TALLINN UNIVERSITY OF TECHNOLOGY
SCHOOL OF ENGINEERING
Department of Electrical Power Engineering and Mechatronics

INDIRECT TRACKING OF OBJECTS WITH GNSS

OBJEKTIDE KAUDNE JÄLGIMINE GNSS-IGA

MASTER THESIS

Student:	Swaminathan Ramachandran
Student code:	195433 MAHM
Supervisor:	Mihkel Tommingas, Industrial PhD Student
Student Code:	215064 IAQD
Co-Supervisor:	Dr. Heigo Mõlder, Researcher

Tallinn 2021

(*On the reverse side of title page*)

AUTHOR'S DECLARATION

Hereby I declare, that I have written this thesis independently.
No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

"21" December 2021

Author: ....
/signature /

Thesis is in accordance with terms and requirements

"......." .... 20….

Supervisor: …........................…..
/signature/

Accepted for defence

".......".....................20… .

Chairman of theses defence commission: ....….
/name and signature/

Non-exclusive Licence for Publication and Reproduction of GraduationTthesis[1]

I, Swaminathan Ramachandran (date of birth: 09/04/1997) hereby

1. grant Tallinn University of Technology (TalTech) a non-exclusive license for my thesis "Indirect tracking of objects with GNSS",

supervised by

Industrial PhD Student, Mihkel Tommingas, 215064IAQD

1.1 reproduced for the purposes of preservation and electronic publication, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright;

1.2 published via the web of TalTech, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright.

1.3 I am aware that the author also retains the rights specified in clause 1 of this license.

2. I confirm that granting the non-exclusive license does not infringe third persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

[1] *Non-exclusive Licence for Publication and Reproduction of Graduation Thesis is not valid during the validity period of restriction on access, except the university`s right to reproduce the thesis only for preservation purposes.*

______________ (*signature*)

21/12/2021 (*date*)

Department of Electrical Power Engineering and Mechatronics

THESIS TASK

Student: Swaminathan Ramachandran, 195433 MAHM

Study programme, MAHM02/18 - Mechatronics

main speciality: Mechatronics

Supervisor(s): Mihkel Tommingas, Industrial PhD Student, 215064IAQD

Co-Supervisor: Dr. Heigo Mõlder, Researcher

Thesis topic:

(in English) *Indirect tracking of objects with GNSS*

(in Estonian) *Objektide kaudne jälgimine GNSS-iga*

Thesis main objectives:

1. To evaluate the performance of the Fieldbee GNSS receiver in different scenarios.
2. Propose a concept of indirect tracking of objects by repeatability tests.
3. To track different objects by evaluating the repeatability of positioning.

Thesis tasks and time schedule:

No	Task description	Deadline
1.	Information search and review of previous work	January 2021
2.	Selection of components and performing experiments	July 2021
3.	Compiling and analyzing the experimental data	September 2021
4.	Results and conclusions	October 2021
5.	Thesis defense	January 2022

Language: English **Deadline for submission of thesis:** "21th December 2021"

Student: Swaminathan Ramachandran ....

/signature/

Supervisor: Mihkel Tommingas ....

/signature/

Co-Supervisor: Heigo Mõlder ....

/signature/

Head of study programme: Anton Rassõlkin ....

/signature/

Terms of thesis closed defence and/or restricted access conditions to be formulated on the reverse side

