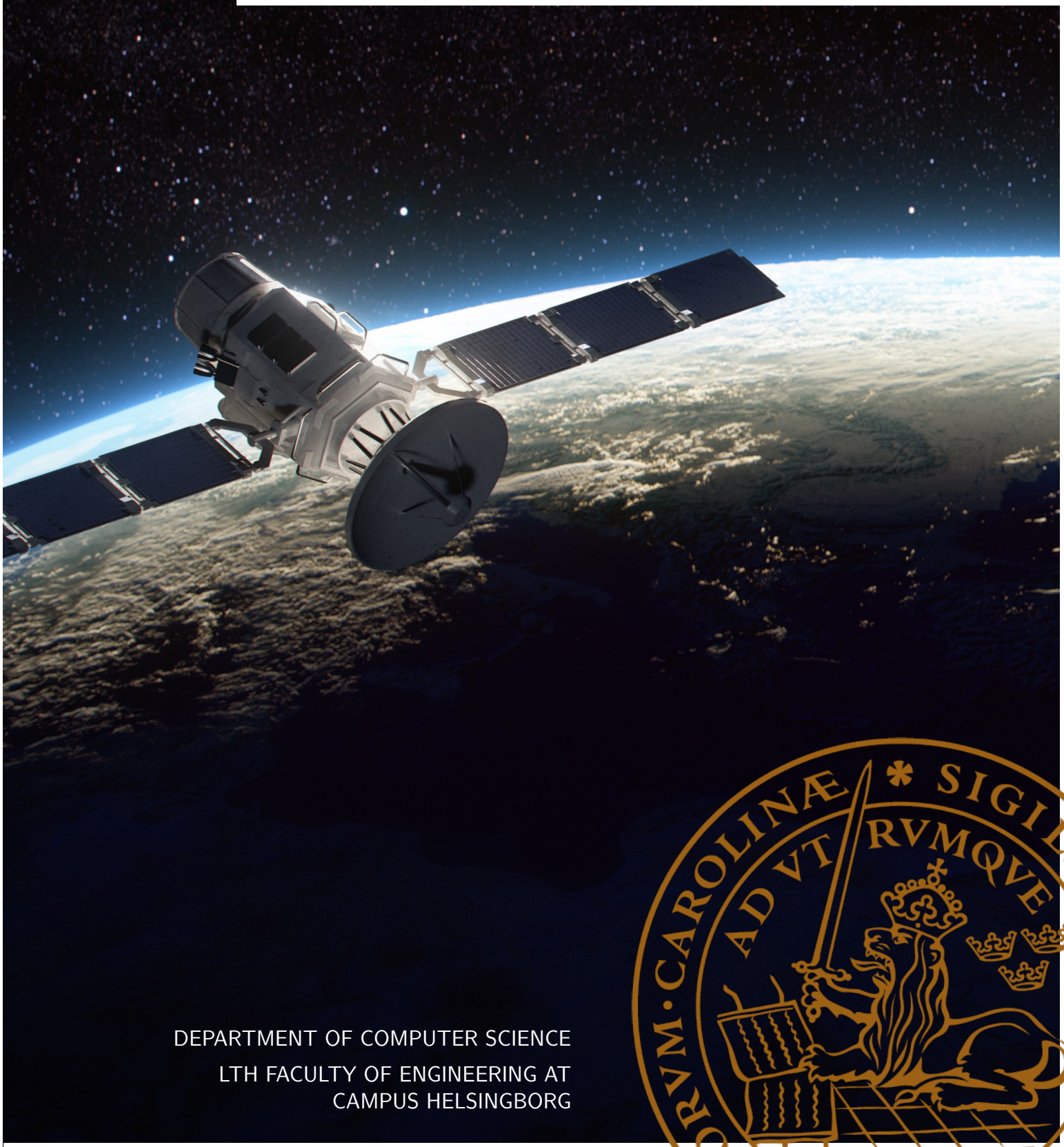
BACHELOR'S THESIS 2024

GNSS Training Simulator

Julius Andersson, Simon Antonsson



DEPARTMENT OF COMPUTER SCIENCE
LTH FACULTY OF ENGINEERING AT
CAMPUS HELSINGBORG

© Copyright Julius Andersson, Simon Antonsson

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

LTH vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

Printed in Sweden
Media-Tryck
Biblioteksdirektionen
Lunds universitet
Lund 2024

KANDIDATARBETE
Datavetenskap

GNSS Training Simulator

GNSS Träningssimulator

Julius Andersson, Simon Antonsson

GNSS Training Simulator

Julius Andersson
Julius.Ola.Andersson@gmail.com

Simon Antonsson
zomon1@hotmail.com

May 27, 2024

Bachelor's thesis work carried out at
the Department of Computer Science, Lund University.

Supervisor: Anders J Johansson, anders_j.johansson@eit.lth.se

Examiner: Christin Lindholm, christin.lindholm@cs.lth.se

Abstract

This thesis was conducted in collaboration with Saab Kockums AB. Saab Kockums is the marine division within Saab and is a large manufacturer of military naval vessels. With the construction of naval vessels comes the need for training opportunities for the crew operating them, and one essential part is simulations of GNSS (Global Navigation Satellite Systems). Currently, the GNSS simulator in use has two problems. Firstly, it is static, with some input values being persistent throughout an entire simulation. The secondary problem being the inability to simulate interference.

This thesis investigates different aspects of GNSS simulations, as well as the implementation of a simulator prototype. The software built is a simulator that utilises the ephemeris data stored in GNSS broadcast messages. By downloading the files and parsing them, orbital parameters can be extracted. It is then possible to determine the positions of satellites at different times. Based on the satellite geometry at a given time in the simulation, it is possible to calculate the accuracy of the receiver position. Also conducted was an investigation into how GNSS interference could be simulated with various approaches being discussed and ultimately implemented.

The result of this thesis is a simulator built in the C# language within the framework .NET. For ease of use and visualisation, the simulator outputs messages with the NMEA 1803 protocol, through a virtual serial port to a third-party program that can visualise the positions of the satellites during the simulation. A GUI (Graphical User Interface) was also constructed to allow the simulation operator to easily change scenes/scenarios during a simulation. In the GUI there are options to configure the positions of the receiver and jammer, and the desired range of the jammer can be modified.

Keywords: GNSS, Simulator, Interference, Satellite positions, Dilution of precision

Sammanfattning

Detta examensarbete har utförts i samarbete med Saab Kockums AB. Saab Kockums är marindivisionen inom Saab och är en stor tillverkare av militära sjöfartyg. Med konstruktion av sjöfartyg uppstår det också ett behov av träningsmöjligheter för besättningen som använder dem. Där är simuleringar av GNSS (Global Navigation Satellite Systems) en väsentlig del. För närvarande har GNSS-simulatorn som används två problem. För det första är den statisk, med vissa värden beständiga under en hel simulering. För det andra så saknas möjligheten att simulera interferens.

Detta examensarbete undersöker olika aspekter av GNSS-simuleringar, tillsammans med en implementation av en simuleringsprototyp. Den framtagna programvaran är en simulator som använder efemerid-data som lagras i GNSS meddelanden. Genom att ladda ner filerna och bearbeta dem kan man extrahera parametrar om satelliternas omloppsbanor. Det är då möjligt att bestämma satelliternas positioner vid olika tidpunkter. Baserat på satellitgeometrin vid en given tidpunkt i simuleringen är det möjligt att beräkna noggrannheten i den resulterande positionen. Även en undersökning genomfördes om hur GNSS-störningar kunde simuleras, med olika tillvägagångssätt som diskuterades och slutligen implementerades.

Resultatet av examensarbetet är en simulator som byggdes i språket C# inom ramverket .NET. För enkel användning och visualisering genererar simulatorn meddelanden med NMEA 1803-protokollet. Meddelanden kan i sin tur skickas genom en virtuell serieport till ett tredjepartsprogram som kan visualisera satelliternas positioner under simuleringens körning. Ett GUI (Grafiskt Användargränssnitt) konstruerades också för att möjliggöra för simuleringsoperatören att enkelt ändra scener/scenarier under en simulering. I användargränssnittet finns också möjligheten att konfigurera mottagarens och störsändarens positioner, och den önskade räckvidden för störsändaren.

Nyckelord: GNSS, Simulator, Interferens, Satellit positioner, Felmarginal av precision

Acknowledgements

To begin, we extend our gratitude to Andreas Carlsson and Saab for granting us the opportunity to work on this thesis. This experience has provided us with invaluable learning opportunities and improved our chances of navigating challenges in future careers.

We also express our sincere appreciation to our supervisor at Saab, Hannes Andersson, for his invaluable assistance with code structuring, project management, and insightful discussions on various topics.

Furthermore, we would like to extend our thanks to Mikael Eriksson for his guidance and discussions on GNSS functionality and for his valuable input and feedback.

Additionally, we would like to acknowledge our supervisor at LTH (Lund University of Technology), Anders J Johansson, for his support with the report and for providing us with great information and resources on GNSS and interference.

We would like to express our gratitude to Christin Lindholm for undertaking the role of the examiner for our thesis.

Finally, we want to thank everyone at the office in Helsingborg for welcoming us, being available for discussions, and for making this thesis work an immensely enjoyable experience.

Contents

1 Introduction **1**
1.1 Goal . . . 1
1.2 Problem . . . 2
1.3 Motivation . . . 2
1.4 Limitations . . . 3
1.5 Work Distribution . . . 3

2 Technical Background **5**
2.1 GNSS - Global Navigation Satellite Systems . . . 5
2.2 WGS84 - World Geodetic System 1984 . . . 5
2.3 ECEF - Earth-Centered, Earth-Fixed . . . 6
2.4 DOP - Dilution of Precision . . . 7
2.5 Ephemeris Files . . . 8
2.6 Rinex v4 . . . 8
2.7 NMEA 0183 . . . 9
2.8 GPS Jamming . . . 9

3 Method **11**
3.1 Office Work . . . 11
3.2 Agile Work . . . 12
3.2.1 Milestones . . . 12
3.2.2 Trello . . . 12
3.2.3 Weekly Logs . . . 12
3.3 Communication . . . 12
3.4 Information Gathering . . . 13
3.5 Pair Programming . . . 13
3.6 Version Control . . . 13
3.7 MVP and Second Iteration . . . 13
3.8 Source & Information Evaluation . . . 14

4 Implementation **15**
4.1 System Architecture . . . 15
4.2 Models . . . 15
4.3 Parsing of Rinex File . . . 16
4.4 Calculating the Satellite's Position in ECEF . . . 17
4.5 Converting Coordinates to ECEF . . . 20
4.6 Updating the Parameters . . . 20

4.7 Predicting the Satellite's Orbit . . . 20
4.8 Visible Satellites . . . 21
4.9 Azimuth & Elevation . . . 22
4.10 Calculating Dilution of Precision . . . 23
4.11 SNR and Field Survey . . . 24
4.12 Interference . . . 25
4.12.1 Different Jamming Techniques . . . 25
4.12.2 Jamming Discussion . . . 25
4.12.3 Fictional Model . . . 26
4.12.4 Signal to Noise Ratio Model . . . 28
4.12.5 Jamming to Signal Ratio Model . . . 30
4.13 Altitude . . . 31
4.14 Generating NMEA-Strings . . . 31
4.15 Third Party Program Utilisation . . . 32
4.15.1 VisualGPSView . . . 33
4.15.2 Virtual Serial Port . . . 33
4.16 GUI . . . 34
4.17 Unit Tests & Documentation . . . 35
4.18 Library Options . . . 36
4.18.1 Conversions . . . 36
4.18.2 Rinex Parser . . . 36
4.19 Accuracy of Position Calculations . . . 36

5 Result **37**
5.1 GUI Design . . . 38
5.1.1 Receiver Configuration . . . 38
5.1.2 Interference Configuration . . . 39
5.1.3 Simulation . . . 41
5.2 Modular Programming . . . 42
5.3 Simulation . . . 42

6 Conclusion **43**
6.1 Process . . . 43
6.2 A Simulation Not an Emulation . . . 44
6.3 Changes from Original Plan . . . 44
6.3.1 Deviation from Original Constellation Plan . . . 44
6.3.2 Addition of GUI . . . 44
6.4 Possible Future Development . . . 45
6.4.1 Wider Support for Constellations . . . 45
6.4.2 Multiple Jammers . . . 45
6.4.3 Jammer Simulation & Spoofing . . . 45
6.4.4 SNR . . . 46
6.4.5 Movement & Receivers . . . 46
6.4.6 Wider Test-cases . . . 46
6.4.7 GUI . . . 46
6.4.8 Integration with Saab Simulator Engine . . . 46
6.5 Ethical Aspects . . . 47

References 49

Appendix A Trello 55

Appendix B Models 57

Appendix C NMEA 59

Appendix D Goal product 61

Terminology

ECEF Earth-Centred, Earth-Fixed

Galileo European Satellite Constellation

GDOP Geometric Dilution of Precision

Glonass Soviet Satellite Constellation

GNSS Global Navigation Satellite Systems

GPS Global Positioning System

GUI Graphical User Interface

HDOP Horizontal Dilution of Precision

MVP Minimum Viable Product

NASA National Aeronautics and Space Administration

NMEA National Marine Electronics Association

PDOP Position Dilution of Precision

RINEX Receiver Independent Exchange Format

TDOP Time Dilution of Precision

VDOP Vertical Dilution of Precision

WGS84 World Geodetic System 1984

Chapter 1

Introduction

This thesis has been conducted in collaboration with Saab Kockums, a large Swedish company in the defense industry. Saab Kockums specializes in the marine environment, particularly in the development and manufacturing of submarines and surface vessels, with the office in Helsingborg having a focus on training and simulations.

To help the military train their personnel they need the opportunity for training without the need for deploying boats or submarines, as it would be an ineffective and cost-heavy way of training. This is where simulators come into discussion. Simulators make it easier to train personnel and it greatly reduces the cost of training. Using simulators is also a safer way of training. This thesis consists of the implementation of a prototype GNSS training simulator with discussions on different ways interference can be simulated.

1.1 Goal

The goal is to implement a prototype in C# that will be able to take input parameters such as receiver and jammer location, type of GNSS (GPS, Galileo, or Glonass), and a On-Off parameter, together with the ability to input ephemeris files. From the program, you should be able to obtain Position, Current time, Quality indicator, Horizontal Dilution of Precision, Vertical Dilution of Precision, Position Dilution of Precision, and the satellites used for the calculations. Additionally, it should also be possible to introduce some form of interference during the simulation process.

1.2 Problem

Saab currently has a functional GNSS simulator, but with certain deficiencies that remove its realism. The simulator takes static inputs such as PDOP, HDOP, and VDOP, and outputs the location accordingly without any calculation being made. It also lacks the possibility of simulating interference, rendering it less effective for realistic training scenarios. The result of this thesis will be answering the following questions:

- How do you create a GNSS simulator that outputs an approximate position?
- How do you calculate and predict a satellite position from an ephemeris file?
- How do you calculate dilution of precision from a group of satellites?
- What are the possibilities for simulating interference and in what ways can it be implemented?

1.3 Motivation

During the autumn of 2023, Julius had the opportunity to do the project course MAMF-40 in collaboration with Saab Kockums in Helsingborg. This experience turned out to be incredibly enjoyable, educational, and rewarding. Later when the opportunity to apply for a thesis project in collaboration with Saab arose, it felt like a natural choice.

The thesis description aligned well with our interests, allowing us to apply our knowledge to a real project. The focus on practical applications rather than theoretical research was particularly appealing, making this project a good fit.

