Radio Device is a tunable transceiver for designing, prototyping, and deploying radio communication systems. It is an ideal product to prototype wireless communications, develop EW (Electronic Warfare) and SIGINT (Signal Intelligence) applications, and deploy wireless systems. GNU Radio [14] is a free and open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in research, industry, academia, government, and hobbyist environments to support both wireless communications research and real-world radio systems. The GNU Radio software provides the framework and tools to build and run software radio or just general signal-processing applications. The GNU Radio applications themselves are generally known as "flowgraphs", which are a series of signal processing blocks connected together, thus describing a data flow.

2.3.2.1 Processor Using Log $|v_1|$ and $|v_2|$

In this type of monopulse processing the component voltage $v1$ and $v2$ go to receivers with logarithmic amplifier-detectors. In place of $v1$ and $v2$, the equivalent voltages $s + d$ and $s - d$ may be used. The logarithmic outputs are subtracted to produce the monopulse output, as shown in Figure 2.12.

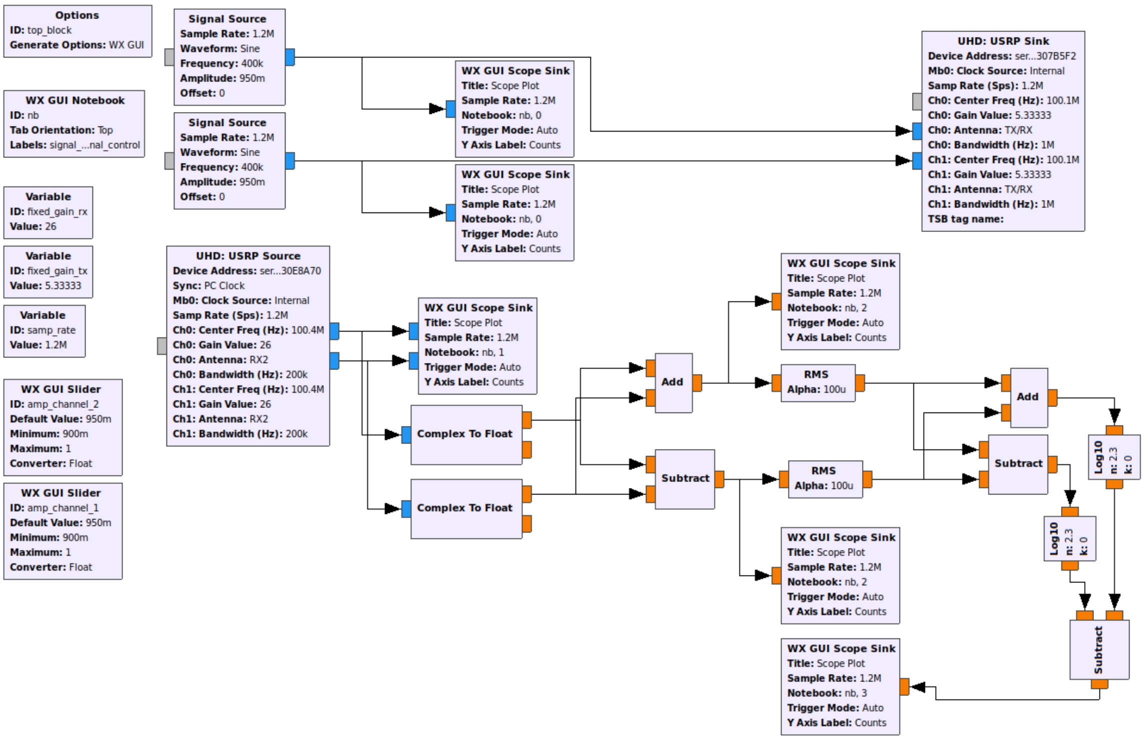
Figure 2.12: Monopulse logarithmic processor.



Source: Monopulse principles and techniques [12].

The Figure 2.13 shows the flowchart of the processor implemented in software using GNU Radio.

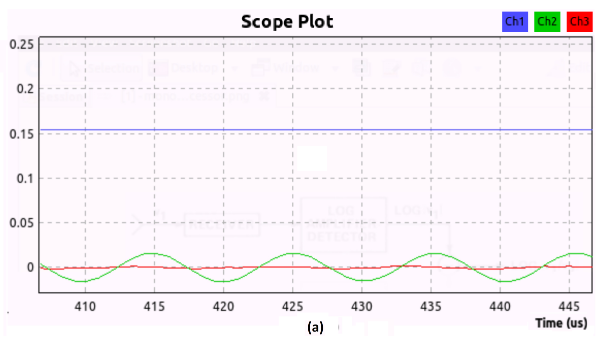
Figure 2.13: Monopulse processor GNU radio flowchart.



Source: Author.

Figure 2.14 shows the results obtained with the implementation in USRP. These signals are used by the antenna control system to move the pointing to the satellite.

Figure 2.14: Sum, difference, and control signals.



Source: Author.

The green and red channel respectively represent the sum and difference signals of the tracked axis, and the blue signal is the result generated by the implemented monopulse processor.

2.4 Orbit Prediction Techniques

Satellites that have considerable tracking movement have known orbits, as seen in Section 2.1. These orbits can be obtained by calculations using satellite parameters or by observing their behavior over time. The orbit prediction algorithms predict the satellite position based on the observation of its movement. After the learning period of the algorithm, no knowledge of beacon signal characteristics or satellite orbital parameters is required for tracking. The orbit prediction algorithm, called Predictive Track, can learn the route of the satellite being tracked by observing the maximum reception position of the beacon signal over a period using Kalman filters. After the learning period, the algorithm is able to estimate the best position for any given time making use of only the collected data.

This section describes the development of the Orbit Prediction Algorithm, its operation, and the use of Kalman filters to learn the satellite orbit. The Predictive Track was implemented in the system described in the 2.4.1, in conjunction with the technique described in 2.4.2.