TABLE OF CONTENTS

LIST OF FIGURES 7

LIST OF TABLES 8

PREFACE 9

LIST OF ABBREVIATIONS 10

1. INTRODUCTION 11

  1.1 Problem Statement 12

2. LITERATURE REVIEW 13

  2.1 GNSS Systems 13

  2.2 Errors in GNSS systems 14

  2.3 Precise Point Positioning (PPP) 15

  2.4 Real Time-Kinematic Positioning (RTK) 16

  2.5 Satellite Based Augmentation System (SBAS) 19

  2.6 Other Services and Techniques implemented 21

  2.7 Summary of Literature 22

  2.8 Objective of the Thesis 24

3. METHODOLOGY 25

  3.1 Hardware Part 25

3.1.1 Fieldbee GNSS L2 receiver 25

3.1.2 Fieldbee GNSS L2 Base Station 27

  3.2 Software Part 28

3.2.1 Python and R 28

3.2.2 Collection of Data 28

3.2.3 True Coordinates and Receiver Coordinates 29

3.2.4 Calculation of Data 29

  3.3 Performance Metrics 30

3.3.1 Accuracy 30

3.3.2 Standard Deviation 30

3.3.3 Root-Mean-Squared Error (RMSE) 31

3.3.4 Distance (Haversine Formula) 31

3.3.5 Distribution Plots 32

4. TEST SETUP 34

  4.1 Static Testing Setup 34

  4.2 Dynamic Testing Setup 35

  4.3 Static Tests at Nordwood Sawmill 36

  4.4 Dynamic Tests at Nordwood Sawmill 36

  4.5 Concept of Indirect Tracking 37

5. RESULTS AND DISCUSSIONS 38
5.1 Open-Space Scenario 42
5.2 Semi-Obstructed Scenario 45
5.3 Obstructed Scenario 47
5.4 Dynamic Test Results 50
5.5 Nordwood Sawmill at Aegviidu 51
5.6 Indirect Tracking: 54
5.7 Summary of Results: 57
6. SUMMARY 58
6. KOKKUVÕTE 60
REFERENCES 62
7. APPENDICES 65

LIST OF FIGURES

Figure 2.1: GNSS RTK Setup [16] 19

Figure 3.1: Fieldbee L2 Receiver........ 26

Figure 3.2: Distribution plots of Latitude and Longitude 33

Figure 4.1: Receiver at a) Open Space b) at Obstructed space........ 35

Figure 4.2: Fieldbee GNSS L2 receiver placed on a Forklift 36

Figure 5.1: Map of X-GIS 38

Figure 5.2: Data points with True Coordinates 39

Figure 5.3: Standard Deviation values under different scenarios 39

Figure 5.4: Standard Deviation values of Latitudes with and without Base station 40

Figure 5.5: Standard deviation values of Longitudes with and without Base station .. 40

Figure 5.6: Data points with and without Base station (Semi-Obstructed Scenario) .. 41

Figure 5.7: Distribution plot with Base Station (Open Space)........ 43

Figure 5.8: Distribution Plot without Base station (Open Space) 43

Figure 5.9: Data points collected (Open Space)........ 44

Figure 5.10: Standard Deviation of Latitudes (Semi-Obstructed)........ 46

Figure 5.11: Standard Deviation of Longitudes (Semi-Obstructed)........ 46

Figure 5.12: Data points collected (Semi-Obstructed) 47

Figure 5.13: Standard Deviation of Latitudes (Obstructed)........ 48

Figure 5.14: Standard Deviation of Longitudes (Obstructed)........ 49

Figure 5.15: Data Points collected (Obstructed) 49

Figure 5.16: Dynamic movement of receiver a) on R-Plot b) on X-GIS Map 50

Figure 5.17: Layout of Nordwood sawmill territory........ 51

Figure 5.18: Standard Deviation of Latitudes at the sawmill 52

Figure 5.19: Standard Deviation of Longitudes at the sawmill 52

Figure 5.20: Data points collected at Aegviidu (Obstructed) 53

Figure 5.21: Data points collected at Aegviidu (Open Space) 54

Figure 5.22: Repeatbility results (Open Space) 55

Figure 5.23: Repeatability results (Semi-Obstructed)........ 56

LIST OF TABLES

Table 3.1: Cost comparison of different receivers 25
Table 3.2: NMEA Format 27
Table 5.1: Results of Open Space 42
Table 5.2: Results of Semi-Obstructed.. 45
Table 5.3: Results of Obstructed Scenario 48

PREFACE

The motivation of the thesis is to propose a concept for indirect tracking of different objects. The thesis topic was proposed by Eliko Tehnoloogia Arenduskeskus OÜ. The measurements were done at different places in Tallinn. Additional measurements were taken at the Nordwood Sawmill in Aegviidu, where this concept can be put to use.

I owe my heartiest gratitude to Sander Ulp for his guidance and the invaluable knowledge I attained during the course of the research. I would also like to express my earnest thanks to Mihkel Tommingas who has helped me learn through out the period of research, being patient to clarify my doubts, giving his valuable feedback and maintaining a friendly atmosphere. I thank them for motivating me to pursue the project to completion regardless of the setbacks faced without whom, this work would not have been successful. I would like to thank everyone at Eliko for being friendly and sharing their knowledge. This opportunity also exposed me to the various cultural, professional and social stances in Estonia.

I extend my heartfelt thanks to Researcher Dr. Heigo Mõlder for accepting to be the co-supervisor for my research and giving valuable comments and suggestions.

I also take this opportunity to thank Tallinn University of Technology for granting me this prestigious opportunity to undertake my Masters studies in Mechatronics at the university.

Last but not the least, I express my gratitude to my parents, friends and relatives for their blessings and everlasting support.