For Saab, this is an opportunity to further collaborate with students and it is also an important part of keeping a tight collaboration with the university. Allowing for thesis opportunities is also a great way for Saab to present and expose the company to students and in the long run a way of recruiting employees. With this thesis, they get valuable input for improvements of the training simulator as well as an investigation into the ability to simulate GNSS interference. It is also beneficial that both theory and code implementation are compiled in the same place, allowing for easy future integration with the actual simulator. Furthermore, thesis opportunities are a great way of limiting the time spent on trying new things.

1.4 Limitations

For this thesis, some aspects fall outside its scope. Firstly, integrating the simulator with Saab's current simulator engine is not within the scope due to the complexity that comes with it, as well as other complications. Additionally, implementing support for other GNSS constellations such as Beidou (China) and QZSS (Japan) is beyond the focus of this thesis. With the utilisation of Win-Forms for the GUI, the complete program is only available on Windows due to the lack of support for Win-Forms on Mac OS and Linux. The use of Visual-GPSView and virtual COM ports also limits the visualisation aid to Windows. However, the main functionality would still work on both Mac and Linux due to .NET 7 multi-platform support. Due to instructions from Saab, the development environment of this program was limited to Visual Studio and C#.NET.

1.5 Work Distribution

In the initial phase of the thesis and for the MVP (Minimal Viable Product), programming tasks were done through pair programming, as well as the gathering of relevant information. Upon achieving a basic operational program the focus was split between refining the computational aspects and the development of a GUI (Graphical User Interface). Throughout this process, the pair programming approach persisted and continuous talks and discussions were held. The work distribution can be seen in Table 1.1 below.

Table 1.1: Work distribution.

Stage	Julius	Simon
Information gathering	50	50
MVP	50	50
Calculation enhancement	90	10
GUI Implementation	10	90
Writing	50	50
Poster	50	50
Presentation	50	50

Chapter 2

Technical Background

Understanding the implementation of a GNSS training simulator requires some knowledge of fundamental theories. Therefore, this chapter provides a basic overview of domain-specific concepts such as Dilution of Precision, NMEA, and GPS jamming. These concepts are some of the foundations for developing a GNSS training simulator.

2.1 GNSS - Global Navigation Satellite Systems

GNSS is a collective term for navigation systems used worldwide. Different countries utilise various navigation systems, such as GPS which is being operated by the USA, Galileo which is under operation by the European Union and Glonass being operated by Russia. China, Japan, and India also have constellations of their own. The first navigation system constructed was GPS, first made for military purposes in the 1970s by the USA and later in the early eighties Glonass was launched by the Soviet Union. Galileo was put in service much later with the start of operations in 2016. Although satellite navigation began in the military domain, over time, GNSS technology has been made accessible for non-military applications, such as agriculture and all forms of traffic, resulting in a revolution in navigation and positioning capabilities globally. [1]

2.2 WGS84 - World Geodetic System 1984

WGS84 is a globally recognized standard reference system used for geographic positioning and mapping purposes. This system establishes a reference ellipsoid, which serves as a mathematical representation of the Earth's shape. Using latitude, longitude, and altitude, WGS84 provides a standardised method for pinpointing locations on the Earth's surface. It enables precise positioning and navigation. WGS84 is used in various applications such as cartography, surveying, and GPS navigation.

It serves as the basis for geospatial data exchange and communication, ensuring consistency and interoperability between different systems and datasets worldwide. Various other very similar reference systems exist such as PZ-90, which is the reference system used by Glonass, and GRS-80 used by Galileo. GRS-80 and WGS84 are often considered synonymous, differing only in their flattening parameter, which results in a 0.0001-meter deviation in half the semi-minor axis. [2]

2.3 ECEF - Earth-Centered, Earth-Fixed

ECEF is a cartesian coordinate system with origo in the earth's center of mass and can be used to determine positions both on Earth and in the surrounding atmosphere. The coordinate system is fixed on Earth and rotates together with Earth. The z-axis points its positive part towards and through the North Pole. The x and y-axes are the equatorial plane with x running through the prime meridian (0° longitude) and the y-axis being 90° east, this can be seen in Figure 2.1.

ECEF, with its three-dimensional coordinates, provides a good framework for pinpointing locations in Earth's atmosphere, with the use of cartesian coordinates (X, Y, Z). This multi-dimensional aspect increases accuracy, especially in applications such as satellite navigation. Additionally, ECEF's three-dimensional representation makes coordinate transformations easier and helps interoperability between different positioning technologies.

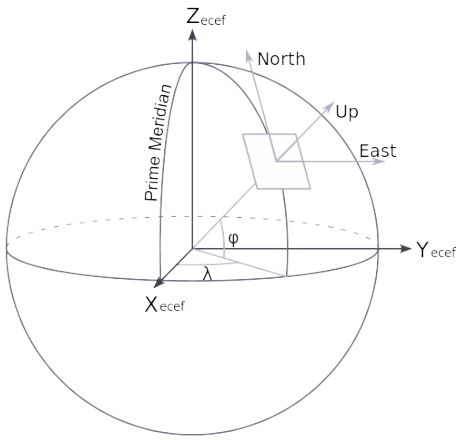


Figure 2.1: ECEF ENU Longitude Latitude relationships [3]

2.4 DOP - Dilution of Precision

DOP is a measurement used in satellite navigation to quantify the geometric quality of the satellite constellation from the receiver's perspective. It can primarily be divided into HDOP (Horizontal Dilution of Precision) and VDOP (Vertical Dilution of Precision), which correspond to the margin of error for longitude, latitude, and altitude, this is gathered by calculating the spread of satellites in the horizontal or vertical plane. However, there are other concepts like PDOP (Position Dilution of Precision) which is a mix of HDOP and VDOP, TDOP (Time Dilution of Precision) that compares the difference in time in the internal clock of the satellite and the receiver, and GDOP (Geometric Dilution of Precision) that is the same as PDOP but with more information like time and velocity calculated into it.

The margin of error is significantly influenced by the receiver's position relative to the positions of the satellites used to determine its location, especially by the number of satellites used and their elevation angles relative to the receiver. Illustrated in Figure 2.3, the margin of error can be represented as an "area" in two scenarios with different relationships between two satellites. If you compare this with the illustrations in Figure 2.2 you get an understanding of why the DOP value can increase with satellites too close to each other and why you would want the satellites further apart. [4]

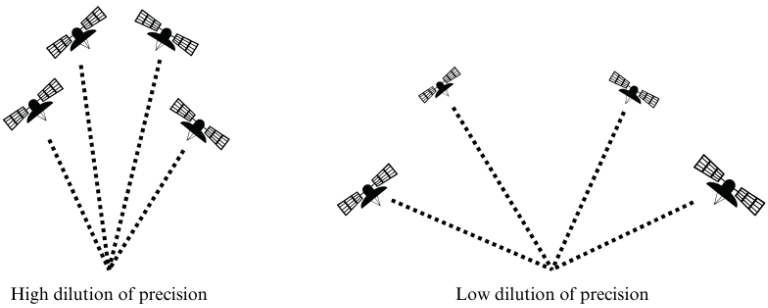


Figure 2.2: Two different scenarios of Dilution of precision.

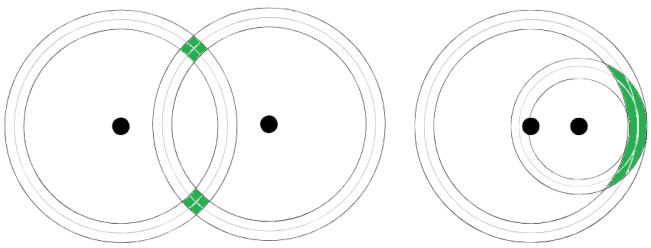


Figure 2.3: Overlapping areas represent DOP.

2.5 Ephemeris Files

Ephemeris files contain information about the position and movement of celestial bodies, such as satellites. They are produced by satellite control centers or other organisations responsible for maintaining and operating the satellite constellation. On NASA's website [5] these files are created several times a day in various versions of the data standard RINEX.

There are different types of Ephemeris files, including broadcast, ground station, and almanac files. The broadcast files contain information broadcasted by the satellites themselves. It includes their current clock time and orbital parameters. The ground station files are produced by having the ground receivers collect data from the satellites they receive signals from, which are on the same side of the globe. The almanac files are calculated predictions derived from the broadcast files and contain information about where the satellites are expected to be in the future. These predictions are typically valid for approximately 90 days. However, the precision of the satellite positions diminishes the further into the future the predictions extend. [6]

2.6 Rinex v4

RINEX v4 represents the most recent iteration of the RINEX format [7], developed to facilitate the easy exchange of satellite navigation data. The RINEX v4 format encompasses three distinct file types: observation data, navigation messages, and meteorological data files. Each file is structured with a header containing metadata detailing the file's contents, followed by a data section containing satellite information. Within observation files, the data section is organised into blocks, with each block representing a message from an individual satellite message. These messages include essential parameters such as ID, date, time, clock drift, eccentricity, position correction parameters, satellite angles, time of message, and time of ephemeris together with several others. [8]

2.7 NMEA 0183

NMEA 0183 is a specification for communication between various marine electronics, such as GPS receivers, echo sounders, sonars, anemometers, and gyrocompasses. The NMEA 0183 standard has 26 different message types [9], each designed to convey specific data related to marine navigation and communication. These messages cover various aspects such as position, speed, course, and sensor data. However, users do not need to use all message types to use NMEA, instead, they can choose only those messages that are relevant to their navigation or communication. Some of the messages in NMEA used for GNSS are GPGGA, GPGSV, and GPGSA which all correspond to messages related to GPS, as can be seen by the prefix "GP" in the first two letters. A block of NMEA data could look like the following message shown in Figure 2.4 where the first line includes data parameters like time, longitude, latitude, number of satellites in view, and various other information. The second line contains up to eleven satellites used for tracking together with the different DOP values, in this case, a PDOP of 1, HDOP of 0.6, and VDOP of 0.7, as can be seen on the last three values before the checksum in the GPGSA message. The checksum is the hexadecimal value after the asterisk. [10].

```
$GPGGA,100000.00,5602.8038,N,01241.6622,E,1,11,0.6,10,M,-15,M,0,,*6B
$GPGSA,A,3,07,08,13,14,15,17,18,22,23,24,30,,1,0.6,0.7*2B
$GPGSV,3,1,11,07,7,81,25,08,11,24,45,13,80,177,48,14,59,93,32*7F
$GPGSV,3,2,11,15,56,284,24,17,12,124,43,18,4,290,3,22,61,138,22*7A
$GPGSV,3,3,11,23,22,320,46,24,22,263,6,30,37,83,26*44
```

Figure 2.4: Example NMEA output.

2.8 GPS Jamming

Jamming within the domain of GPS communication is the objective of obstructing a GPS receiver's ability to either establish communication with satellites or to interfere and disrupt an already established connection. It works by utilising one of the various jamming techniques. These techniques involve transmitting a signal with the same frequency as the target communication frequency, which in the GPS scenario is the civilian L1 band which operates at 1.575 GHz, but with a higher signal power. It should however be noted that today the L5 band also exists, operating at 1.176 GHz, which some civilian receivers can utilise. This work however only looks at the L1 frequency band.

If the jamming signal is high enough the GPS receiver will no longer be able to sufficiently work and compute its location. Due to the GPS signal's very low strength at the earth's surface, it's an easy signal to jam and is therefore often exposed to various jamming attacks. An often occurring scenario for this is GPS jammers installed in vehicles to hinder employers from tracking an employee's location. These types of jammers use the 12V cigarette outlet in cars to transmit a signal strong enough to jam GPS signals in a radius of a few meters to sometimes tens of meters. [11]

Other communications technologies that can be targeted for jamming attacks are for example Bluetooth and WiFi communications, which likewise to GPS is relatively easy to jam. [12]

Chapter 3

Method

Given Saab's high security standards, the initial phase of the thesis progressed slowly while waiting for security and background checks to be completed and approved. During this time, efforts were made to gather as much information as possible. Once the checks were approved, work at the office began with the goal of maximizing on-site productivity. This chapter details the methodology employed during the thesis work. In Figure 3.1 the different phases is shown and what parts was implemented at each stage of the development. Also indicated are the phases that were done in a more iterative manner, where continuous tests and evaluations were made throughout the implementation.

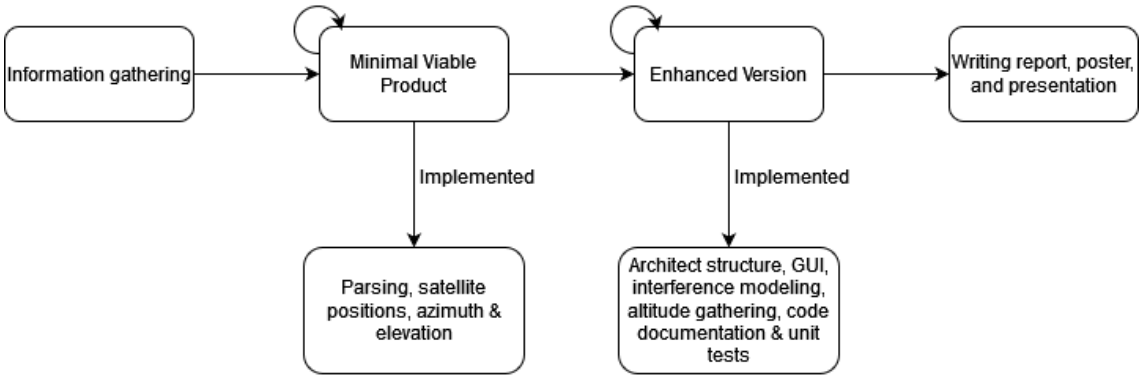


Figure 3.1: Summarize of what was done in different phases.

3.1 Office Work

With most of the work during the thesis taking place at Saab's office in Helsingborg, not only did it help with the work structure but it also helped a great deal having the possibility for guidance short at hand. A lot of ideas and problem-solving came from casual talks and brainstorming, which would not have been possible if not for the office environment.

3.2 Agile Work

As only two people were working on this thesis, the work was done in an agile way. Meetings were held together with the supervisor multiple times each week where discussions on progress, challenges, and potential adjustments were covered in order to finish the project goals. These frequent meetings with the supervisor at Saab provided valuable opportunities for real-time feedback and ensured alignment throughout the thesis process.

Furthermore, the flexibility of working with an agile method allowed for adaptive planning and response to issues and changing priorities. For instance, if certain aspects of the work encountered unexpected obstacles or required more attention than initially anticipated, adjustments were made without compromising the overall project timeline.

3.2.1 Milestones

Throughout the development process, the work was organized into milestones to provide clarity on completed tasks and identify the next steps. This approach made the division of work easier and prevented getting stuck on a particular problem. Therefore, if a challenge arose that couldn't be immediately resolved, the ability to move on to the next part ensured continuous progress without unnecessary delays.

3.2.2 Trello

A Trello KANBAN board, shown in Appendix A.1 , was created for easier organisation of tasks. The board was split in two, where one was used for tasks related to the development of the program and another one was created for tasks related to the writing of this thesis. This way, it was easy to see what needed to be done, and what tasks lay ahead. By ordering them with the highest priority at the top, it also created a structure for which tasks to focus on.

3.2.3 Weekly Logs

Each week, informal logs were written to document the work completed during that time. The primary purpose of maintaining a logbook was to serve as a reference during the development of the thesis rapport.