2.4.1 Algorithm Operation

The developed orbit prediction algorithm uses the best position data found in successive Kalman Track runs, seen in Section 2.3.1, to feed the Kalman filter inputs that will learn and predict the orbit of the tracked satellite. Predictive Track operation can be abstracted in three primary operating states: two learning states, and one orbit prediction. The first state is short-term learning, the second state is long-term learning, and the third state is the orbital prediction:

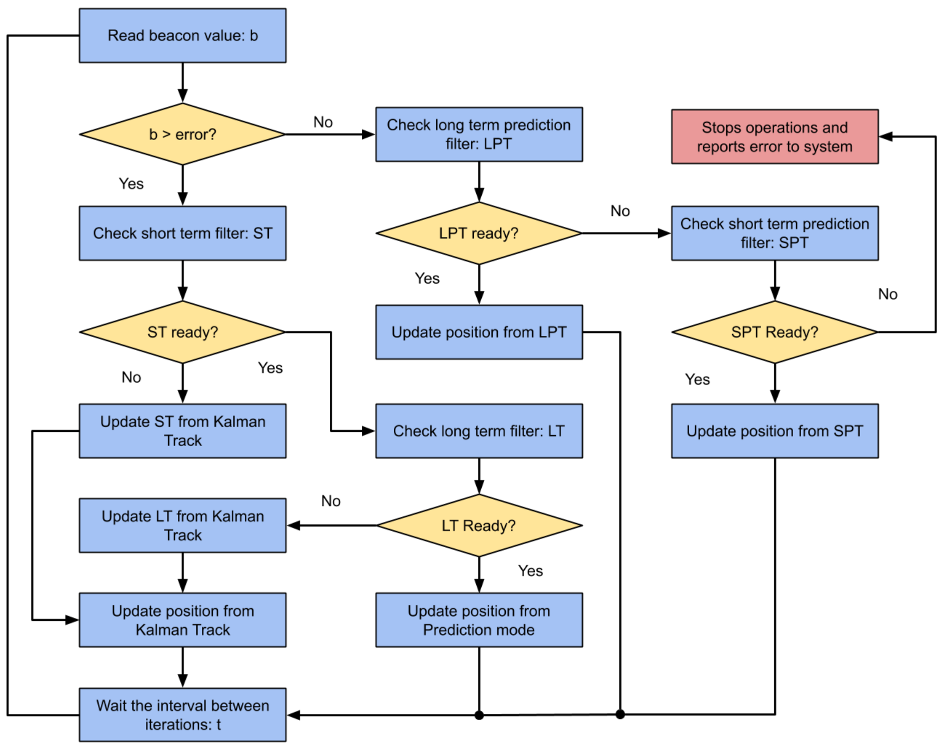
Short-term learning: The first state uses Kalman Track to generate the best position data from successive executions with a defined time interval. These generated data are used as input to the prediction filter responsible for short term predictions, training the filter to predict position based on the current time instant only. After short-term learning is completed, the system identifies that the algorithm can move to the next state.

Long-Term Learning: The second state uses Kalman Track's short-term learning and generated data to train the prediction filter responsible for making long-term predictions. At first, the information generated in the short term prediction filter is carried over to the long term filter, then new data is generated from new Kalman Track runs.

Orbit Prediction: The third state uses the prediction filter that was trained in the long-term learning phase to update the antenna position with the predictions generated by the filter. With each prediction, what was learned is lost, and the validation of the prediction may expire after some time. When the validity of predictions expires, the algorithm changes the state to long-term learning to re-teach the prediction filter.

The transition between states is caused by the learning rates of Kalman filters and the beacon signal strength of the tracked satellite. The flowchart of the Predictive Track algorithm is represented in Figure 2.15.

Figure 2.15: Predictive Track execution flowchart.



Source: Author.

2.4.2 Structure and Learning

The algorithm records the best positions found by running Kalman Track, and these positions serve as input to feed the prediction filters in the learning phase. Prediction filters are composed of two Kalman filters, one for each axis used in the tracking. A prediction filter response is the composition of filter responses forming coordinates in the azimuth and elevation space. Increased data samples generated improves the predictions of the filters, resulting in better tracking accuracy. The data generated by the search algorithm consist of coordinates of the axis position and the time they were obtained, as shown in Table 2.2.

Table 2.2: Sample structure.

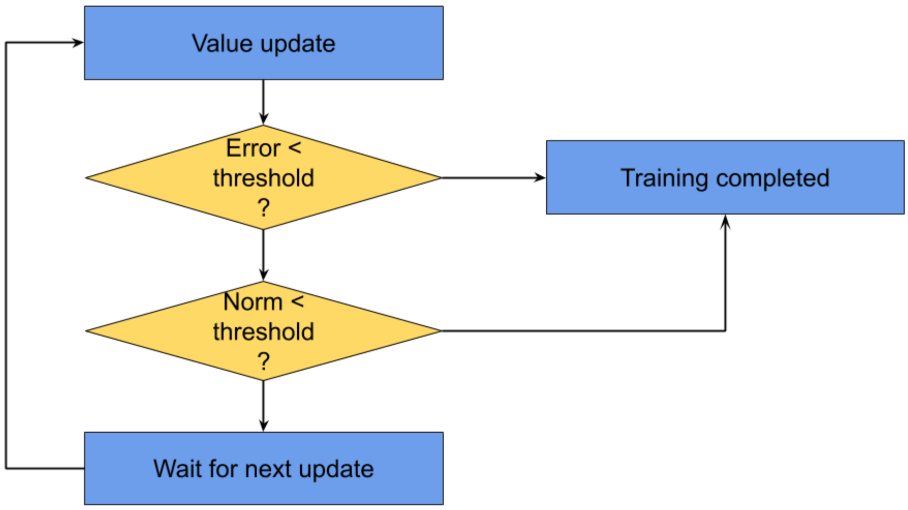
Sample	Azimuth	Elevation	Time
0	x_0	y_0	t_0
1	x_1	y_1	$t_0 + t$
2	x_2	y_2	$t_0 + 2t$
		$\dots$	
n	x_n	y_n	$t_0 + nt$

Source: Author.

The time interval t represents the time between interactions, configured by the user. The Kalman filter's input are the respective Azimuth and Elevation values at each search algorithm interaction.

Although it is not used for position updating in the training phase, the current state prediction generated in the learning step is used to measure filter learning. Learning assessment is based on Kalman filter covariance matrix norms and the Euclidean distance from the prediction position as indicated by the Kalman Track. The learning phase structure that tests Kalman filters is valid for the short and long term prediction filters, shown in Figure 2.16.

Figure 2.16: Kalman filter learning flowchart.



Source: Author.

In the prediction stage, the algorithm only evaluates the norm of the arrays, since the antenna position is updated by the prediction filter, instead of the search algorithm.