Keywords: GNSS, RTK, Indirect Tracking

LIST OF ABBREVIATIONS

GNSS -Global Navigation Satellite System

GPS – Global Positioning System

GLONASS – GLObal NAvigation Satellite System

BDS – BeiDou navigation satellite System

NMEA – National Marine Electronics Association

SIS – Signals In Space

RTK – Real Time Kinematic

PPP – Precise Point Positioning

SF-PPP – Single Frequency Precise Point Positioning

IGS – International GNSS Service

SBAS – Satellite Based Augmentation Sytem

EGNOS – European Geostationary Navigation Overlay Service

CODE – Centre of Orbit Determination in Europe

SPP – Single Point Positioning

SISRE – Signal In Space Range Error

AR – Ambiguity Resolution

ESA - European Space Agency

X-GIS – Estonian Geographic Information System

RMSE – Root Mean Square Error

OTF-AR – On The Fly-Ambiguity Resolution

ISO – International Organization of Standardization

ANN – Artificial Neural Network

SHM – Structural Health Monitoring

1. INTRODUCTION

Global Navigation Satellite System (GNSS) is a satellite navigation system that uses satellite signals to determine the geographic location of a user's receiver on Earth. The satellites provide signals from space that transmit positioning and timing data (position, velocity, time) to the GNSS receivers. The receivers make use of this data to determine the location. Currently there are four GNSS systems used in the world.

1. Global Positioning System (GPS), United States
2. GLObal NAvigation Satellite System (GLONASS), Russia
3. Galileo, Europe
4. BeiDou (BDS), China

The initial development of the first GNSS systems was in the field of military. But with the free availability of GNSS signal and easy access to cheap GNSS receivers, the GNSS technology is having an extensive use in civil, industrial, and scientific areas as well. GNSS is known to provide worldwide services in positioning and navigation. The performance of these services is directly related to the quality of satellite signals.

A GNSS receiver is an electrical device that receives and digitally measures the signals from a navigation satellite constellation to estimate its position, velocity, and time. They are the central item for satellite positioning. The satellites send signals over two carrier frequencies in the L-Band referred to as L1 and L2. The data from these signals is sent to the receivers on Earth. The utilization of phase measurements of carrier signals can achieve centimeter-level positioning accuracy in ground-based positioning systems [1].

The GNSS receivers (also referred to as rovers) comprise of two sections. Specifically, the antenna and the processing unit. The antenna is utilized for receiving the satellite signal, while the processing unit measures the data from these signals. The received data contains the information, which is utilized to register the exact position of the receiver with regards to the satellite. It then makes an interpretation of this information into an Earth-based coordinate system giving an outcome in terms of latitude, longitude, and height.

The receivers usually output coordinates as NMEA (National Marine Electronics Association) messages. This data is based on the ASCII characters, which communicates at a rate of 38400 bits per second, which is the same as the baud rate of 38400 characters per second. The GNSS data that is transmitted and received from the receiver is in a standard NMEA-0183 format. These NMEA sentences are represented in different message types such as $GPGGA, $GPGGL, $GPVTG etc. [2]. These messages usually contain the information regarding latitudes, longitudes, UTC time, number of satellites in view etc.

1.1 Problem Statement

In the usual approach, the objects that need to be tracked are equipped with a GNSS receiver. For example, in an outdoor warehouse if there is a need to track the cargo then each package should have a GNSS receiver. This approach would quickly scale to be expensive if the number of objects to be tracked is large. Furthermore, the receivers with higher accuracy also increase the cost of the device. In the early stages of development of receivers around 1982, the cost of the first commercial receivers was up to 250,000€. Nowadays a high-end geodetic GNSS receiver would cost around 15,000€. If the user considers a single-frequency geodetic receiver, he or she still needs to pay around 1000-1500€ for a single receiver. This may pose a problem for companies requiring GNSS receivers for various tasks such as surveying, tracking cargo, monitoring, and collecting data from many locations [3]. Also, some GNSS receivers lack the positioning accuracy of the object to be tracked.