3.3 Communication

When working outside the office environment, for example on weekends or evenings, Discord served as the primary communication platform. Otherwise, most communication occurred verbally in the office. Communication with the supervisor at LTH primarily occurred through email, while meetings were occasionally held using a combination of Zoom and onsite sessions in Helsingborg.

3.4 Information Gathering

Due to the nature of programming, it was easy to gather information where a simple Google search could result in having multiple solutions for a problem. And with programming, verifying the correctness is a matter of implementing it and making sure it compiles. When it comes to GNSS it was in general very well documented, information could be found in articles, books, manuals, and more. This made it easy to find, however, because there was so much information it made the process of getting the useful data and putting all the pieces together complicated.

3.5 Pair Programming

During more complex implementations like calculations of azimuth, as well as in the beginning, pair programming was used to solve problems faster and allow for easy discussion and communication when encountering problems. It was also a great tool in the early stages when first learning a new language and environment.

3.6 Version Control

Git was used for easy version control and collaboration. It also allowed for easy "stepbacks" if some experimental implementation did not turn out the way it was expected. The combination with GitHub made it possible for the supervisor at Saab to check the code and give feedback on things needing changes or modifications.

3.7 MVP and Second Iteration

The development of the program was split into two different phases with different focuses. In the initial phase of development, an MVP (Minimal Viable Product) was created where the focus was to implement as many functions and features as possible. Here, the need for a GUI, unit tests, and extensive documentation was ignored, only utilizing a terminal as the user interface.

After the most important functionality features were in place, a more well-structured working prototype of the program was developed, where a GUI was designed and implemented. With guidance and help from Saab, a 'system architecture' was created focusing on achieving low coupling and high cohesion to facilitate future maintenance and enhancements. In the later version, four unit tests were added for some of the essential parts of the calculations. Another focus was placed on documentation, with functions in the code being well-documented and explained.

3.8 Source & Information Evaluation

During the work of this thesis, multiple types of sources were used to gather information. Because a lot of calculations are related to space, geodesy, and satellites, it was natural to turn to space agencies such as NASA and ESA (European Space Agency) for information. These are also sources [5, 8, 26, 30] that can be deemed reliable.

Another well-used source for this thesis is the Swedish land survey authority Lantmäteriet [2,16,23]. As a part of the Swedish Department of Rural Affairs and Infrastructure, the source can be seen as reliable. The information from published scientific articles can also be deemed reliable as they have been thoroughly analysed by experts before publication [4, 11, 12, 18–22].

The references not falling under the previous categories [7, 13–15, 24, 25, 27, 28] are those where information was gathered and validated by implementation, commonly solutions to coding problems, and then verified by comparing the resulting output to the expected result. By finding the same solution to a problem on several sites gave the solution more credibility. Also, by comparing solutions found on the internet to knowledge gathered during education at LTH, bad and non-functional sources or solutions could be promptly discarded without the need for extensive implementation before evaluation.

Information gathered from manuals and documentation from big multinational corporations [1, 6, 9, 10, 17] may also be considered trustworthy cause of the likelihood of extensive research, analysis, and tests in their products. For instance, Ublox being a leading manufacturer of GPS modules will most likely have trustworthy information in that domain.

Chapter 4

Implementation

The main task of this thesis was to implement a working prototype of a GNSS simulator with a focus on aiding training. The implementation required several steps including creating functions for computations, conversions, writing- and reading data, specific library and third-party program utilisation. Additionally, various methods for simulating interference were developed, along with a GUI (Graphical User Interface) to control the simulation and specify the scenarios to be simulated.

4.1 System Architecture

After a working prototype was created the system was redesigned with a modular approach, where each component had a defined responsibility. It was created this way for easy maintenance and reusability. It was divided into four different modules: GUI, DL (Data Layer), GNSS Calculation, and NMEA Generator. The GUI module had two tasks, where one was to get inputs from the GUI and send them to the simulation, and the second was to start and stop the simulation. In the data layer module, the main simulation classes existed, including those responsible for runtime management and control, such as the "runtime" and "controller" classes. The GNSS Calculation module houses all classes responsible for performing various calculations. A dedicated NMEA generator module was used for NMEA message generation. Later a module for the unit tests was added.

4.2 Models

For each broadcast file, there were multiple models to organise the data. A visual representation can be seen in Figure 4.1 and code examples in Appendix B.1-B.4. The broadcast model contains all the values from a broadcast message, such as message time and satellite ID. The satellite model compiles a list for a satellite, grouping all objects from the first model with matching IDs. Finally, the GNSS model generates a list of all the satellite objects categorized by the specified GNSS Type.

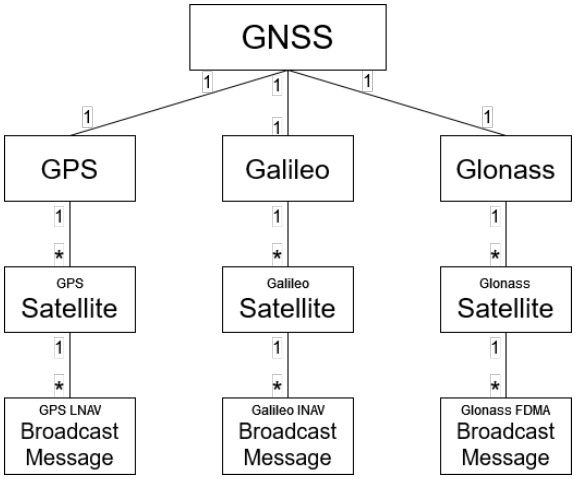


Figure 4.1: Visual Representation of the models

4.3 Parsing of Rinex File

At first, to get all the parameters needed for the satellite position computation, all the data had to be extracted from the Rinex file. Not every parameter was needed for the calculations, but for the sake of simplicity, all of them were extracted and the selection of which ones were required was done at a later stage. The choice of being able to use all the parameters could be useful if a version with more focus on accuracy were to be developed, for example utilising the clock correction parameters. Some difficulties were encountered due to the inconsistency of the files. If the satellite had been unable to send a value, that place was left empty creating the need for several edge cases in the parsing process.

The choice of going for version 4 of the RINEX format was made due to it having the least amount of inconsistencies. However, it also turned out to be the best documented version of RINEX.

An example of a broadcast message from a GPS satellite with PNR 6 can be seen in Figure 4.2. PRN stands for pseudorandom noise and is used to differentiate satellites from each other. [13]

```
> EPH G06 LNAV
G06 2024 01 01 12 00 00 4.595094360411e-04-1.250555214938e-11 0.000000000000e+00
     1.600000000000e+01 5.368750000000e+01 3.736941372894e-09 4.931821174148e-01
     2.838671207428e-06 3.086271579377e-03 6.595626473427e-06 5.153540245056e+03
     1.296000000000e+05-1.676380634308e-08-1.618394971961e+00-6.891787052155e-08
     9.894701503520e-01 2.650625000000e+02-8.254774330428e-01-7.843898158414e-09
     1.914365455291e-10 1.000000000000e+00 2.295000000000e+03 0.000000000000e+00
     2.000000000000e+00 0.000000000000e+00 3.725290298462e-09 1.600000000000e+01
     1.224180000000e+05 4.000000000000e+00
```

Figure 4.2: One broadcast message in the LNAV format

4.4 Calculating the Satellite's Position in ECEF

After successfully parsing the broadcast file, calculations to determine the position of a satellite based on the collected values were conducted. Upon calculating the satellite's position, the result was compared with a web tool that could display the historical positions of satellites [14]. For further certainty and validity, comparisons with a mobile app were also conducted [15].

To calculate the positions of satellites in accordance with the earth-centered and earth-fixed coordinates system there is a list of formulas to follow [4, pp. 224-225]. These calculations were applied on both GPS and Galileo, even though Galileo is based on the GRS80 ellipsoid, but as mentioned before, WGS84 and GRS80 ellipsoids can be considered the same [2]. Most of the variables used for the formulas are from the broadcasted parameters. However, some of them are physiological constants, such as the earth's rotation rate and the gravity constant for the WGS84 reference system.

Keplerian Parameters

$\sqrt{A}$ Square root of the semi-major axis

e Eccentricity

i_0 Inclination angle at reference time

Ω_0 Longitude of ascending node at reference time

ω Argument of perigee

$\overline{M_0}$ Mean anomaly at reference time

Perturbation Parameters

Δn Mean motion difference from computed value

$\dot{\Omega}$ Rate of change of right ascension

i Rate of change of inclination

C_{us} Amplitude of the sine harmonic correction term to the argument of latitude

C_{uc} Amplitude of the cosine harmonic correction term to the argument of latitude

C_{is} Amplitude of the sine harmonic correction term to the angle of inclination

C_{ic} Amplitude of the cosine harmonic correction term to the angle of inclination

C_{rs} Amplitude of the sine harmonic correction term to the orbit radius

C_{rc} Amplitude of the cosine harmonic correction term to the orbit radius

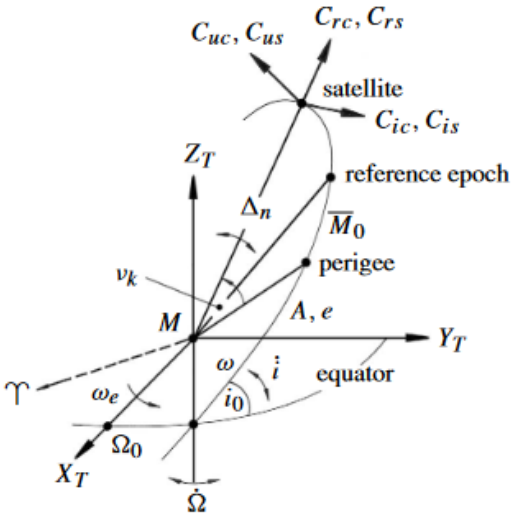


Figure 4.3: Parameters found in the broadcast message [4, p.225]

Formulas

	Explanation
$t_k = t - \Delta t_{oe}$	Time elapsed since reference epoch
$T = \dfrac{2\pi}{\sqrt{\frac{GM}{A^3}}}$	Satellite orbital period
$n_0 = \sqrt{\dfrac{GM}{A^3}}$	Computed mean motion
$n = n_0 + \Delta n$	Corrected mean motion
$\overline{M_k} = \overline{M_0} + nt_k$	Mean anomaly
$E_k = \overline{M_k} + e \sin E_k$	Kepler's equation of eccentricity
$\cos v_k = \dfrac{\cos E_k - e}{1 - \cos E_k}$	True anomaly
$\sin v_k = \dfrac{\sqrt{1 - e^2} \sin E_k}{1 - e \cos E_k}$	True anomaly
$\Phi_k = v_k + \omega$	Argument of latitude
$\delta u_k = C_{uc} \cos 2\Phi_k + C_{us} \sin 2\Phi_k$	Argument of latitude correction
$\delta r_k = C_{rc} \cos 2\Phi_k + C_{rs} \sin 2\Phi_k$	Radius correction
$\delta i_k = C_{ic} \cos 2\Phi_k + C_{is} \sin 2\Phi_k$	Inclination correction
$u_k = \Phi_k + \delta u_k$	Corrected argument of latitude
$r_k = A(1 - e \cos E_k) + \delta r_k$	Corrected radius
$i_k = i_0 + \dot{i} t_k + \delta i_k$	Corrected inclination
$X'_k = r_k \cos u_k$	Position in the orbital plane
$Y'_k = r_k \sin u_k$	Position in the orbital plane
$\Omega_k = \Omega_0 + (\dot{\Omega} - \omega_e) t_k - \omega_e t_{0e}$	Corrected longitude of ascending node
$X_k = X'_k \cos \Omega_k - Y'_k \sin \Omega_k \cos i_k$	Earth-centered X-coordinate
$Y_k = X'_k \sin \Omega_k + Y'_k \cos \Omega_k \cos i_k$	Earth-centered Y-coordinate
$Z_k = Y'_k \sin i_k$	Earth-centered Z-coordinate

Performing these calculations with a specified time parameter will result in a satellite's position in the ECEF coordinate system. Having a time variable that changes over time will give the perception of satellites moving as the simulation is running.

4.5 Converting Coordinates to ECEF

Due to the input values for the simulated receiver's location being in latitude, longitude, and altitude, a function for converting from geodetic coordinated to ECEF was necessary.
Latitude: ϕ, longitude λ and altitude: h

$$\phi = \frac{\phi \times \pi}{180} \tag{4.1}$$

$$\lambda = \frac{\phi \times \pi}{180} \tag{4.2}$$

In equation 4.3, f is the flattening for the WGS84 ellipsoid and the result of $\frac{a-b}{a}$ where a and b denotes the semi-major and semi-minor axes of the ellipsoid respectively. In this case, e^2 represents the square of the first numerical eccentricity of the WGS84 ellipsoid

$$e^2 = f(2 - f) \tag{4.3}$$

N' is the vertical radius of curvature, which is the distance of the orthogonal vector from a geodetic coordinate at the surface of the ellipsoid to the Z-axis.

$$N' = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \tag{4.4}$$

$$x = (N' + h) \cos \phi \cos \lambda \tag{4.5}$$

$$y = (N' + h) \cos \phi \sin \lambda \tag{4.6}$$

$$z = ((1 - e^2)N' + h) \sin \phi \tag{4.7}$$

Performing this will give the XYZ coordinates with reference to the WGS84 ellipsoid [16]. In the equations 4.5, 4.6, and 4.7 above, h denotes the height above the ellipsoid.

4.6 Updating the Parameters

A rough rule of thumb is that ephemeris data are only valid for up to two hours after t_{oe} (Time of Ephemeris) due to the position error growing exponentially. After two hours it is considered too inaccurate. [17] Therefore, during a simulation, the parameters used for the position calculation had to be updated. The interval in GPS broadcast messages is every two hours and for Galileo it is every ten minutes. However, it was observed that on some occasions, a message failed to transmit, resulting in a gap during which no broadcast message was delivered from a satellite. To address this issue, the program had to check the last broadcast from each satellite and calculate the position based on the received values and the time elapsed since the message was transmitted.

4.7 Predicting the Satellite's Orbit

The broadcasts occur at specific intervals, causing a 'jumping motion' for the satellites, as their positions are only updated when a new message is received. To address this issue, predicting the satellite's future position became necessary. This prediction involved utilising the same formula used to calculate the satellite's position but with the given epoch t adjusted to include the time between the transmission time and the predicted time.

4.8 Visible Satellites

To calculate which satellites are theoretically visible from a location the known XYZ coordinates of the specific satellite and the simulated receiver's XYZ coordinate were used. The criteria that had to be fulfilled for a satellite to be considered visible from a given location, was that the satellite had an elevation between $0°$ and $90°$, shown in Figure 4.4. However, these are under perfect conditions and strictly theoretical. In reality, there are several factors impacting the visibility of a satellite such as clouds, mountains, and buildings. In Figure 4.4 you can see a visualisation of the scenario utilised in the program. The rotation of the earth is represented by ω_e making the visible area the top half of an infinitely large sphere.