Chapter 3

Reinforcement Learning applied to Satellite Tracking

Reinforcement learning (RL) is a computational approach to understanding and automating goal-directed learning and decision-making. It is distinguished from other computational approaches by its emphasis on learning by an agent from direct interaction with its environment, without relying on exemplary super-vision or complete models of the environment. According to [15], reinforcement learning is the first field to seriously address the computational issues that arise when learning from interaction with an environment in order to achieve long-term goals. Reinforcement learning uses a formal framework defining the interaction between a learning agent and its environment in terms of states, actions, and rewards. This framework is intended to be a simple way of representing essential features of the artificial intelligence problem. These features include a sense of cause and effect, a sense of uncertainty and nondeterminism, and the existence of explicit goals.

Deep Learning (DL) is defined as a sub-field of machine learning (ML) that is based on learning several levels of representations, corresponding to a hierarchy of features or factors or concepts, where higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts. Deep learning is part of a broader family of machine learning methods based on learning representations [16]. Approaches to implementing Deep Reinforcement Learning in continuous space problems are presented in [17]. The antenna pointing position can assume any angle value, limited by its mechanical construction and the number of axes of freedom it has, which makes it hard and laborious to define fixed states that an agent RL can assume. Therefore, the approach of continuous space satellite tracking is necessary.

This chapter describes the development and implementation of a satellite tracking algorithm in a simulation environment using RL and DL methods. The characteristics of the simulation environment in which the agent interacts are presented in Section 3.1, and the characteristics of the Deep Q Network (DQN) agent are presented in Section 3.2.

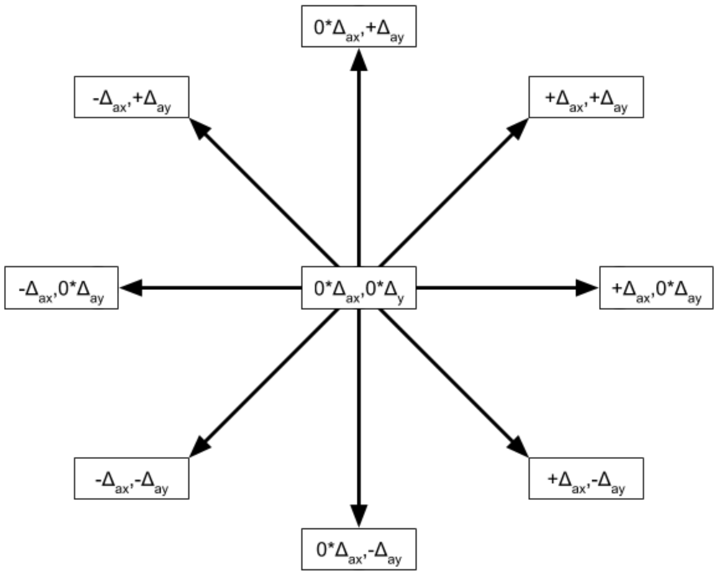
3.1 Environment

The simulation environment is developed based on the actual tracking environment and the operation of RL algorithms. Python programming language was used in conjunction with the OpenAI Gym library [18] to implement the simulation environment. Gym is a toolkit for developing and comparing reinforcement learning algorithms, providing environments already deployed in the library, and offering the option to create custom environments. In Subsection 3.1.1 is described how the antenna pointing movement is updated, in Subsection 3.1.2 is shown how the beacon value is calculated, and in Subsection 3.1.3 the rewards policy is explained.

3.1.1 Antenna Movement

The software agent moves the antenna pointing direction, changing the acceleration within time intervals. The possible action space in the environment is made up of acceleration values that are added to the current acceleration, updating this value. The agent chooses one of the possible actions within the action space that the environment has. The agent promotes the tracking on one or two axes, and this changes the action space: 3 actions when using only one axis and 9 actions when using two axes, as shown in the Figure 3.1.

Figure 3.1: Software agent action space.



Source: Author.

The vector A contains the space of actions, each cell position corresponds to a possible action, and the cell content corresponds to the acceleration variation value. For a drive axis, the value of A is updated with the vector $A = A_{X1}$, and A_{X1} is

$$A_{X1} = [-\Delta a, 0 * \Delta a, \Delta a]. \tag{3.1}$$

For two axes of motion, A is updated with the matrix A_{X2}. Each cell of matrix has the value of Δa_x and Δa_y for each axis of motion, the value of matrix A_{X2} is

$$A_{X2} = \begin{bmatrix} (-\Delta a_x, +\Delta a_y) & (0 * \Delta a_x, +\Delta a_y) & (+\Delta a_x, +\Delta a_y) \\ (-\Delta a_x, 0 * \Delta a_y) & (0 * \Delta a_x, 0 * \Delta a_y) & (+\Delta a_x, 0 * \Delta a_y) \\ (-\Delta a_x, -\Delta a_y) & (0 * \Delta a_x, -\Delta a_y) & (+\Delta a_x, -\Delta a_y). \end{bmatrix} \tag{3.2}$$

The movement is updated individually on each axis with the acceleration variation of the double value contained in the cell. The algorithm 1 exemplifies how the position value is updated from the agent's chosen action.

Algorithm 1: Simulation movement values update.

Data: Software Agent action

Result: New movement values (position, speed and acceleration)

```
Initialization of action space vector A and Δt;
Read current antenna pointing position s and s_0;
Initialization of movement values a = 0, v = 0, v_0 = 0;
while not at end of the simulation do
    Read new agent action act;
    Update acceleration value a = a + A(act);
    Update speed value v = a + Δt + v_0;
    Update position value s = v + Δt + s_0;
    if Position s out of limits then
        Inform the error to simulation;
    else
        Return a,v and s values to simulation;
    end
end
```

The agent enters the environment from the input act. After the action is chosen, the environment updates the antenna pointing movement variables a, v and s. The agent action updates the acceleration value, the acceleration updates the speed value, and the speed updates the space value. Velocity and acceleration are the first and second derivatives of space with respect to time, respectively. Instantaneous velocity v is calculated from the discrete integration of the instantaneous acceleration a value seen in

$$v = a * \Delta t + v_0, \tag{3.3}$$

Δt represents the time interval between each iteration during the episode, and v_0 represents the initial velocity value. The discrete integration of velocity results in the antenna pointing position s at that moment demonstrated in

$$s = v * \Delta t + s_0. \tag{3.4}$$

Time interval Δt is the same as that used for speed calculation, and s_0 is the initial position of the antenna pointing. The initial position value is set at the beginning of the simulation.