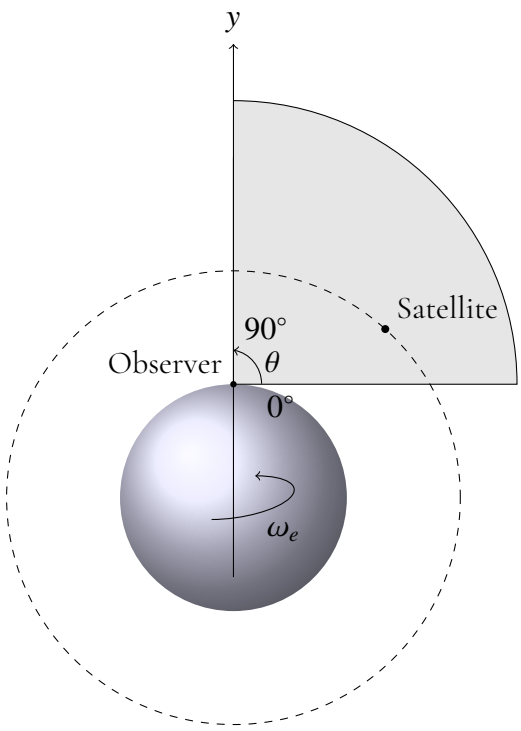


Figure 4.4: Satellite Visibility where θ represents the elevation interval from $0°$ to $90°$ degrees.

4.9 Azimuth & Elevation

To be able to see which satellites were visible the local azimuth and elevation had to be calculated. This was done using two points in the ECEF coordinate system, one for the receiver position and one for the satellite position. Local in this case refers to that the azimuth and elevation values are as if you were standing at the receiver's coordinates. Therefore, if the elevation is 90° the satellite is in zenith from your point of view. The necessary calculations can be seen in Table 4.1 below and the result will give both azimuth and elevation in degrees. These calculations were especially necessary for the visualisation of the satellite position.

Both the calculation for finding the elevation and the calculation for finding the azimuth used equation 4.8 through 4.10 for determining the vector between the satellite and the receiver.

$$dx = x_1 - x_2 \tag{4.8}$$

$$dy = y_1 - y_2 \tag{4.9}$$

$$dz = z_1 - z_2 \tag{4.10}$$

Table 4.1: Calculations for Local Azimuth and Elevation

Elevation Calculations	Azimuth Calculations
$P = x_1 \cdot dx + y_1 \cdot dy + z_1 \cdot dz$ (4.11)	$Cos_n = (-z_1 \cdot x_1 \cdot dx) - (z_1 \cdot y_1 \cdot dy) + (x_1^2 \cdot y_1^2) \cdot dz$ (4.17)
$n_1 = \sqrt{x_1^2 + y_1^2 + z_1^2}$ (4.12)	$Cos_d = \sqrt{(x_1^2 \cdot y_1^2) \cdot (x_1^2 + y_1^2 + z_1^2) \cdot (dx^2 + dy^2 + dz^2)}$ (4.18)
$n_2 = \sqrt{dx^2 + dy^2 + dz^2}$ (4.13)	$\cos\phi = \frac{Cos_n}{Cos_d}$ (4.19)
$E_c = \frac{P}{n_1 \cdot n_2}$ (4.14)	$Sin_n = (-y_1 \cdot dx) + (x_1 \cdot dy)$ (4.20)
$E_d = \arccos(E_c) \cdot \left(\frac{180}{\pi}\right)$ (4.15)	$Sin_d = \sqrt{(x_1^1 \cdot y_1^2) \cdot (dx^2 \cdot dy^2 \cdot dz^2)}$ (4.21)
Elevation $= 90° - E_d$ (4.16)	$\sin\theta = \frac{Sin_n}{Sin_d}$ (4.22)
	Azimuth $= \arctan(\frac{\sin\theta}{\cos\phi}) \cdot \left(\frac{180}{\pi}\right)$ (4.23)

4.10 Calculating Dilution of Precision

When calculating the DOP (Dilution of Precision), a minimum of four satellites was required. It was however possible to calculate the DOP values with more than four satellites. Utilising more than four satellites will most often result in better DOP values. In Table 4.2 it is possible to see what values of the dilution of precision are considered ideal, good, etc. In the calculations, the x, y, and z are the coordinates of the receiver, and x_i, y_i, and z_i are the individual coordinates of the i-th satellite used for the calculation.

$$R_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - x)^2} \tag{4.24}$$

$$A = \begin{bmatrix} \frac{x_1-x}{R_1} & \frac{y_1-y}{R_1} & \frac{z_1-z}{R_1} & 1 \\ \frac{x_2-x}{R_2} & \frac{y_2-y}{R_2} & \frac{z_2-z}{R_2} & 1 \\ \frac{x_3-x}{R_3} & \frac{y_3-y}{R_3} & \frac{z_3-z}{R_3} & 1 \\ \frac{x_4-x}{R_4} & \frac{y_4-y}{R_4} & \frac{z_4-z}{R_4} & 1 \\ \frac{x_i-x}{R_i} & \frac{y_i-y}{R_i} & \frac{z_i-z}{R_i} & 1 \end{bmatrix}$$

A multiplication with the transposed A-matrix is done and the inverse gives the matrix Q.

$$Q = (A^T A)^{-1} \tag{4.25}$$

$$Q = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & \sigma_{xi} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} & \sigma_{yi} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 & \sigma_{zi} \\ \sigma_{xi} & \sigma_{yi} & \sigma_{zi} & \sigma_i^2 \end{bmatrix}$$

From the Q-matrix it is possible to extract the desired parts from the dilution of precision, which in this case was the HDOP and the VDOP. The equation for PDOP is the joint values of the horizontal dilution of precision and the vertical dilution of precision. The HDOP is in turn the sum of the precision error in the East-West line and the North-South line. From the Q-matrix it would also be possible to extract the GDOP (Geometric Dilution of Precision) as well as the TDOP (Time Dilution of Precision), those were however not relevant to the work in this thesis.

$$PDOP = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \tag{4.26}$$

$$HDOP = \sqrt{\sigma_x^2 + \sigma_y^2} \tag{4.27}$$

$$VDOP = \sqrt{\sigma_z^2} \tag{4.28}$$

Table 4.2: Rating of DOP values [18]

DOP Value	Rating
< 1	Ideal
$1 - 2$	Excellent
$2 - 5$	Good
$5 - 10$	Moderate
$10 - 20$	Fair
> 20	Poor

To calculate the position accuracy from the currently visible satellites the value of PDOP together with the receiver's GPS position error is multiplied. The GPS position error in civilian receivers is usually around three which is the value used in the program.

$$\text{Position accuracy} = \text{PDOP} \cdot \boldsymbol{GPS_{error}}$$

4.11 SNR and Field Survey

It was not possible to gather an SNR (Signal to Noise ratio) value from the broadcast messages and therefore another approach was needed. To get the simulator as realistic as possible it was necessary to still find a way to get an accurate value that could be put in the NMEA strings. After determining that the first implementation for this was ineffective, where the value was a random integer between 0 and 50, a better implementation for the generation of SNR-value was achieved by conducting a small field study.

By using equipment supplied by Saab, real-time messages/measurements were captured outside the office in Helsingborg. These messages were then examined by sampling elevations and SNR-values of 25 random satellites. With this data gathered, a function to generate a simulated SNR value was created in order to generate random numbers, where the numbers followed a normal distribution within a specified range. This was achieved by utilising a mean value and a standard deviation together with the box-muller method [19]. If the elevation is less than **10°** it generates a SNR-value between 0-25, using a median of 14 and a standard deviation of 4. In the case of an elevation greater than **10°**, it instead generates a random SNR-value between 10 and 45, using a median of 25 and a standard deviation of 5. The sample elevation value and respective SNR value can be seen in Table 4.3 below.

Table 4.3: Data from mini survey.

Data from field survey			
Elevation < 10°	**SNR**	**Elevation > 10°**	**SNR**
6	13	57	34
7	19	41	30
10	11	56	38
5	13	43	36
6	8	42	9
10	16	19	8
4	0	20	11
10	13	57	11
6	26	56	23
10	22	19	29
Average	**14.1**	43	31
		13	30
		19	27
		25	31
		20	26
		Average	**24.9**

4.12 Interference

Initially, various ideas and thoughts were discussed on how it would be possible to simulate interference ranging in different complexity levels. It felt necessary to create an implementation that would have the purpose of the simulator at focus, which was the training. But on the other hand, a "too simple or too inaccurate" implementation would not be enough either. At a later stage, well into the implementation, a different approach emerged after discussions with the supervisor from LTH. This sub-chapter will discuss this further.

4.12.1 Different Jamming Techniques

In the domain of jamming, there are various methods available to cause interference. One such method is spot jamming, where all the power is concentrated on a single frequency with the goal to "overload" the receiver making it unable to process incoming signals. Another method is sweep jamming, and similar to spot jamming, all the power is focused on one frequency. However, sweep jamming continuously changes frequencies and "sweeps" back and forth over the frequency band to cover the entire bandwidth of the target communication band. In this case, it would be the L1 band at 1.575 GHz. For simplicity in this thesis, spot jamming was used and the assumption that the communication between the satellites and receiver was only being carried out on one specific frequency was taken. Typically, jamming is achieved by directing a cone-like beam in a specific direction to minimize interference with unintended systems. However, for the purpose of not making the simulation too complex, jamming is executed equally in all directions.

4.12.2 Jamming Discussion

A first idea that was brought up during early discussions was to simply remove a random or a specified amount of the satellites visible from the receiver location. This however fell under the "too simple or too inaccurate" category even if it could be used to potentially create interesting scenarios from a training perspective.

The first version for simulating interference that was implemented was for the user to be able to remove a specified range of the azimuth where satellites would disappear. One scenario could be that satellites in the azimuth range between 40 and 85 degrees from the receivers would no longer be visible. This could also be further extended with implementation for controlling the elevation, where for example satellites in the range from 20-35 degrees in elevation were removed. This implementation could also allow for joint functionality, removing only the satellites that were between 250 and 320 degrees azimuth on an elevation of 25-60 degrees. However, because this model did not give the option of setting the jammers position, it was not used in the final product.

Therefore, after discussions with employees at Saab on how to create a version that would make the most sense from a training perspective, a more complicated version was implemented. It should however be noted that this model is strictly fictional and is not correlated to how a real jammer would act. Despite this, the model was one of the two methods implemented in the final version of the program.

The other method of modeling interference in the final version was implemented after a meeting with the supervisor from LTH, with the goal of creating a more realistic model. This model relied more on the actual physics of electromagnetic waves thus resulting in a more realistic representation of jamming.

One aspect of the modeling of interference that in the early stages went untouched was to take into consideration that the interference or jamming would most likely occur at sea. The effectiveness of jamming is very different in a suburban environment compared to at sea, where the obstructions of buildings, land, and mountains are very limited. In [20] it is shown that in a suburban environment, an estimated effective range of a 1000W jammer is approximately 2 km if the goal is to hinder a receiver from tracking satellites. In free space however, which is close to that experienced at sea, the range was instead much longer with the possibility to hinder tracking at a distance of 140 km. Because the jamming signal is traveling so far, the biggest limitation on jamming over the sea is the line of sight, as the line of sight between two antennas both 25 meters above the sea is only approximately 40 km, far less than the potential jamming range.

4.12.3 Fictional Model

The fictional model implements interference by giving the jammer a specified location and range, where the interference was modeled as a sphere with the center at the jammer's location and the radius of the sphere representing the jammer's range. Now, if the vector from the receiver location to the satellite position intersected the sphere, that specific satellite would no longer be considered visible. This method allowed for the elevation of the satellites to be considered as well as the azimuth, unlike the first iteration. Now a satellite with high elevation would still be visible due to the shape of the sphere. A sketch of this can be seen in Figure 4.5.

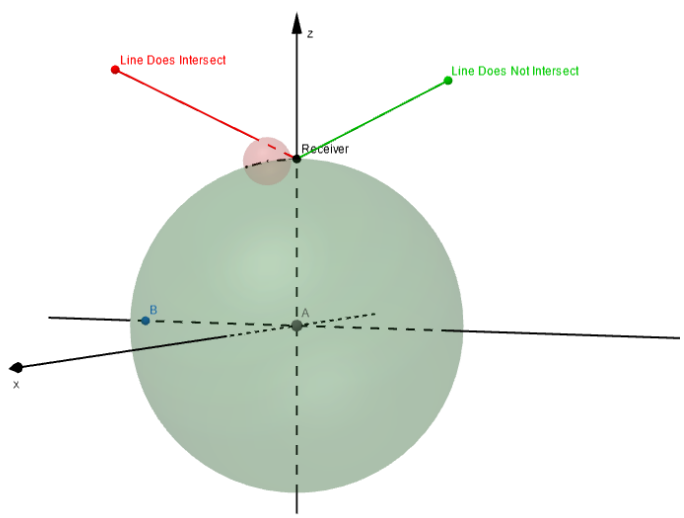


Figure 4.5: Scenario of two cases, one intersects, one does not.

For this implementation a handful of calculations were necessary. First, calculations for the vector from the receiver to the satellite and the vector from the jammer to the satellite, using equation 4.29.

$$\vec{v} = (x_1 - x_2, y_1 - y_2, z_1 - z_2) \tag{4.29}$$

Then the length of the line segment from the receiver to the satellite was calculated using equation 4.30.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{4.30}$$

The unit vector of the line segment between the receiver and the satellite was obtained by equation 4.31, where rs_i denotes the x, y, and z of the receiver to the satellite vector.

$$\vec{uv} = (\frac{rs_1}{d}, \frac{rs_2}{d}, \frac{rs_3}{d}) \tag{4.31}$$

Then the dot-product between the unit vector and the vector from the jammer to the satellite was calculated to get the closest points to the jammer on the line, between the receiver and the satellite:

$$\text{Dot product} = \vec{v_1} \cdot \vec{v_2} = x_1 \times x_2 + y_1 \times y_2 + z_1 \times z_2 \tag{4.32}$$

1. If the dot product is less than or equal to 0, it means the closest point lies beyond the satellite (before the line segment begins). In this case, the closest point is the satellite itself.

2. If the dot product is greater than or equal to the length of the line segment, it means the closest point lies beyond the receiver (after the line segment ends). In this case, the closest point is the receiver itself.

3. If the dot product is within the range (0, length from receiver to satellite), it means the closest point to the jammer lies within the line segment. To calculate the coordinates of this point formula 4.33 is used:

$$\text{Closest point} = (x_1 + x_2 \times (\vec{v_1} \cdot \vec{v_2}), y_1 + y_2 \times (\vec{v_1} \cdot \vec{v_2}), z_1 + z_2 \times (\vec{v_1} \cdot \vec{v_2})) \tag{4.33}$$

Then it was just a matter of checking whether or not the distance between the jammer and the closest point was less than or equal to the jammer's radius. If the distance was less than the jammer radius, the line intersects the sphere thus causing "interference" and the specific satellites will be removed from the sky.

Since electromagnetic waves would pass through each other unchanged, and no line of sight were considered in this model, it does not reflect reality in any way and was, as noted before, just a "fictional" implementation. Although, after speaking with employees at Saab it serves as a decent solution with respect to the goal of training, as the user can easily impact DOP values as a training leader by increasing the sphere. Would you increase the sphere to be big enough, it would result in the effect of total loss of satellite communication.

4.12.4 Signal to Noise Ratio Model

The goal of the signal to noise ratio model was to be able to estimate the signal strength from the satellite as well as the signal strength from the jammer, given that the jammer was on. This model required the receiver and jammer to have a line of sight to each other, with consideration for the curvature of the earth. The need for the receiver and jammer to have a line of sight is due to the fact that high-frequency waves, such as in GPS communication do not bounce on the ionosphere, thus making it impossible to jam from far away utilising sky wave propagation. [21]

This approach is relevant since most of the simulations will take place at sea level, where for the most part, nothing is interfering with the signals more than potential interference and the limit of visibility beyond the horizon. For calculating the line of sight various methods exist where three were found and compared. The first approach relies on determining how far away the horizon is based on the altitude above the sea level. That can be calculated by using $\sqrt{2\pi rh}$ where h denotes the height above the sea level, and r is the radius of the earth. If the combined horizon distance of two antennas is equal to, or greater than the distance between the receiver and the jammer there is a line of sight.

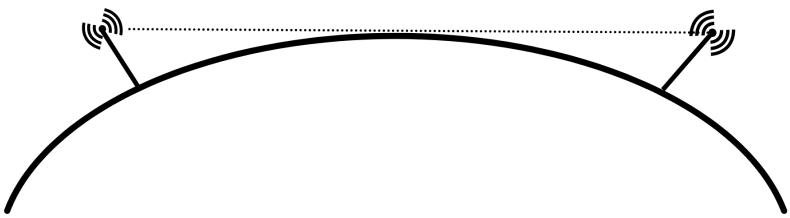


Figure 4.6: Explanation of Line of Sight

Another way is to create a WGS84 ellipsoid representation and see if the vector between the two positions intersects the ellipsoid, if so, there is no line of sight. The last, and the one used is the formula for the radio horizon between two antennas given by $2.2(\sqrt{h_1} + \sqrt{h_2})$ where h_1 and h_2 is the height of the two antennas, with the result in nautical miles. They all, however, gave approximately the same values for the same scenario.

For calculating the power received at a distance d between two antennas, which in this case is between the satellite and the receiver, and between the jammer and the receiver, Friis simple transmission formula [22] shown in equation 4.34 was used.

$$P_r = P_t G_t G_r (\frac{\lambda}{4\pi d})^2 \tag{4.34}$$

Here, P_r denotes the power received in watts, P_t is the power transmitted, G_t and G_r are constants of the two antennas. λ is the wavelength of the signal and d the distance between the antennas. λ for this case will be calculated from $\lambda = \frac{c}{f}$ where c is the speed of light and f the frequency, which in this case is the L1 band transmitting at 1.575GHz resulting in a wavelength of 0.19m.

With this, both a signal from the satellite, as well as noise from the jammer could be obtained. To calculate the SNR equation 4.35 can be used.

$$SNR = 10\log\left(\frac{P_{rs}}{(P_{rj} + P_n)}\right) \tag{4.35}$$

where P_{rs} is the received signal from the satellite, P_{rj} is the received signal from the jammer and P_n is the surrounding noise from the antenna, the environment, and more.

To calculate the received signal at the earth's surface from a GPS satellite with an altitude of 202000km (height of a GPS satellite in zenith) above the earth's surface the use of formula 4.34 is needed. Here the assumption of antennas having a constant of one (0dB gain) is made. That is $G_r = 1$ & $G_t = 1$ and the approximation that the satellite transmitting at 200W would get the following:

$$P_t = 200\left(\frac{0.1903444178}{4\pi 20000000}\right)^2 = 1.147 \cdot 10^{-16} W$$

or expressed in decibels, -159.40 dBW. The conversation to dBW is $10\log(P_r)$ with P_r in watts. Using the same formula for calculating the power received by the jammer at a realistic distance of 40km away would result in the following:

$$P_t = 1000\left(\frac{0.1903444178}{4\pi 40000}\right)^2 = 1.43 \cdot 10^{-10} W$$

which is approximately -98.4dBW. With both values in dB, it is possible to calculate the signal-to-noise ratio with $SNR = -159.4 - (-98.4 + P_n) = -61$dB, P_n being small enough to be negligible.

As a rule of thumb, an SNR value lower than 15 would be too low for the GPS, and would therefore not be usable. With the result of -61dB, this model would consider the receiver jammed. However, reducing the transmitted power to 0.1W would still result in an SNR of -21dB, also indicating the receiver to be jammed. With a jammer at 0.01W causing the receiver to be deemed jammed, this model was determined to be unrealistic, thus limiting this approach's implementation.

4.12.5 Jamming to Signal Ratio Model

It was discovered that GPS receivers are fairly resistant against jamming, which was the reason the previous approach did not work. Because of this, it was not possible to model the jamming signal as noise, and therefore looking at SNR to see if the receiver was jammed, was not a viable option.

According to [20], in a "free space" environment a difference in decibel of 47dB was needed in order to stop the tracking of satellites in the case where a connection had already been established. However, the threshold was 27dB if no connection had been previously made between the satellite and the receiver. With this knowledge in mind, the new method instead focused on the jamming-to-signal ratio shown in equation 4.36, where P_{rj} is the power received from the jammer and P_{rs} is the power received from the satellite, both in decibels.

$$\frac{J}{S} = P_{rj} - P_{rs} \tag{4.36}$$

The calculation for the power received uses the previous formula for microwave propagation, equation 4.34. If the jamming-to-signal ratio is lower than 47dB the possibility for the receiver to keep tracking the satellites is still possible. For simplicity, only the threshold of 47dB was considered. This method still assumes the antenna constants to be one.

This model provided a more realistic result compared to the signal-to-noise ratio approach. A 1000W jammer would still jam the receiver, but the 0.1W jammer would not accomplish the same task. However, even though the receivers are more resistant to jamming than first thought, the conclusion was that the receivers would be easy to jam given the potential of high-power jammers on a ship. Therefore, with the assumption of "free space", it results in a scenario where it will, in almost all cases, be the radio horizon determining if a receiver is jammed or not. A 1000W jammer, resulting in a possible jamming radius of 140 km is not unrealistic for a jammer placed on a ship. However, for the radio horizon to be that large, the jammer would need to be at an altitude of 1200 m. This leads to the jamming range almost always extending beyond the line of sight. Consequently, it was estimated that jamming is possible whenever there is a line of sight. Thus the implementation for this functionality only needed to check if there was a line of sight and if the jammer was on. It was implemented this way for easier maintainability and future development.

With the implementation of this model, the simulator consists of two different models for simulating interference, this new one, focusing on a more realistic approach utilising jamming-to-signal ratio. The second one is the previous one utilising a sphere and vectors, both with their respective advantages.

4.13 Altitude

A value for the altitude is needed to calculate the line-of-sight between the jammer and receiver, and since a solution for calculating the altitude based on the gathered data was not found, another solution was explored to get some approximate data. At Lantmäteriet [23] it is possible to download a DEM (Digital Elevation Model) for an area of interest with a resolution of 50x50 meters. Saab itself did not need a solution for generating an altitude, only the possibility for the input of an altitude into the simulation controller. However, for the sake of this thesis and for the line-of-sight calculation, altitude data was collected and used. The DEMs store the altitude in meters above the sea level. When downloading the files they come in a .tif format. For fast implementation, a library named "GeoTiffElevationReader" was used [24]. This made it possible with a couple of lines to retrieve the altitude at a certain location. All the library-function required was an input of coordinates in decimal degrees.

The digital elevation models from Lantmäteriet used plane coordinates making a conversion from decimal degree coordinates to plane coordinates necessary. An explanation of this conversation was not found, however, a library with this functionality was integrated. With the library "MightyLittleGeodesy" [25] it was possible to convert from geodetic coordinates to plane coordinates, therefore the altitude data could be collected.

Another solution was later found, with coverage for the entire world map, and was ultimately implemented at the end. At NASA's Earth data site [26] it is possible to download DEMs for an area of interest anywhere in the world. This approach also resulted in an increase in the resolution from 50x50 to 30x30 meters. Another improvement with this solution is that it was possible to input the decimal coordinates directly into the GeoTiffElevationReaders function, therefore the need for a conversation was not necessary resulting in a great improvement regarding the readability and maintainability of the code.

4.14 Generating NMEA-Strings

To be able to use VisualGPSView the data had to be packaged as NMEA-strings. By understanding the structure of the strings [9] and the different aspects that had to be considered, it was possible to create functions for string generation. The three different NMEA strings necessary for this case were the GGA, GSA, and the GSV. It was not possible to gather all the required data for the NMEA message from the previously computed data. Thus making it necessary to fabricate some of the values to get it working. The SNR number is for example such a value, which fabrication was previously explained. Other examples are different values for various mode selections.

For coordinates expressed in latitude and longitude, various formats exist. Dº M' S" (Degrees Minutes Seconds), Dº M. M' (Decimal Minutes), and D.Dº (Decimal Degrees) are all different ways of expressing a location in latitude and longitude. The GGA message requires the format DMS while the input from the GUI is expressed as DD. To convert from DD to DMS, formulas 4.37 & 4.38 were used.

$$D = \text{integer of DD} \tag{4.37}$$

$$MM = (|DD| - |D|) \cdot 60 \tag{4.38}$$

The GGA message contains among other things information about the position of the receiver in latitude and longitude, UTC time of the position, quality indicator, the number of satellites in use, HDOP, and receiver height above the ellipsoid and above the geoid.

The GSA message contains information about modes used, PRN number for satellites used, and the DOP values HDOP, VDOP, and PDOP. A problem with the GSA message is that it only takes 12 satellites, in the program which means that only the first 12 will get sent in the GSA message.

For the GSV message there can be multiple messages connected to each GGA and GSA message and the messages contain information about the individual satellites. Among other things, information about azimuth, elevation, SNR, and PRN are sent through the GSV message. Each GSV message can contain information about up to four satellites.

```
$GNGGA,120015.00,562.8020000000000778000,N,01241.6580000000000001000,E,1,18,0.6,3.636,M,-15,M,0,,*6E
$GPGSA,A,3,05,07,08,09,11,13,14,15,16,18,20,27,1.1,0.6,0.9*34
$GPGSV,4,1,13,05,34,215,26,07,62,79,22,08,2,74,15,09,27,102,29*4A
$GPGSV,4,2,13,11,15,228,24,13,27,272,25,14,14,161,21,15,1,284,15*4A
$GPGSV,4,3,13,16,7,23,14,18,7,345,13,20,59,242,27,27,8,44,15*40
$GPGSV,4,4,13,30,70,183,28*4F
$GLGSA,A,3,07,10,11,30,33,,,,,,,,1.1,0.6,0.9*24
$GLGSV,2,1,5,71,36,106,29,74,23,317,21,75,13,343,30,94,40,68,25*6F
$GLGSV,2,2,5,97,13,237,26*6E
```

Figure 4.7: Result of the generation of NMEA strings

In Figure 4.7 the "GN" prefix is used for the GGA message to indicate the use of multiple constellations in the GSA and GSV messages. Shown is also the "disguised" Galileo message with the prefix "GL" instead of the normal "GA", which is done for the third-party program to be able to read the string.

4.15 Third Party Program Utilisation

A choice to use third-party programs for visualization was made to avoid the necessity of developing the visualization component. It felt unnecessary to spend time developing a visualisation tool when free versions are available that meet the required needs.

4.15.1 VisualGPSView

The VisualGPSView tool [27] made it possible to visualize satellite locations and all relevant parameters of interest, such as PDOP, HDOP, and VDOP, without the need for any implementation, this can be seen in Figure 4.8. It also showcases the SNR (Signal-to-Noise Ratio) for each of the visible satellites. VisualGpsView works by reading NMEA strings through a serial port in order to visualise GPS and Glonass, as can be seen in Appendix C.1. Due to its specification for only reading GPS and Glonass, it was necessary to "disguise" the Galileo NMEA strings as Glonass for visualisation purposes.

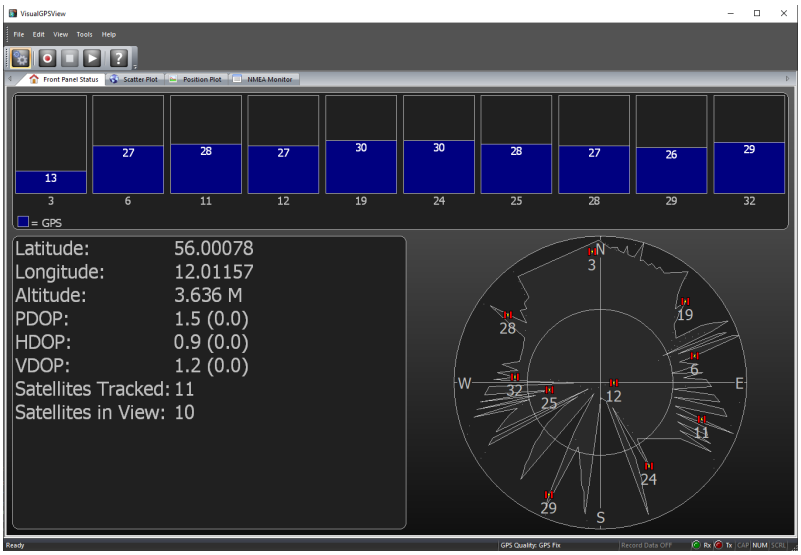


Figure 4.8: VisualGPSView

4.15.2 Virtual Serial Port

Because VisualGPSView reads input through a serial port, and it was not possible to write directly from the program to VisualGPSView, a virtual COM port was needed. Free Virtual Serial Ports by HDD Software [28] made it possible to write to COM1 from the simulator's control class, and the emulator then sent that data to COM2 which was set as input to VisualGPSView. An illustration of this can be seen in Figure 4.9. This made for seamless visualisation of the data produced by the application in "real-time".

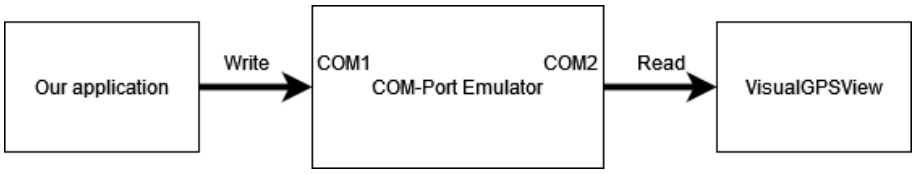


Figure 4.9: Virtual COM ports

4.16 GUI

The Graphical User Interface that has been developed was created as a Windows Forms application in C#, and can be seen in Figure 4.10. The GUI serves as a convenient and user-friendly platform for acquiring inputs essential for the simulation process, together with the Visual-GPSView they make visualising how the simulation function and its output possible. Within the GUI, users can specify inputs crucial for the simulation, such as what type of GNSS should be used, the start time for simulation, what file to use, what position the receiver should be placed at, and interference configuration.

Implemented in the GUI, a "Terminal" feature has been integrated to display various messages including "Help Messages" and "Error Messages," along with notifications regarding the start and stop of the simulation. Additionally, there is an output section designed to provide users with real-time updates on the current position accuracy. This section is regularly refreshed throughout the simulation process.

Figure 4.10: GUI for the simulator.

4.17 Unit Tests & Documentation

Four unit tests were constructed for continuous verification of the functionality of calculations. The first test was for satellite visibility, with an easy-to-verify scenario. The second test was for the conversion between geodetic coordinates to ECEF coordinates. The third test was for the functionality of checking if the vector between the receiver and the satellite intersects the sphere representing the jammer, the scenario tested can be seen in Figure 4.11. The last test was for the calculations on distance between two positions.