3.1.2 Beacon Signal

With the value of the antenna pointing position on the movement axis, it is possible to obtain the corresponding beacon signal strength value. This value is based on the mathematical model presented in Section 2.1 and is adapted for one or two axes tracking. With the value of the antenna pointing position on the movement axis, it is possible to obtain the corresponding beacon signal strength value $b(x)$. For an axis, the model is

$$b(x) = k * (x - p)^2 + L, \tag{3.5}$$

k is the parabolic curvature coefficient, p is the center position, and L is the attenuation of the signal. For two axes the beacon $b(x, y)$ model is

$$b(x, y) = k_x * (x - p)^2 + k_y * (y - q)^2 + L_{xy}, \tag{3.6}$$

k_x and k_y are the parabola coefficients, p and q correspond to the 2D position of the center of the parabola, and L_{xy} is the attenuation in the signal. The value of the beacon signal is calculated with each new antenna pointing position reached.

3.1.3 Rewards

The rewards of the developed environment are based on the observation variables that the agent has available, and the maximum beacon signal position, which is the goal. The metrics used to reward the agent decisions and how this value is updated are described in this Subsection. The Directional evaluation metric is presented in 3.1.3.1, the Beacon value metric are presented in 3.1.3.2, and the reward update for the agent is presented in 3.1.3.3.

3.1.3.1 Directional Evaluation Metric

The metric that uses the move direction to reward the agent is called Directional evaluation. This metric uses the velocity vector and the action chosen by the agent to evaluate if the antenna pointing movement is going in the right direction, the center of the parabola. The

Algorithm 2 describes the operation of the metric for a movement axis.

Algorithm 2: Directional evaluation reward.

Data: The action chosen act, the parabola center p.

Result: Reward value

Read current pointing position s;
Read current velocity value v;
Set reward value $reward = 0$;
Initialization maximum and minimum awards values max, min;
if $s < p$ **then**
  if $v > 0$ **then**
    Update reward value $reward = reward + max$;
  end
  if $act = 3$ **then**
    Update reward value $reward = reward + min$;
  end
else
  if $v < 0$ **then**
    Update reward value $reward = reward + max$;
  end
  if $act = 1$ **then**
    Update reward value $reward = reward + max$;
  end
end
return reward

The highest score goes to the velocity vector in the right direction since it represents the pointing approaching the center of the parabola. The lowest score goes to the choice of actions because the agent's goal is to stay on top and he will need to use the action in the opposite direction of the velocity vector to slow down the pointing speed.

3.1.3.2 Beacon Value Metric

The reward metric implemented to recompense the agent from the beacon value of his position is named Beacon evaluation. The beacon value is used because it is directly proportional to the proximity to the center of the parabola. The obtained beacon-related agent reward is given in

$$\text{reward} = \text{reward} + c * (\text{trunc}(b) - 5), \tag{3.7}$$

b is the value of beacon, and c is a coefficient for scaling the reward value. The $trunc$ function removes the fractional part of the number, creating fixed and scaled values for possible beacon values.

3.1.3.3 Reward Update

After the function execution $metrics()$, the value obtained in the exposed metrics (Beacon and Directional evaluation) is added to the total value accumulated acc until the simulation end or until the operation is stopped. This approach presented the best results during the software agent tests. The algorithm 3 shows how the environment gives rewards to the software agent.

There are three conditions for stopping the simulation, and each situation has an associated reward value:

Time limit: The first is when the simulation timeout is reached. When this occurs, the agent is rewarded with the accumulated reward value;

Out of limits: The second is when the antenna pointing assumes a position outside the established limits. When this occurs, the agent is punished with the remaining time in the simulation;

Successful finish: The third is when the pointing reaches the maximum reception position of the signal. This is the goal of the agent, and he is rewarded with the value accumulated during the simulation plus the simulation time remaining.

The first two situations can produce bad results, but the agent is encouraged to stand in a position within the established limits rather than leaving the satellite pointing, which in the real case would make communication impossible. The duration time is used to reward the agent when the goal is achieved by representing one of the goals of the tracking techniques, achieving the best antenna pointing in the shortest time. Time is also used to punish the agent when he goes out of bounds, to discourage him from going out of bounds in less time so that he receives less punishment.

Algorithm 3: Reward Update.

Data: Movements variables

Result: Reward value

Read current pointing position s;
Read current velocity value v;
Read current acceleration value v;
Set reward value $r = 0$;
Set accumulate value $acc = 0$;
Set stopping condition $stop = False$;
while *not at end of the simulation* **do**
 Apply metrics to update the accumulate value $acc = metrics()$;
 if *at end of the simulation* **then**
  Update the reward value with accumulate $r = r + acc$;
  Stop simulation $stop = True$;
 else
  if *beacon below the limit* **then**
   Punish the agent with the remaining time $r = -(limit - time)$;
   Stop simulation $stop = True$;
  else
   if *antenna pointing into valid position* **then**
    if *velocity into valid value* **then**
     if *acceleration into valid value* **then**
      Reward the agent with the remaining time $r = (limit - time)$;
      Update the reward value with accumulate $r = r + acc$;
      Stop simulation $stop = True$;
     end
    end
   else
    Update the reward value with accumulate $r = r + acc$;
   end
  end
 end
end

3.2 Software Agent

The algorithm used in the simulation RL agent is DQN. In [19] and [20], the concepts of DQN and the operation of these algorithms are presented. The choice of algorithm is made by its generalist ability, and the approach of continuous spaces demonstrated in [17]. Python programming language was used in conjunction with the Keras library [21] to implement the software agent. The Table 3.1 shows the characteristics of the DQN agent.

Table 3.1: Agent characteristics.

variable	value	Description
ϵ	0.05	aka exploration rate, this is the rate in which an agent randomly decides its action rather than prediction.
ϵ decay	0.5	decay of the variable ϵ by use
γ	0.5	aka decay or discount rate, to calculate the future discounted reward.
α	0.001	Determines how much neural net learns in each iteration.
α decay	0.01	decay of the variable ϵ by use
hidden layers	13	number of hidden layers
neurons per layers	25	number of neurons per layers

Source: Author.

The observation variables that are passed to the agent are beacon value, pointing position, velocity, and acceleration. These variables are described in 3.1 and are passed on to the agent to make his decision.