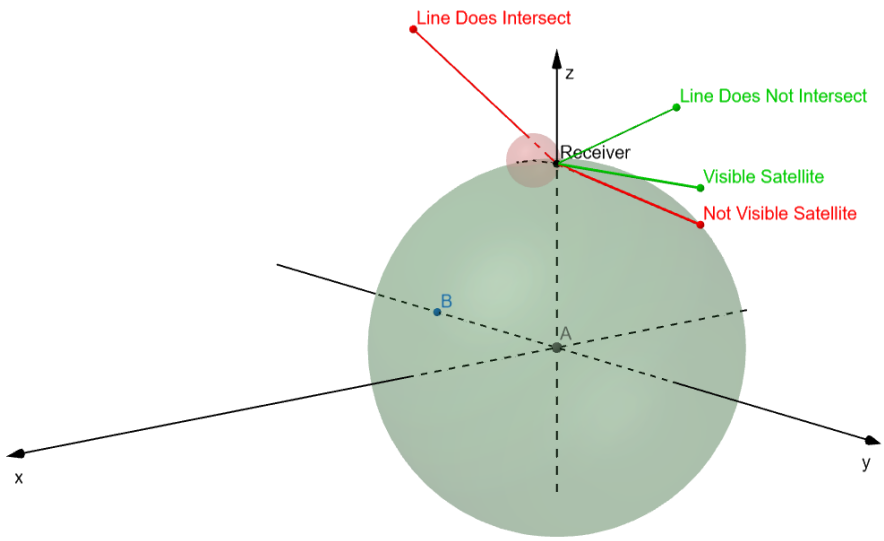


Figure 4.11: Scenario tested for intersection and visibility.

A big emphasis was also laid on documentation, making the implemented functions easy to understand for anyone needing to either use them or modify them in the future. Documentation according to the standards in C# was made, allowing for specialised programs to read the documentation if that need would ever arise. An example of this can be shown in Figure 4.12 below.

```
/// <summary>
/// Calculate the checksum for the NMEA-Strings.
/// For each character, it performs a bitwise XOR operation with the current checksum.
/// </summary>
/// <param name="message">The string to be calculated.</param>
/// <returns>The checksum for this message string.</returns>
```

Figure 4.12: Example of documentation

4.18 Library Options

For some of the functionality various libraries existed, however, they were for the most part too complicated and would require too much time to delve into. Therefore, it was easier to implement the desired functionality ourselves, and it minimized the risk of wasting time reading up on libraries that still would end up being unused.

4.18.1 Conversions

For conversions and position calculations, a library called AstroLib was at an early stage considered. The use of a library would limit the possibility of errors compared to own implementations of the calculations. However, the library was extensive, and required a lot of "reading up on" and the idea was discarded early on.

4.18.2 Rinex Parser

Some time was spent at the beginning trying to find a parser for the Rinex files, however, the ones found were either for a different version, in a different language, or only had support for parsing GPS broadcast messages, and not multi-constellation support. Because of this, it was easier to implement it ourselves, and this also allowed for more adaptability and modifications when needed.

4.19 Accuracy of Position Calculations

The accuracy of the satellite positions obtained from the calculations is not perfect, as it did not account for the clock correction parameters broadcasted in the broadcast message. In an optimal scenario, this would be used to tweak the values of the time parameters. Additionally, no calculations were made regarding potential leap seconds.

Any stratospheric delays are not taken into consideration either. However, for this use case and the relevancy of the training simulator, it would have been unnecessary time spent trying to account for these things.

Chapter 5

Results

The final product is a locally run program designed to function as a prototype GNSS simulator for training purposes. It serves as a test and evaluation for future GNSS simulation within Saab's Naval SE simulator. The program effectively addresses the primary challenge of persistent simulation values outlined in the problem, while also making strides in introducing interference simulation capabilities. Throughout the development of this program, the questions posed at the beginning of the thesis were addressed as follows:

- **How do you create a GNSS training simulator that outputs an approximate position?**
  To create a GNSS training simulator capable of outputting approximate position calculations, first, a calculation of the dilution of precision has to be done. This value, along with the receiver's GPS error, makes it possible to determine an approximate position.

- **How do you calculate and predict a satellite position from an ephemeris file?**
  To be able to calculate a satellite's position, the parameters from the ephemeris file first have to be extracted, which can be done with a parser program. After that, calculations following those explained in section 4.4 will provide the possibility to obtain a current position, as well as a predicted future position for a satellite.

- **How do you calculate dilution of precision from a group of satellites?**
  For a group of four or more satellites, the dilution of precision can be calculated using the formulas described in section 4.10. By assessing the distribution and quality of the satellite positions, a value for the dilution of precision can be derived. With the derived value it is then possible to extract the desired values for HDOP, VDOP, and PDOP.

- **What are the possibilities for simulating interference and in what ways can it be implemented?**
  There are various options for simulating interference as previously discussed in section 4.12, each with different levels of physical or realistic execution. A conclusion was also taken that a more realistic implementation does not directly translate to a better training simulation, as there are other aspects to consider when the focus is on training.

5.1 GUI Design

The design of the GUI has the mindset of being easy to understand and use, therefore it is only one page for less confusion, so the training leader can use it without hours of training beforehand, the full GUI can be seen in Figure 5.1. Additionally, the colors follow the main colors of Saab [29] to make it align with their other work. The GUI consists of three main parts:

- Receiver Configuration
- Interference Configuration
- Simulation

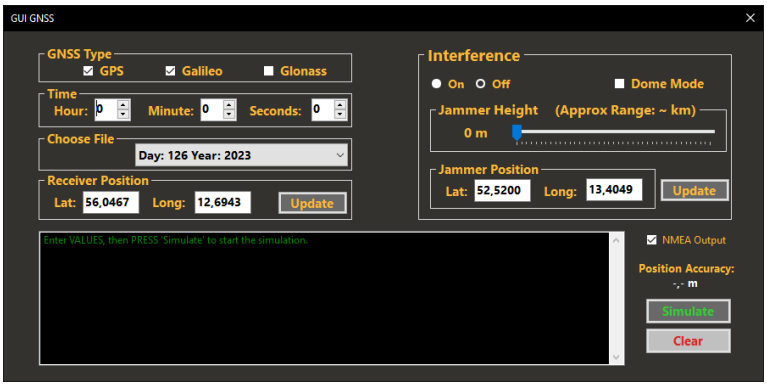


Figure 5.1: GUI for the simulator

5.1.1 Receiver Configuration

The "Receiver Configuration" consists of a couple of check buttons, text boxes, a drop-down menu, and an update button. These are mainly used to get inputs for the receiver to use during simulation.

"GNSS Type" seen in Figure 5.2 allows users to choose what type of GNSS they want to use for their simulation, the choices are GPS, Galileo, and Glonass. However, currently, Glonass is not implemented.

Figure 5.2: "GNSS Type" GUI

"Time" seen in Figure 5.3 enables users to set the simulation start time. It follows the format of HH:MM:SS (Hours:Minutes:Seconds) in a 24-hour clock, not allowing more than 23 hours and 59 minutes and seconds.

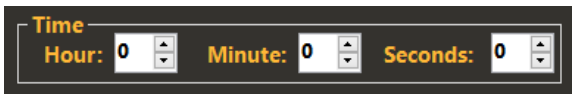


Figure 5.3: "Time" GUI

"Choose File" seen in Figure 5.4 creates the possibility for users to choose what broadcast file to use from a drop-down menu. The list contains all the broadcast files in the "Broadcast" folder, and it is presented to the user with which day of the year(1 - 365) and what year the broadcast file contains.

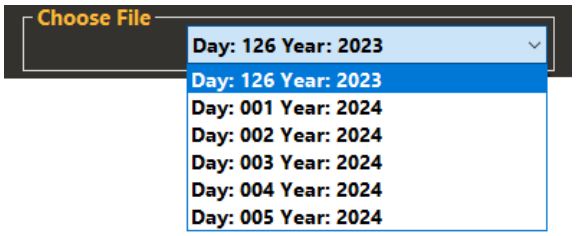


Figure 5.4: "Choose File" GUI

"Receiver Position" seen in Figure 5.5 gives the users the option to set latitude and longitude where they want the receiver position to be in the simulation. The format for the coordinates is DDD, MMSS (Degree, Minutes Seconds) and values for latitude are between -90 and 90 and longitude between -180 and 180. The "Update" button allows the users to update the receiver position while the simulation is running.

Figure 5.5: "Receiver Position" GUI

5.1.2 Interference Configuration

The "Interference Configuration" consists of two different layouts depending on if the "Dome Mode" is checked or not. Each layout consists of two radio buttons, a checkbox, a slider, two text boxes, and an update button. The values from these are used to control the position and strength of the jammer for the simulation if it is toggled on.

"On-Off" seen in Figure 5.6 allows the users to toggle if they want to use the jammer in the simulation.

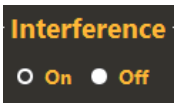


Figure 5.6: "On-Off" GUI

"Dome Mode" seen in Figure 5.7 allows the users to change the interference mode from the "Jamming to Signal" mode, explained in 4.12.5, to the "Dome Mode", explained in 4.12.3.

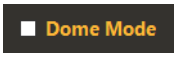


Figure 5.7: "Dome Mode" GUI

"Jammer Height" seen in Figure 5.8 provides users with the option to select the elevation of the jammer above sea level and is only visible when not in "Dome Mode". Based on this height and the elevation of the receiver, an approximate range value is calculated. The values on the slider ranges between 0m and 200m.

Figure 5.8: "Jammer Height" GUI

"Jammer Radius" seen in Figure 5.9 gives the users a way to choose how big the area for the jamming should be. It is a slider that has values from 0km to 500km and is only visible when in "Dome Mode".

Figure 5.9: "Jammer Radius" GUI

"Jammer Position" seen in Figure 5.10 provides the users with an option to set the position of the jammer. The format for the coordinates is DDD, MMSS (Degree, Minutes Seconds) and values for latitude are between -90 and 90 and longitude between -180 and 180.

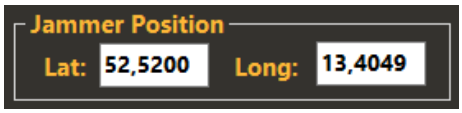


Figure 5.10: "Jammer Position" GUI

Interference "Update" seen in Figure 5.11 button gives the users the ability to update the interference configuration while a simulation is running. The values that can be updated are the "On-Off" toggle, "Jammer Radius" and "Jammer Position".

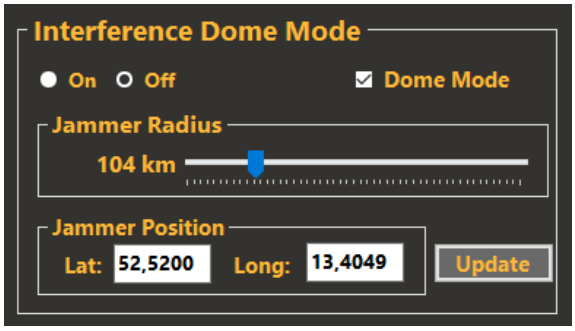


Figure 5.11: "Interference Configuration" GUI

5.1.3 Simulation

The "Simulation" part of the GUI consists of one checkbox, two buttons, a "Terminal" text box, and a position accuracy output.

"Terminal" gives the users feedback for easier use of the simulator. At the start, the color is green and gives instructions on how to start the simulation as can be seen in Figure 5.12. When starting the simulation, updating values, and when stopped the "Terminal" changes to a white color and types out what happened as seen in Figure 5.13. When there is an error with wrong values or double pressing the "Simulate" button the color changes to red and an error message is typed out as seen in Figure 5.14.

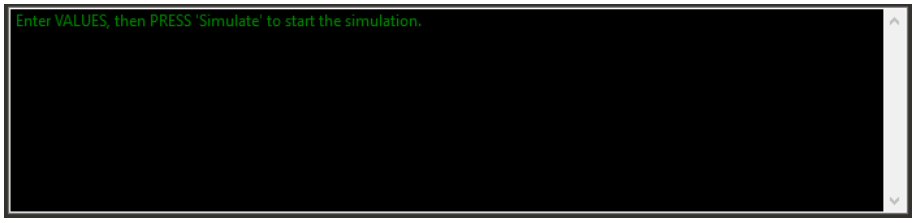


Figure 5.12: "Terminal" at startup

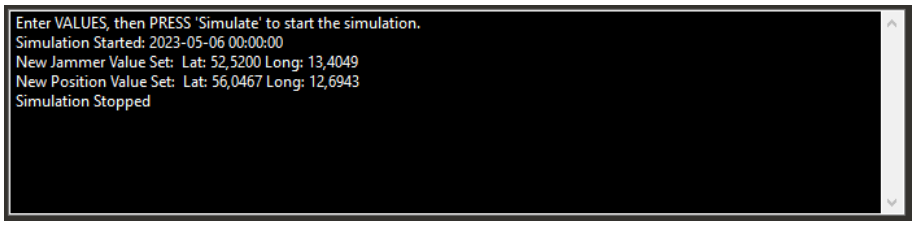


Figure 5.13: "Terminal" after a simulation has been conducted

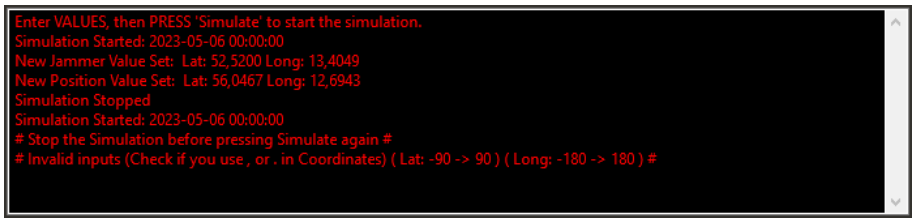


Figure 5.14: "Terminal" error

"Simulate" and "Stop/Clear" seen in Figure 5.15 buttons are used to start and stop the simulation. The "Stop/Clear" button changes depending on if the simulation is running or not. "Clear" removes all the text in the "Terminal".

Figure 5.15: "Simulate" & "Stop/Clear" GUI

"NMEA Output" seen in Figure 5.16 gives the users an option to make the simulation generate NMEA strings through the COM port.

Figure 5.16: "NMEA Output" GUI

"Position Accuracy" seen in Figure 5.17 provides the user with information on how accurate the current calculated position is, this calculation method can be seen in 4.10.

Figure 5.17: "Position Accuracy" GUI

5.2 Modular Programming

The current program is designed to transmit NMEA strings to the COM port, allowing VisualGPSView to interpret and visualize satellite positions and DOP values. However, the program employs modular programming, enabling users to integrate it with other simulations and transmit alternative data types such as position, time, or altitude.

5.3 Simulation

The simulation operates in several stages, it begins by reading the broadcast file, then it calculates the positions of all satellites based on the simulation time. Subsequently, it determines which satellites are visible from the receiver's current position and ensures that any potential jamming does not interfere with satellite connections. If the "NMEA Output" option is enabled, the simulation converts the current values into NMEA strings, which are then outputted via the COM port.

Chapter 6

Conclusion

The result of this thesis meets the initial purpose and requirement on all parts except for the implementation to include Glonass as an option in the simulator. The goal of the thesis as described in the first meeting at Saab can be seen in Appendix D.1. The goal of extracting a position, time fix, and quality indicators in terms of HDOP, VDOP, and PDOP are all met. The possibility of simulating interference to some extent is also achieved.

The final product of this thesis is a prototype with the potential for further development and integration into Saab's training simulator. This prototype gives the users the power to create GNSS scenarios by configuring both the receiver and jammer settings.

An interesting aspect to consider is the writer's very limited knowledge of orbits and "geodesy" before the start of this thesis. The inexperience in developing programs in C# is also a thing to consider. However, the result of this thesis shows how you're able to very quickly learn new stuff and that with the background from LTH, you have the tools to solve complex problems.

6.1 Process

The decision not to integrate the simulator with Saab's own simulator engine allows for a much more flexible work process. That decision made it possible for the software to be "stored" outside of Saab and it made it possible to also work on the software remotely. Even though the work mostly stayed at the office, it was now possible to catch up and make small experimental implementations when at home.

6.2 A Simulation Not an Emulation

This thesis was aimed towards the development of a simulator for training purposes, rather than an exact emulation of reality. As such, certain simplifications or adjustments were made in the program, making some implementations diverge from real-world scenarios and physics. After discussions with employees at Saab, the conclusion was that for training purposes there are different aspects to consider. An implementation exactly mirroring the real world might not be the best approach, there is no direct correlation between the realism of the simulator to the efficiency of it is training capabilities.

Instead of employing real-time antenna data to determine visible satellites, the simulation utilizes pre-existing broadcast files from different dates and times. These files are then used to calculate and simulate satellite positions. Additionally, the program can simulate HDOP, VDOP, and PDOP values based on the satellite geometry from different days. This approach allows for the simulation of training scenarios using approximate DOP and position accuracy, which is the primary focus of the training simulation.

6.3 Changes from Original Plan

Some changes naturally arose during the development of the simulator. While some parts of the initial scope had to be removed due to complexity, it was later possible to bring some parts back into the scope.

6.3.1 Deviation from Original Constellation Plan

In the original plan of the work for this thesis, the goal was to implement support for GPS, Galileo, and Glonass. However, as more knowledge was acquired, the difficulties of implementing Glonass was discovered. GPS and Galileo had a straightforward solution for calculating their positions in the ECEF coordinate system, and doing it at different times was simply a matter of changing a constant. For the Glonass system, it was a different solution. The calculation of positions at t_i from t_e (Time of Ephemeris) needed to be performed using several differential equations that described the satellite motion. The equations in question then need to be integrated using 4th-order Runge-Kutta technique. [30]

6.3.2 Addition of GUI

At the initial description of the thesis, the decision to exclude a GUI from the scope was determined. However, with the exclusion of implementation for Glonass a GUI was instead implemented. Also, during the development of the program, the need for a GUI grew with the amount of functionality that was added. Being able to control the entire simulation from a console application would not have been suitable for the amount of parameters that can be tweaked.

6.4 Possible Future Development

For this simulator, there is a bundle of potential future developments that could be interesting to investigate and explore. Both functional additions, but also changes and optimisations to allow for easier future development and maintainability.

6.4.1 Wider Support for Constellations

With more time it would be possible to add support for the other GNSS constellations. Since Glonass was not implemented that could be a start. Implementing Japan's QZSS would be relatively easy due to the constellation using the same fundamentals as GPS and Galileo for determining its position. Implementing "tracking" for Chinas Beidou and the IRNSS (Indian Regional Navigation Satellite System) would also be an interesting aspect to consider. With those implementations, you would have support for all the current GNSS constellations.

6.4.2 Multiple Jammers

One interesting future development to consider would be the possibility of adding multiple jammers at different positions. This idea was considered but with it needing significant changes it was determined not to be included in the program. However, with more time, this would make for an interesting aspect of the simulations, for example being able to block out two directions of view.

With an implementation of interference according to the one described in 4.12.5 where the range of the jammers is very limited, functionality for multiple jammers would be even more interesting. This could probably make for interesting training scenarios.

6.4.3 Jammer Simulation & Spoofing

Other ways of simulating jamming would be interesting to investigate. You could have multiple implementations of jamming for the possibility of simulating various scenarios. Though it would be difficult, investigating the possibility of implementing some way to simulate spoofing would make for an interesting simulator. This is also a part that could be implemented with almost infinite complexity.

The model for jamming-to-signal allowed for a simple implementation, with the assumption of that the jamming range always exceeds the line of sight distance. If an implementation were made with the actual calculations, it could allow for the simulation of jamming by plane. In this case, the potential jamming range would not always exceed the line of sight. Thus allowing for potentially interesting simulation scenarios.

6.4.4 SNR

With other models for implementing interference, and calculations for power received from the satellite being made, different approaches for the SNR values could also be investigated. By utilising the then calculated P_{rs} you could generate random noise instead of generating a "random" SNR, for a slightly more realistic approach.

6.4.5 Movement & Receivers

At the current stage of the simulator, the position is fully static and the only value chaining is the DOP-values and it is there the accuracy is being determined, based on the positions and geometry of the satellites. It would be interesting to add functionality for simulated movement. One way to develop this would be with controls in the GUI where it is possible to "drive" the vessel being simulated. Another approach would be to have the possibility to add a determined "travel route", like going from Visby to Helsingborg and being able to load that route into the program with a set speed. Or even with speed controls.

Together with this, an interesting enhancement would be to have multiple movable receivers within a simulated environment. Rather than solely simulating the position of a single position, there is the possibility of simulating multiple positions within a simulated world.

6.4.6 Wider Test-cases

At the current state of the program, there are four test cases for various calculations. For the robustness of the program, a wider range of test cases would be beneficial. Especially some tests for validating the position calculations of the satellites. It was not implemented due to its complexity and high amount of dependent variables. It would be a valuable enhancement due to the possibility of those calculations having a high probability of introducing bugs.

6.4.7 GUI

The GUI can be further developed in various aspects. Enhancing the usability with the use of user tests could be an interesting aspect to consider. For ease of use, a drag-and-drop for the Rinex file would make for a better experience. The same approach could apply to the potential of elevation data files.

6.4.8 Integration with Saab Simulator Engine

A final future development, that was kept out of the scope of this thesis was the integration with Saab's simulator engine. In the future, the program, or parts of it will hopefully be integrated with the engine providing a more realistic training ground for personnel.

6.5 Ethical Aspects

Doing work in the defense and military industry naturally raises the discussion of ethical aspects of the work. However, the defense and military sector is the origin of many advanced technologies that are being taken for granted in today's day and age [31]. In addition to things like internet, microwaves, and duct tape, one such example is precisely GPS and navigational systems. GPS was created by the United States of America in the 70s for military use, but has over the years been more applicable for civilian use and is today used by various industries and in countless domains.

With there being various branches of the military domain, one could also argue that the different branches have different ethical impacts. Doing technical research for a defense company, like navigational research, is widely different from researching weapons and explosives. Another aspect is where the line of responsibility is drawn, in what scenario is the user the one who is accountable and in what scenario can the accountability instead fall on the producer or manufacturer? Is the producer always inherently responsible or is the producer stripped of responsibility when the user and producer are different entities?

For this, there are many possible ethical aspects to consider, and it is eventually down to the individual to decide on what he or she thinks is right from their own perspective.

References

[1] "What are Global Navigation Satellite Systems?", *HEXAGON Novatel*, Accessed: April 11, 2024. [Online] Available: https://novatel.com/tech-talk/an-introduction-to-gnss/what-are-global-navigation-satellite-systems-gnss

[2] "WGS 84", *Lantmäteriet*, Accessed: April 11, 2024. [Online] Available: https://www.lantmateriet.se/en/geodata/gps-geodesi-och-swepos/reference-systems/three-dimensional-systems/wgs-84/

[3] Mike1024, Wikipedia, Accessed: April 11, 2024. [Online] Available: https://en.wikipedia.org/wiki/File:ECEF_ENU_Longitude_Latitude_relationships.svg

[4] G. Seeber, *Satellite Geodesy*, 2nd completely revised and extended edition, Germany, Walter de Gruyter GmbH & Co. KG, 2003. pp.211-403 [Ebook] Available: https://www.geokniga.org/bookfiles/geokniga-seeber-g-satellite-geodesy-2003.pdf

[5] "Broadcast ephemeris data", *CDDIS.nasa.gov*, Accessed: April 11, 2024. [Online] Available: https://cddis.nasa.gov/Data_and_Derived_Products/GNSS/broadcast_ephemeris_data.html

[6] "GNSS Ephemerides and Almanacs", *HEXAGON Novatel*, Accessed: April 11, 2024. [Online] Available: https://novatel.com/support/known-solutions/gnss-ephemerides-and-almanacs

[7] "GPS Navigation RINEX 4 (MGEX unofficial format)", 2021. [Online] Available:https://server.gage.upc.edu/gLAB/HTML/GPS_Navigation_Rinex_v4_CNAV.html

[8] *The Receiver Independent Exchange Format Version 4.00*, Rinex 4, 2023. [Online] Available: https://files.igs.org/pub/data/format/rinex_4.00.pdf

[9] "NMEA-0183 messages: Overview", *Trimble GNSS receivers help portal*, [Online] Accessed: April 8, 2024. [Online] Available: https://receiverhelp.trimble.com/alloy-gnss/en-us/NMEA-0183messages_MessageOverview.html

[10] *u-blox M9 SPG 4.04*, version 32.01, u-blox, Switzerland, 2023. pp.21-44 [Online] Available: https://content.u-blox.com/sites/default/files/u-blox-M9-SPG-4.04_InterfaceDescription_UBX-21022436.pdf

[11] A. Di Fonzo, M. Leonardi, G. Galati, P. Madonna and L. Sfarzo, "Software-Defined-Radio techniques against jammers for in car GNSS navigation," 2014 *IEEE Metrology for Aerospace (MetroAeroSpace)*, Benevento, Italy, 2014, pp. 320-325 [Online] Available: https://ieeexplore.ieee.org/abstract/document/6865942

[12] H. Pirayesh and H. Zeng, "Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey," in *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 767-809, Second quarter 2022. [Online] Available: https://ieeexplore.ieee.org/abstract/document/9733393?casa_token=Nfl4cQxImbYAAAAA

[13] Spirent, "What is PRN code?", *Spirent*, Oct. 1, 2024. [Online] Available: https://www.spirent.com/blogs/2011-10-17_what_is_prn_code

[14] D. Ford, "Live Map of Satellite Positions", *In-The-Sky.org*, Accessed: April 11, 2024. [Online] Available: https://in-the-sky.org/satmap_worldmap.php

[15] GNSS View, Version 5.0.2, NEC Corporation [App] Available: https://apps.apple.com/se/app/gnss-view/id924350018

[16] "Transformation mellan X, Y, Z och φ, λ, h", Lantmäteriet, Oct. 11 2012. Accessed: April 11, 2024. [Online] Available: https://www.lantmateriet.se/globalassets/geodata/gps-och-geodetisk-matning/xyz_geodetiska_koord_och_exempel.pdf

[17] "GNSS Ephemerides and Almanacs," *Novatel*, Accessed: April 27, 2024. [Online] Available: https://novatel.com/support/known-solutions/gnss-ephemerides-and-almanacs

[18] O.K Isik, J.Hong, I.Petrunin, and A.Tsourdos, "Integrity Analysis for GPS-based navigation of uavs in urban environment," *Robotics*, vol. 9, no. 3, Aug. 2020. Accessed: April. 12, 2024. [Online] Available: https://www.mdpi.com/2218-6581/9/3/66

[19] G. E. Box and M. E. Muller, “A note on the generation of random normal deviates,” *The Annals of Mathematical Statistics*, vol. 29, no. 2, pp. 610–611, Jun. 1958. [Online] Available: https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-29/issue-2/A-Note-on-the-Generation-of-Random-Normal-Deviates/10.1214/aoms/1177706645.full

[20] L. Faria, C.A.M Silvestre, M.A.F Correia, N.A Roso, "GPS Jamming Signals Propagation in Free-Space, Urban and Suburban Environments," Journal of Aerospace Technology and Management, vol.10, Feb.2018. [Online] Available: https://www.researchgate.net/publication/323405032_GPS_Jamming_Signals_Propagation_in_Free-Space_Urban_and_Suburban_Environments

[21] V.J.Ostdiek, D.J.Bord, *Inquiry into physics*, Third Edition, Minneapolis/St. Paul : West Pub. Co., 1995, ch. 8, pp. 315-316.

[22] H. T. Friis, "A Note on a Simple Transmission Formula," in Proceedings of the IRE, vol. 34, no. 5, pp. 254-256, May 1946.

[23] "Geotorget", *Lantmäteriet*, Accessed: April 12, 2024. [Online] Available: https://geotorget.lantmateriet.se/geodataprodukter/markhojdmodell-nedladdning-grid-50

[24] GeoTiffElevationReader, Version 1.0.1, [Software], msg_kurt, Accessed: April. 12, 2024. [Online] Available: https://www.nuget.org/packages/GeoTiffElevationReader/1.0.1?_src=templat

[25] MightyLittleGeodesy, Version 1.0.2, [Software], bjorn-sallarp, Accessed: April. 12, 2024. [Online] Available: https://www.nuget.org/packages/MightyLittleGeodesy/1.0.2?_src=template

[26] "Earthdata Search", *search.earthdata.nasa.gov*, Accessed: April. 12, 2024. [Online] Available: https://search.earthdata.nasa.gov/search/granules?p=C1711961296-LPCLOUD&pg[0][v]=f&pg[0][gsk]=-start_date&fi=ASTER&tl=1712687657!3!!

[27] VisualGPSView, [Software], Anaheim California, Monte Variakojis, Accessed: April. 12, 2024. [Online] Available: https://www.visualgps.net/#visualgpsview-content

[28] Free Virtual Serial Ports, Version 5.27.00.1156, [Software], HHD Software, 2023, Accessed: April. 12, 2024. [Online] Available: https://freevirtualserialports.com

[29] "Colours", *Saab*, [Online] Accessed: April 17, 2024. [Online] Available: https://brand.saab.com/point/en/saab/digital-colours

[30] "GLONASS Satellite Coordinates Computation," *EESA Navpedia*, Accessed: April. 11, 2024. [Online] Available: https://gssc.esa.int/navipedia/index.php?title=GLONASS_Satellite_Coordinates_Computation

[31] "Science and innovation," *North Atlantic Treaty Organization*, Accessed: April. 16, 2024. [Online] https://www.nato.int/cps/en/natohq/declassified_215371.htm

Appendices

Appendix A

Trello

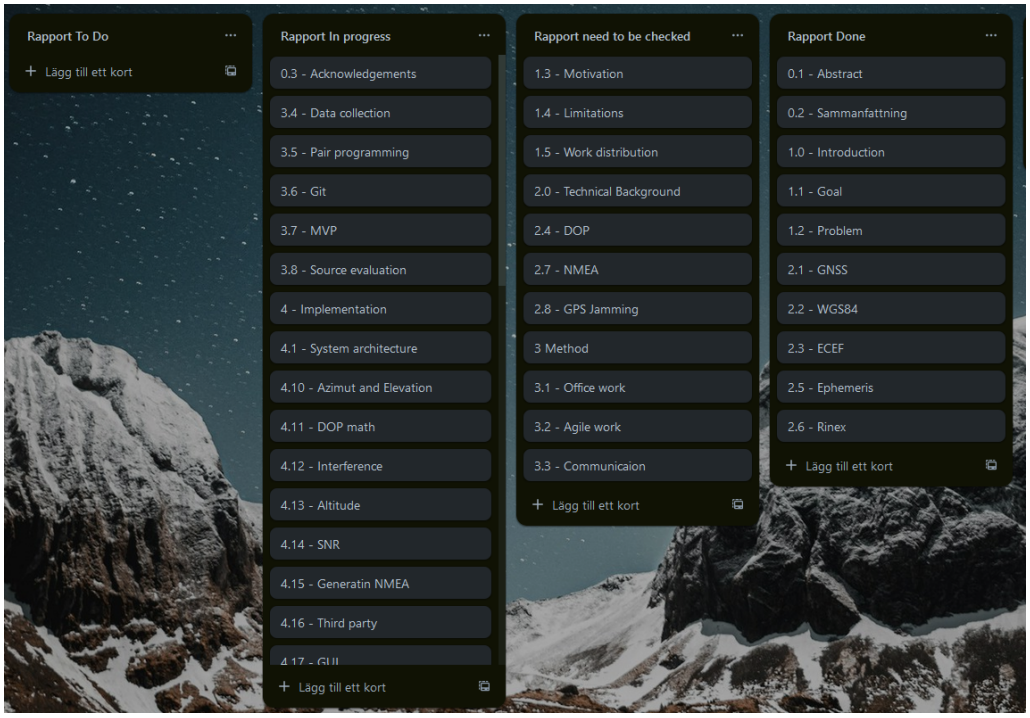


Figure A.1: Trello Screenshot

Appendix B

Models

```
public class GNSS
{

    public Galileo Galileo {get;set;}
    public GPS Gps { get; set; }
    public Glonass Glonass { get; set; }
}
```