Chapter 4

Test and Results

The satellite tracking techniques described in the chapter are implemented in the antenna control system, and tracking information over time is stored in a file. The file contains the position values in the movement axes and the beacon signal strength. With this information, it is possible to evaluate the performance of the techniques. This chapter describes the tests done on the Predictive Track algorithm using the actual antenna and control system. This algorithm was chosen because it executes the Kalman track during its learning stage, comparing a prediction algorithm with a search algorithm. Section 4.1 explains the real scenario where the technique will be evaluated, and Section 4.2 will show the test results.

4.1 Test Scenario

The techniques are implemented in a real antenna located at the Guamá Science and Technology Park (PCT Guamá) where the Telecommunications, Automation and Electronics Research and Development Center (LASSE) is also located. The antenna is controlled by the developed control system, which implements tracking techniques to track satellite movement over time. The scenario used for testing satellite antenna tracking techniques is described in this section, where 4.1.1 shows the characteristics of the satellite-tracked during the tests and 4.1.2 shows the conditions during the duration days.

4.1.1 Satellite

The satellite tracked during the tests was BRAZILSAT B4, launched on August 16, 2000, and its mission duration estimated at 12 years. Being 19 years old (7 years out of life), the tracked satellite is in an inclined orbit, as shown in Figure 4.1.

Figure 4.1: Orbit of the satellite BRAZILSAT B4 in relation to Belém do Pará.

Source: N2YO.

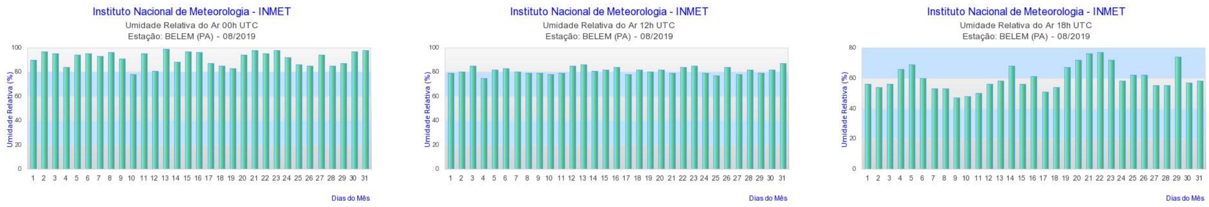
The disturbance caused in its orbit is sufficient for the antenna, stopped at its initial pointing, to completely lose the satellite signal for a day and make any communication impossible.

4.1.2 Climatic Factors

The tests occur between August 20th and 27th and during this period, the screening continued uninterrupted. Using data from the National Institute of Meteorology (INMET), it was possible to obtain the graphs of weather conditions recorded during the test period in Belém (in the State of Pará). This information should be taken into consideration because the weather can impact on signal reception [22] and perhaps in the tracking of satellites. Information on temperature, relative humidity, and rainfall accumulated in Belém do Pará during the month covering the duration of the tests are presented. The objective is to relate weather information to possible disturbances that occur during signal reception to minimize the stochastic effects of real-world testing.

Using meteorological data, an attempt was made to relate temperature, humidity, and precipitation to signal reception. Figure 4.2 shows the humidity information measured at 0 hours (4.2a), 12 hours (4.2b), and 18 hours (4.2c) each day.

Figure 4.2: Relative humidity throughout the day.

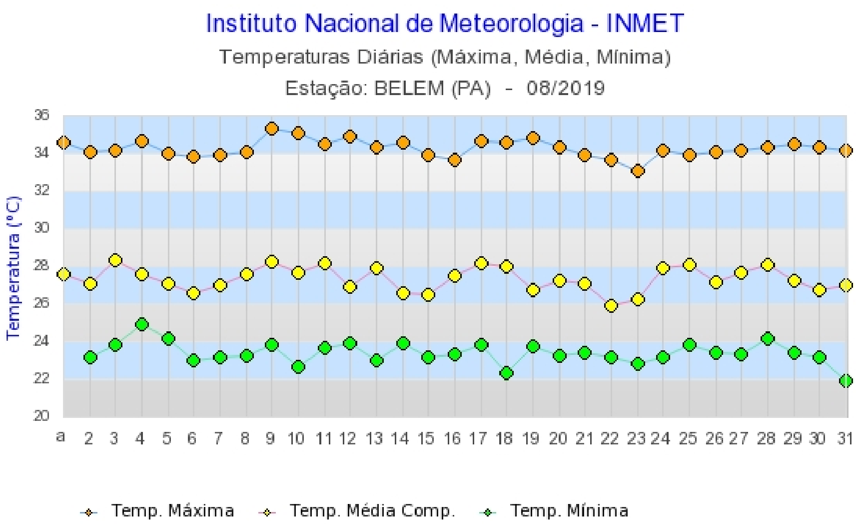


(a) humidity at 0 hours. **(b)** humidity at 12 hours. **(c)** humidity at 18 hours.

Source: National Institute of Meteorology (INMET).

The temperature may affect the channel where the satellite link is established, as well as affect the performance of equipment outdoors. Figure 4.3 shows the average, maximum, and minimum temperature curves over August.

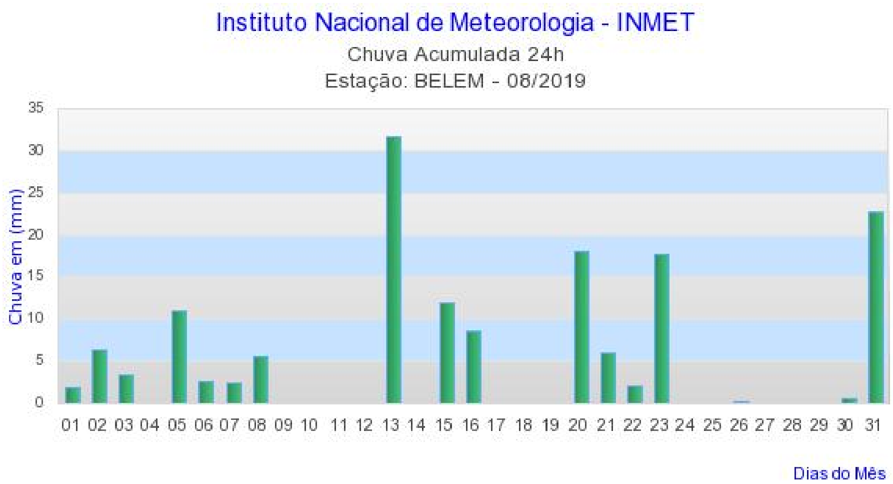
Figure 4.3: August temperature graphs.



Source: National Institute of Meteorology (INMET).