Figure B.1: GNSS Model

```
public class GPS
{
    public List<GPSSatellite> satList { get; set; }
}
```

Figure B.2: GPS Model

```
public class GPSSatellite
{
    public List<BroadCastDataLNAV> Data { get; set; }
    public string Id { get; set; }

}
```

Figure B.3: GPSSatellite Model

```
public class BroadCastDataLNAV
{

    public string SatId { get; set; }
    public DateTime DateTime { get; set; }
    public double ClockBias { get; set; } //Not Used
    public double ClockDrift { get; set; } //Not Used
    public double ClockDriftRate { get; set; } //Not Used

    //Orbit1
    public double IssueOfDataEphemeris { get; set; }
    public double RadiusCorrectionSinusComponent { get; set; }
    public double Delta_n0 { get; set; }
    public double M0_Angle { get; set; }

    //Orbit2
    public double LatitudeCorrectionCosinusComponent { get; set; }
    public double OrbitEcentricity { get; set; }
    public double LatitudeCorrectionSinusComponent { get; set; }
    public double SqrtOfMajorAxis { get; set; }

    //Orbit3
    public double TimeOfEphemeris { get; set; }
    public double InclinationCorrectionCosinusComponent { get; set; }
    public double OmegaAngle0 { get; set; }
    public double AngularVelocity { get; set; }

    //Orbit4
    public double InitialInclination { get; set; }
    public double RadiusCorrectionCosinusComponent { get; set; }
    public double OmegaAngle { get; set; }
    public double AngularVelocityPerSec { get; set; }

    //Orbit5
    public double InclinationRate { get; set; }
    public double CodesOnL2Channel { get; set; } //Not Used
    public double ContinousGpsWeek { get; set; } //Not Used
    public double L2PDataFlag { get; set; } //Not Used

    //Orbit6
    public double SpaceVehicleAccuarcy { get; set; } //Not Used
    public double SpaceVehicleHealth { get; set; } //Not Used
    public double TotalGroupDelay { get; set; } //Not Used
    public double IssueOfDataClock { get; set; }

    //orbit7
    public double TransmissionTime { get; set; }
```

Figure B.4: BroadCastDataLNAV Model

Appendix C

NMEA

```
Receive
$GPGSV,4,1,13,05,34,215,26,07,62,79,27,08,2,74,15,09,27,102,24*42
$GPGSV,4,2,13,11,15,228,24,13,28,272,26,14,14,161,28,15,1,284,13*49
$GPGSV,4,3,13,16,7,23,15,18,7,345,15,20,59,242,25,27,8,44,16*46
$GPGSV,4,4,13,30,70,183,26*41
$GLGSA,A,3,07,10,11,30,33,,,,,,,1.1,0.6,0.9*24
$GLGSV,2,1,5,71,36,106,29,74,23,317,32,75,13,343,27,94,40,67,27*66
$GLGSV,2,2,5,97,13,237,25*6D
$GNGGA,120038.00,562.80200000000000778000,N,01241.65800000000001000,E,1,18,0.6,3.636,M,-15,M,0,,*61
$GPGSA,A,3,05,07,08,09,11,13,14,15,16,18,20,27,1.1,0.6,0.9*34
$GPGSV,4,1,13,05,34,215,29,07,62,78,27,08,2,74,13,09,27,102,29*47
$GPGSV,4,2,13,11,15,228,22,13,28,272,29,14,14,161,30,15,1,284,14*4E
$GPGSV,4,3,13,16,7,23,13,18,7,345,13,20,59,242,25,27,8,44,14*44
$GPGSV,4,4,13,30,70,183,23*44
$GLGSA,A,3,07,10,11,30,33,,,,,,,1.1,0.6,0.9*24
$GLGSV,2,1,5,71,36,106,28,74,23,317,26,75,13,343,24,94,40,67,32*65
$GLGSV,2,2,5,97,13,237,29*61
$GNGGA,120039.00,562.80200000000000778000,N,01241.65800000000001000,E,1,18,0.6,3.636,M,-15,M,0,,*60
$GPGSA,A,3,05,07,08,09,11,13,14,15,16,18,20,27,1.1,0.6,0.9*34
$GPGSV,4,1,13,05,34,215,29,07,62,78,28,08,2,74,14,09,27,102,25*43
$GPGSV,4,2,13,11,15,228,29,13,28,272,28,14,14,161,29,15,1,284,14*4C
$GPGSV,4,3,13,16,7,23,12,18,7,345,13,20,59,242,29,27,8,44,13*4E
$GPGSV,4,4,13,30,70,183,27*40
$GLGSA,A,3,07,10,11,30,33,,,,,,,1.1,0.6,0.9*24
$GLGSV,2,1,5,71,36,106,28,74,23,317,25,75,13,342,29,94,40,67,29*60
$GLGSV,2,2,5,97,13,237,25*6D
$GNGGA,120040.00,562.80200000000000778000,N,01241.65800000000001000,E,1,18,0.6,3.636,M,-15,M,0,,*6E
$GPGSA,A,3,05,07,08,09,11,13,14,15,16,18,20,27,1.1,0.6,0.9*34
$GPGSV,4,1,13,05,34,215,30,07,62,78,29,08,2,74,15,09,27,102,24*4A
$GPGSV,4,2,13,11,15,228,26,13,28,272,23,14,14,161,31,15,1,284,15*40
$GPGSV,4,3,13,16,7,23,14,18,7,345,13,20,59,242,25,27,8,44,14*43
$GPGSV,4,4,13,30,70,183,25*42
$GLGSA,A,3,07,10,11,30,33,,,,,,,1.1,0.6,0.9*24
$GLGSV,2,1,5,71,36,106,25,74,23,317,22,75,13,342,23,94,40,67,30*68
$GLGSV,2,2,5,97,13,237,26*6E
$GNGGA,120041.00,562.80200000000000778000,N,01241.65800000000001000,E,1,18,0.6,3.636,M,-15,M,0,,*6F
$GPGSA,A,3,05,07,08,09,11,13,14,15,16,18,20,27,1.1,0.6,0.9*34
$GPGSV,4,1,13,05,34,215,29,07,62,78,31,08,2,74,13,09,27,102,28*41
$GPGSV,4,2,13,11,15,228,28,13,28,272,30,14,14,161,28,15,1,284,14*45
$GPGSV,4,3,13,16,7,23,13,18,7,345,13,20,59,242,26,27,8,44,13*40
$GPGSV,4,4,13,30,70,183,24*43
$GLGSA,A,3,07,10,11,30,33,,,,,,,1.1,0.6,0.9*24
$GLGSV,2,1,5,71,36,106,28,74,23,317,26,75,13,342,30,94,40,67,26*64
$GLGSV,2,2,5,97,13,237,27*6F
$GNGGA,120042.00,562.80200000000000778000,N,01241.65800000000001000,E,1,18,0.6,3.636,M,-15,M,0,,*6C
$GPGSA,A,3,05,07,08,09,11,13,14,15,16,18,20,27,1.1,0.6,0.9*34
$GPGSV,4,1,13,05,34,215,27,07,62,78,28,08,2,74,14,09,27,102,27*4F
$GPGSV,4,2,13,11,15,228,29,13,28,272,25,14,14,161,28,15,1,284,14*40
$GPGSV,4,3,13,16,7,23,15,18,7,345,14,20,59,242,24,27,8,44,13*43
$GPGSV,4,4,13,30,71,183,30*47
$GLGSA,A,3,07,10,11,30,33,,,,,,,1.1,0.6,0.9*24
$GLGSV,2,1,5,71,36,106,30,74,23,317,27,75,13,342,23,94,40,67,28*60
$GLGSV,2,2,5,97,13,237,25*6D
$GNGGA,120043.00,562.80200000000000778000,N,01241.65800000000001000,E,1,18,0.6,3.636,M,-15,M,0,,*6D
$GPGSA,A,3,05,07,08,09,11,13,14,15,16,18,20,27,1.1,0.6,0.9*34
$GPGSV,4,1,13,05,34,215,26,07,62,78,36,08,2,74,12,09,27,102,26*46
$GPGSV,4,2,13,11,15,228,27,13,28,272,27,14,14,161,26,15,1,284,13*45
$GPGSV,4,3,13,16,7,23,13,18,7,345,15,20,59,242,27,27,8,44,12*46
$GPGSV,4,4,13,30,71,183,30*47
$GLGSA,A,3,07,10,11,30,33,,,,,,,1.1,0.6,0.9*24
$GLGSV,2,1,5,71,36,106,25,74,23,317,26,75,13,342,22,94,40,67,29*65
$GLGSV,2,2,5,97,13,237,20*68
```

Figure C.1: NMEA captured in VisualGPSView

Appendix D

Goal product

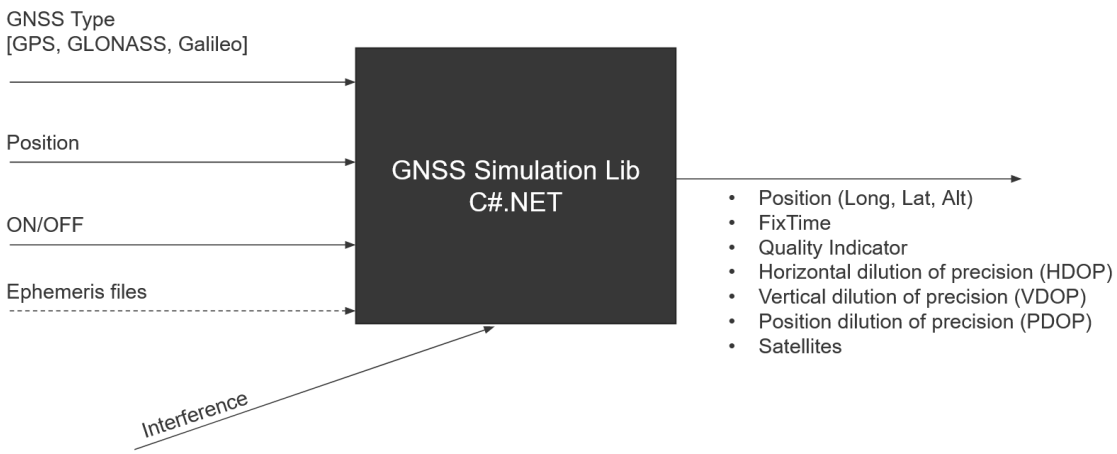


Figure D.1: Goal of thesis as shown at first meeting

Students Julius Andersson, Simon Antonsson
Supervisor Anders J Johansson (LTH)
Examiner Christin Lindholm (LTH)

Faculty of Engineering | Lunds Tekniska Högskola | Presented 2024-05-30

GNSS Training Simulator

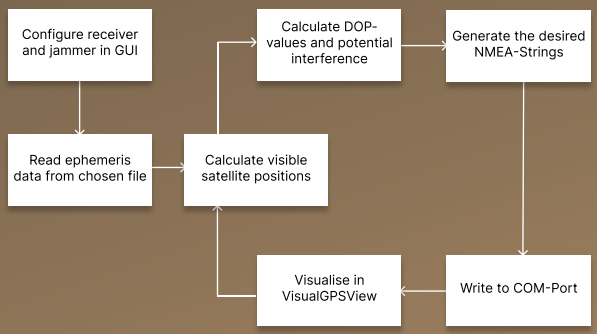
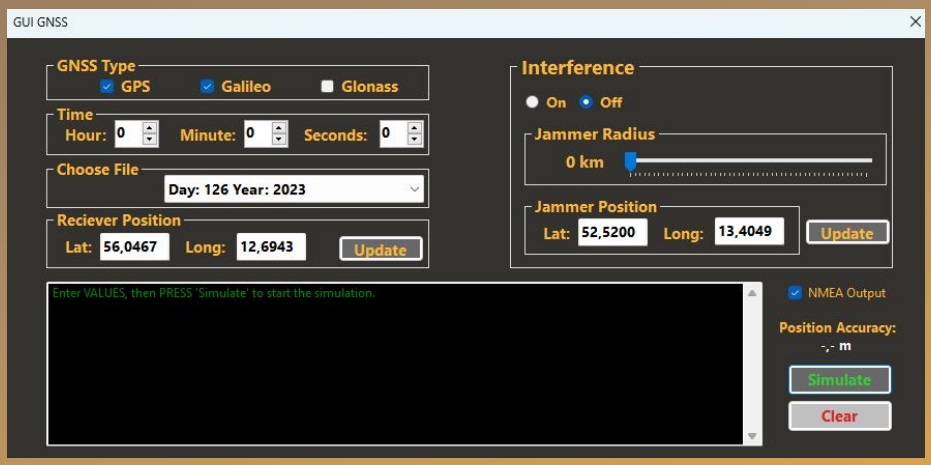
The current training simulator at Saab uses static values during the simulation. However, for a training simulator to be effective, it needs to simulate a scenario as close to reality as possible. Therefore, this thesis explored the possibility of creating a training simulator prototype, which uses old satellite positions to calculate the values. Additionally, various approaches modeling the interference were researched and in the end, two models were implemented for simulating interference, which in the current training simulator was not possible.

Persistent values throughout a session in the current simulator, coupled with the inability to simulate interference, diminish its effectiveness in creating realistic settings and scenarios. Consequently, the training simulator falls short of adequately preparing naval personnel for real-world scenarios and therefore has to be improved.

The approach for the development of the program was divided into two phases. The first phase focused on the development of a "Minimal Viable Product" that did not take any inputs and only used the terminal to show the values of the calculations. The second phase included the development of a program with well-structured code, a model for interference, and an implementation of a GUI for gathering inputs from the user.

To solve the problem of having persistent values, satellite broadcast files downloaded from NASA has been utilised in order to create a realistic setting of the satellite constellation. With the real position of satellites at a set time, it is possible to calculate the Dilution of Precision values accordingly. Additionally, to solve the absence of interference simulations, various models and approaches were explored, with two being ultimately implemented. One focuses on a more realistic approach according to physics, looking at things such as radio wave propagation and line of sight. With this model, you are either fully jammed, or not jammed at all. The second model has a bigger focus on the effectiveness of training, where configuring the jamming radius and position gives the user the flexibility to go from minimal interference all the way to fully jamming the receiver.

The result of this thesis is a locally run program that functions as a prototype GNSS training simulator, with support for the GPS and Galileo constellations. The program was successfully implemented with functionality for simulating interference as well as generating the output DOP-values, which are calculated based on the satellite geometry.



In conclusion, the work meets the goals and requirements that were set at the start of the thesis. During the work, a discussion arose about training purposes and how a more realistic model of interference does not directly convert to a better training simulator. Although the goals were met, various possible future developments could be interesting to explore. Such examples could be integrating other GNSS constellations, adding multiple jammers, or enhancing the GUI. In the end, this prototype will hopefully contribute to the further development of Saab's training simulator.