The effect of rain on the transmission link is very relevant [22], is known as the primary signal attenuation factor at frequencies above 10 GHz. Figure 4.4 shows the graph of accumulated rainfall during August.

Figure 4.4: August accumulated rainfall.



Source: National Institute of Meteorology (INMET).

Considerations on the relationship between weather factors and satellite signal reception will be made after the screening results are presented in Section 4.2.

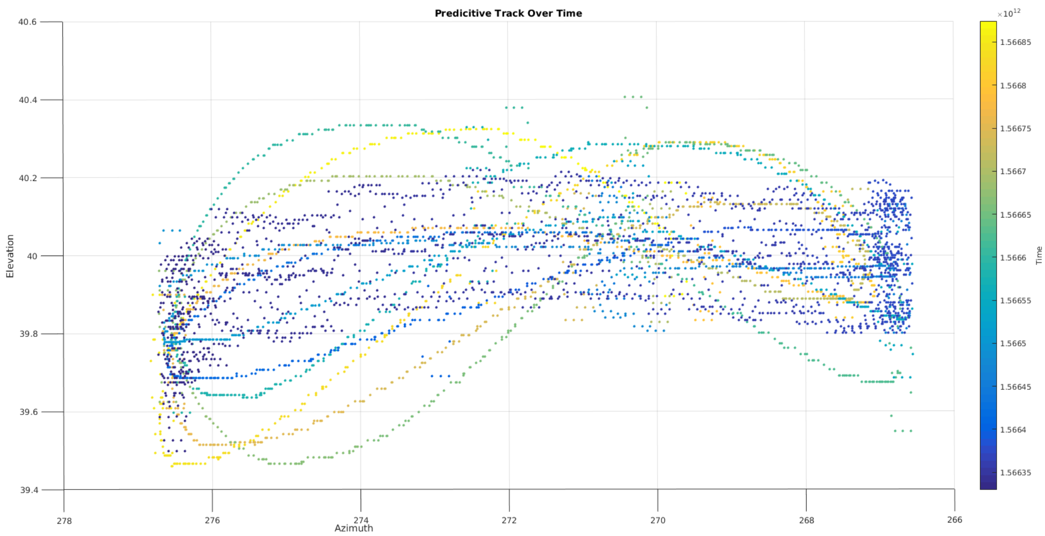
4.2 Algorithms Execution

This section presents the results of the b4 satellite tracking tests using the Predictive Track tracking technique described in 2. The execution of the algorithms is recorded in a system file, containing the positions of the antenna movement axes and the beacon signal strength. This log is read by a script developed for MATLAB / GNU Octave, which interprets the results and generates graphs. In 4.2.1 presents the results of the orbit obtained with BRAZILSAT B4 satellite tracking, a movement described in the azimuth axis and elevation of the antenna over time. In 4.2.2 presents graphs of beacon signal reception, showing signal strength throughout the test.

4.2.1 Tracked Satellite Orbit

Figure 4.5 shows the record of antenna pointing positions throughout the execution of the algorithm, resulting in satellite orbit through the Predictive Track. The duration of the test covers the learning and prediction stage of the algorithm.

Figure 4.5: Brasilsat B4 orbit estimated from the Predictive Track.



Source: Author.

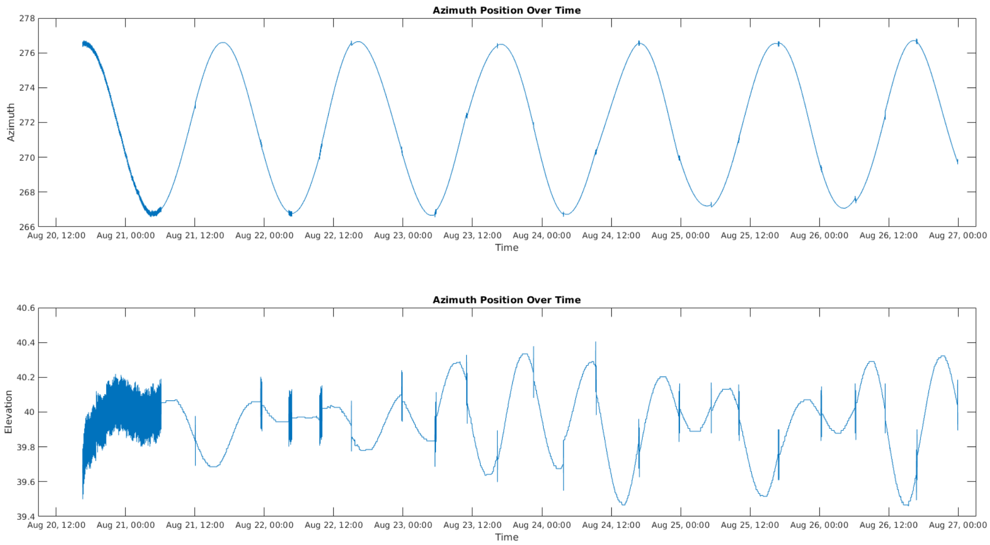
The orbit obtained during the tests coincides with the model presented in Equation 2.1, validated by the orbit seen in N2YO. The learning stage has variance in its estimated pointing position compared to predictions. This variance is explicit in Figure 4.6, which shows the movement with each axis.

The learning stage happens in the first part of the execution. In the first 12 hours of the algorithm execution, the movement presents more variation. This variation decreases in the prediction stage, because the antenna pointing movement is given from the prediction model, and no area search is necessary.

4.2.2 Beacon Signal Strength

During the Predictive Track execution, the beacon signal strength was collected at the antenna pointing position. Although the tested algorithm does not require signal strength for its

Figure 4.6: Azimuth and elevation over time.



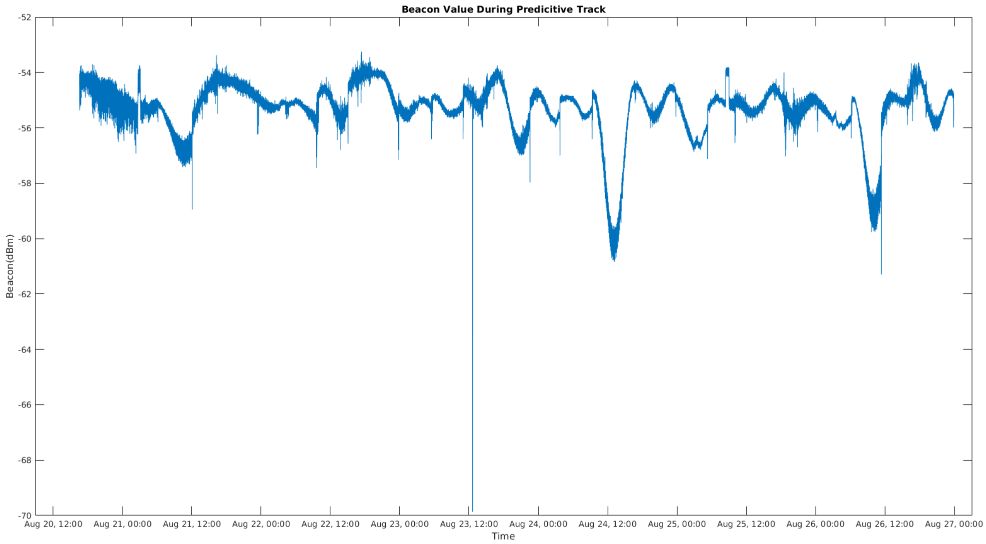
Source: Author.

operation in the prediction stage, beacon signal reception remains the benchmark for assessing link quality. During the installation of the antenna and equipment, the responsible engineer informed that the signal received at -58 dBm would be sufficient to maintain the signal quality. This information is used as a parameter to evaluate the algorithm performance. Figure 4.7 graphs the beacon signal strength by the duration of the test.

The intensity of the recorded beacon signal has an average of -55.29 dBm with a standard deviation of 0.99 dB, remaining above the quality threshold stipulated by the empirical reference of -58 dBm. The signal has a smaller envelope during the learning stage compared to the prediction stage. This reduced envelope costs a higher number of moves made by the antenna but does not affect significant gains for the reception, as the signal remains in range. To compare the performance of the techniques, the beacon signal recording obtained during the test was divided into 12 sample windows, shown in Figure 4.8.

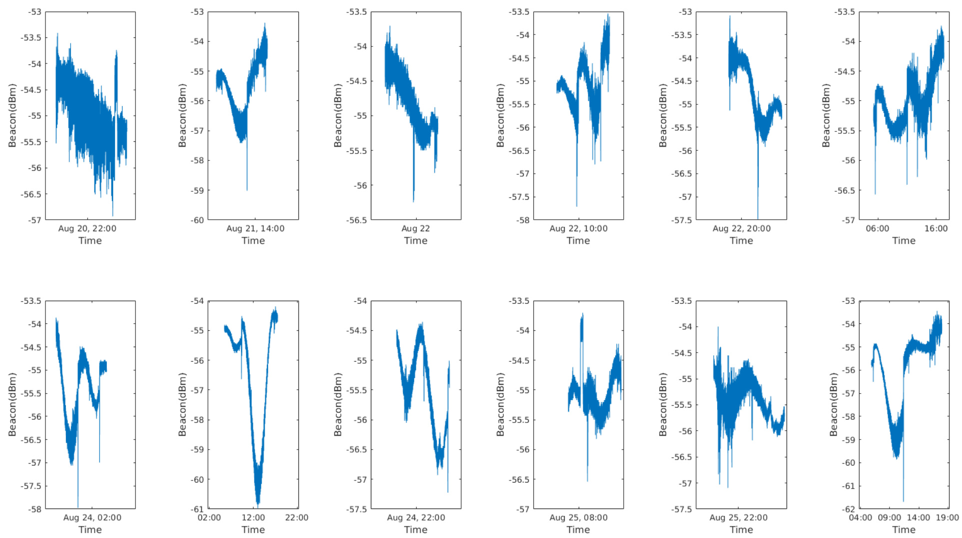
The first window corresponds to the execution of Kalman Track, and the remaining 11 windows are the result of Predictive Track. For each window, the mean, standard deviation, and maximum and minimum values of the samples were measured. The measurements are shown in the Table 4.1.

Figure 4.7: Beacon signal strength over time.



Source: Author.

Figure 4.8: Beacon signal sample windows.



Source: Author.

Table 4.1: Beacon signal sample windows result.

Window	**Mean**	**Std. deviation**	**Max**	**Min**
0	-54.731 dBm	0.463 dB	-53.412 dBm	-56.931 dBm
1	-55.526 dBm	0.871 dB	-53.379 dBm	-59.016 dBm
2	-54.793 dBm	0.391 dB	-53.705 dBm	-56.246 dBm
3	-55.049 dBm	0.484 dB	-53.542 dBm	-57.713 dBm
4	-54.734 dBm	0.605 dB	-53.086 dBm	-57.485 dBm
5	-54.969 dBm	0.426 dB	-53.737 dBm	-56.572 dBm
6	-55.409 dBm	0.675 dB	-53.868 dBm	-57.973 dBm
7	-56.44 dBm	1.892 dB	-54.194 dBm	-60.971 dBm
8	-55.504 dBm	0.674 dB	-54.356 dBm	-57.224 dBm
9	-55.013 dBm	0.348 dB	-53.705 dBm	-56.54 dBm
10	-55.43 dBm	0.346 dB	-53.998 dBm	-57.094 dBm
11	-55.92 dBm	1.575 dB	-53.444 dBm	-61.688 dBm

Source: Author.

From the 11 sample windows of Predictive Track, the results were averaged for comparison with the window representing Kalman Track performance. The evaluated performance is described in the Table 4.2.

Table 4.2: Algorithms performance results.

Window	**Mean**	**Std. deviation**	**Max**	**Min**
Kalman Track	-54.731 dBm	0.463 dB	-53.412 dBm	-56.931 dBm
Predictive Track	-55.409 dBm	0.605 dB	-53.705 dBm	-57.485 dBm

Source: Author.

The execution of the search algorithm obtained better results than the predictions made by the Predictive Track estimation model, but the results presented are not discrepant. The best results presented by Kalman Track have the cost of the largest number of movements made in the antenna, in contrast to the savings of movements made by Predictive Track has a greater inaccuracy in orbit. It is necessary to assume the compromise established in the tracking, as both techniques present advantages and disadvantages from the perspective presented.

Climatic factors did not have much influence on the tracking algorithm during the tests, even with the heavy rains and high temperatures recorded. A power outage during the tests. This power outage did not affect the functioning of the algorithm, but at noon on August 23, a rapid negative spike in the signal can be observed.

Chapter 5

Conclusion

This work implements satellite tracking techniques in a simulation environment and in a real control system, in order to generate the performance evaluation and compare the techniques in several proposed situations. Also presented is the development of the virtual environment based on the satellite tracking scenario to interact with an RL agent.

The main contribution of the work is the comparison between two developed algorithms that are based on different techniques, during a test period seven times longer than the period of the tracked satellite orbit and uninterruptedly. These comparisons (Kalman Track Algorithm Tracking Performance vs. Predictive Track) can be modified to generate new results with the performance of different techniques. The definition of metrics to measure satellite tracking performance based on stochastic behaviors of the beacon signal obtained during the tests is fundamental for the development of this work, allowing to create a standard evaluation methodology for all techniques implemented in the system used. To the best of the author's knowledge, the reviewed papers, and the satellite tracking literature do not present any consensus on the performance evaluation methodologies, and few studies present real results. These metrics will serve as a baseline for tracking performance comparisons of future work in the same area.

The data acquisition from a real satellite link enriches the research, leaving the scope of computer simulations and implementing the techniques in an antenna control system. The records of the tests performed with the screening techniques will be available for validation and reproduction of the results obtained in this work.

The results obtained in the tests confirm the mathematical equations used to model the behavior of the beacon signal along with the antenna pointing movement space, and the orbital motion of a satellite in an inclined orbit. About the beacon signal, the Kalman Track operating period shows that the algorithm can estimate a position close to maximum reception over time using the proposed model, where it is also possible to observe the presence of the described noise. About the orbit obtained during the tracking, the results are in agreement with the orbit shown in the N2YO and the mathematical model.

The uncertainty problem involved in using only prediction model-based tracking techniques as the sole reference for antenna pointing correction is overcome, with the addition of an area search algorithm to train the model with estimates of the best position. Predictive Track proves to be a solution in inclined-orbit satellite tracking techniques, as model prediction performance result similar to Kalman Track search algorithm performance. The results obtained with the two tracking techniques tested are consistent with other databases presented in the work, reinforcing the data validity and creating a position reference based on various estimates made by different techniques.

Other contributions of this work were the implementation of a monopulse system-based screening technique, and development of the virtual environment based on the satellite-tracking scenario, to interact with an RL agent. The implementation of a monopulse processor using SDR and GNU Radio is useful as it serves as an alternative to non-availability of single-pulse antennas and equipment, behaving as expected in the literature. The RL Agent using DQNs has not obtained consistent results as of this writing. Despite this, the environment in which it interacts is up and running and available for use and collaboration.

5.1 Publications

Throughout the master's degree course in electrical engineering at Electrical Engineering Graduate Program (PPGEE), the author of this work has published two related papers:

SBRT 2018 The paper named *Development and Evaluation of Least Square Satellite Tracking in Real Antenna Control System* was published at the XXXVI Brazilian Symposium on Telecommunications and Signal Processing in September 2018. This work describes the implementation of a satellite tracking technique in a real antenna control system. The technique described is Kalman Track, and uses least squares estimation to search for the best antenna pointing position.

ENCOM 2019 The paper named *Satellite Tracking Environment for Reinforcement Learning using OpenAI Gym* accepted and will be published the IX National Conference on

Communications, Networking and Information Security in October 2018. The paper implemented the satellite tracking scenario in a simulation environment for Reinforcement Learning agents using OpenAI Gym. This environment is based on a case study of antenna control and actual satellite tracking using data collected during testing.

5.2 Future Works

Tests will continue with satellite tracking techniques, acquiring more data to increase the database sample and make comparisons between different periods of the year. The new database will also be used to further study the correlation between climate factors and the quality of satellite signal reception.

The RL-based satellite tracking algorithm will be optimized for its operation within the developed environment. Other approaches will also be made using ML techniques, approaching the functioning of the Kalman Track algorithm.

Bibliography

[1] United Nations Office for Outer Space Affairs. (2019) Outer Space Objects Index. [Online]. Available: http://www.unoosa.org/oosa/osoindex/index.jspx?lf_id=

[2] SpaceX. (2019) Arabsat-6a mission. [Online]. Available: https://www.spacex.com/news/2019/04/12/arabsat-6a-mission

[3] ——. (2015) Reusability: The key to making human life multi-planetary. [Online]. Available: https://www.spacex.com/news/2013/03/31/reusability-key-making-human-life-multi-planetary

[4] T. Kelso, "Basics of the geostationary orbit," *Satellite Times*, vol. 4, no. 7, pp. 76–7, 1998.

[5] n2yo. (2019) Live real time sattelite traking and predicitions. [Online]. Available: https://www.n2yo.com

[6] J. I. Laine, "2d model-based step-track procedure," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 4, pp. 1386–1391, 2000.

[7] T. L. Sarmento, "Técnicas de rastreio de satélites para antenas de 3 eixos," Undergraduate's thesis, Faculdade de Engenharia da Computação e Telecomunicações, 2017.

[8] F. R. Hoots and R. L. Roehrich, "Models for propagation of norad element sets," AEROSPACE DEFENSE COMMAND PETERSON AFB CO OFFICE OF ASTRODYNAMICS, Tech. Rep., 1980.

[9] K. Dismukes, "Definition of two-line element set coordinate system," *National Aeronautics and Space Administration*, 2011.

[10] T. Sarmento, A. Nakamura, L. Correa, M. Takeda, A. Castro, and A. Klautau, "Development and evaluation of least square satellite tracking in real antenna control system," *arXiv preprint arXiv:1908.02304*, 2019.

[11] G. Hawkins, D. Edwards, and J. McGeehan, "Tracking systems for satellite communications," in *IEE Proceedings F (Communications, Radar and Signal Processing)*, vol. 135, no. 5. IET, 1988, pp. 393–407.

[12] S. M. Sherman and D. K. Barton, *Monopulse principles and techniques*. Artech House, 2011.

[13] E. research. (2019) Ettus research. [Online]. Available: https://www.ettus.com/

[14] G. Radio. (2019) About GNU Radio. [Online]. Available: https://www.gnuradio.org/about/

[15] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 2, no. 4.

[16] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[18] OpenAI. (2019) Gym OpenAI. [Online]. Available: https://gym.openai.com/

[19] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, "Deep exploration via bootstrapped dqn," in *Advances in neural information processing systems*, 2016, pp. 4026–4034.

[20] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[21] Keras. (2019) Keras: The Python Deep Learning library. [Online]. Available: https://keras.io/

[22] A. Spaulding, "Atmospheric noise and its effects on telecommunication system performance," *Handbook of Atmospheric Electrodynamics*, vol. 1, 2017.