This thesis is based on quantitative research where experiments are made with a GNSS receiver and a base station using the Real Time Kinematic (RTK) technique. The research topic was given by Eliko Tehnoloogia Arenduskeskus OÜ who specialize in positioning. Eliko had chosen the Fieldbee RTK L2 GNSS system as it is one of the inexpensive options available in the market that offers RTK accuracy with the receiver and a base station included. The price of the devices economically conforms to the budget calculated by Eliko internally for this use case. The devices are quoted to achieve an accuracy of 50 cm requirement which was set by Eliko for the use case. The purpose of this thesis is to evaluate the performance of the system under practical conditions. The measurements were taken under different scenarios such as open space area, semi-obstructed area, and obstructed area in Tallinn and Nordwood Sawmill at Aegviidu. The sawmill was chosen to track the cargo indirectly by placing the receivers on the forklifts that carry the objects. This would reduce the number of trackers required. But for this method, the receiver needs to be robust and determine the position accurately. The measured data is compiled to evaluate the accuracy by calculating different parameters such as accuracy, precision, distribution of points etc. The results would determine the suitability of this system to solve the given use case.

2. LITERATURE REVIEW

This chapter describes the introduction, existing solutions, problems, and research done on the tracking, navigation applications using the different GNSS services.

2.1 GNSS Systems

A Global Navigation Satellite System comprises of three significant components also called as segments [4]. These are:

(i) Space Segment – This consists of the GNSS satellites which orbit above the Earth at a distance of about 20,000 km. Every system (GPS, GLONASS, etc) consists of its own satellite constellations. These satellites send the signals to the receivers on Earth.

(ii) Control Segment – Control segments are a group of ground-based master control stations, data uploading stations and monitor stations. To maintain the accuracy, the master control station modifies the satellite's orbit parameters and onboard high-precision clocks as needed. The signals and status of the satellites are monitored via the monitor stations. The signals are analyzed at these stations which then sends orbit and time corrections to the satellites via data uploading stations.

(iii) User Segment – These consists of the equipment that process the satellite signals and uses them to provide positioning information on Earth. The equipment includes smartphones, GNSS receivers which can be used for survey and mapping applications.

GNSS Receivers:

The GNSS receivers are the major components that are responsible for processing the L – Band Signals in Space (SIS) received from the satellites. These receivers scan for the signals transmitted by the satellites and synchronize with them. They then extract the information from the received data to calculate the user's position, velocity, and precise time.

The main components of a GNSS receiver include:

(i) An Antenna, which is used to receive the GNSS signals, the noise signals, and possible interferences.

(ii) A Front End which is used for converting, filtering, amplifying, and digitizing the signals received.

(iii) Baseband Signal Processing which uses routines of several signal processing to receive and track different signals.

(iv) Applications Processing which performs different tasks based on the required application.

The leading companies in manufacturing GNSS receivers are Trimble, Leica, Fieldbee etc.

2.2 Errors in GNSS systems

There are different factors which affect the accuracy of GNSS positioning. These include the physical location of the system, the atmospheric conditions, satellite positions and their line-of-sight etc. These may block or degrade the satellite signals from reaching the receiver which results in inaccurate or no readings.

In conditions where there are no obstructions such as an open space, the accuracy of standard GNSS receivers are typically around two meters. In most cases these receivers rely on the time it takes for the satellite signal to reach them. Errors in time nanosecond can have a negative impact on the accuracy.

The loss of accuracy due to errors in the orbital satellite position was found to be around 2.5 meters [5]. The satellite clock errors can add about 1.5 meters of loss in accuracy. Inaccuracies can also be caused by the varying of troposphere and ionosphere conditions. The GNSS receiver enhances its precision and accuracy by using the GNSS correction data to eliminate the errors. This is achieved by monitoring the GNSS signals from a base station in a known location. The deviations that occur in position of the base station is monitored and sent to the rover.

To achieve high accuracy positioning using GNSS, two main techniques namely, Real Time Kinematic (RTK) and Precise Point Positioning (PPP) have been developed.

2.3 Precise Point Positioning (PPP)

PPP is a popular GPS service giving high precision without the use of additional GNSS infrastructure. Single Frequency PPP (SF-PPP) acquires high positioning accuracy even with the use of low-cost GNSS receivers. This technology was used in lane identification and mapping on a motorway using GPS with successful results. But the problem lies in the performance which depends strongly in open areas. Peter F.de Bakker and Christian C.J.M Tiberius extended this technology from GPS to other GNSS services such as the GLONASS. Several experiments with the GNSS receivers were performed to demonstrate the performance of SF-PPP with multi GNSS systems. The results indicated that the implementation of SF-PPP with multi GNSS systems showed improved performance when compared to GPS only in the case of closed or obstructed areas [6].

The PPP method was first proposed by Zumberge et al. in 1997 with GPS service [7]. PPP is a technique that is used to determine the position of a receiver without having any communication with the reference station. PPP highly depends on the availability of satellite orbit, satellite clock and also on ionospheric products which help in limiting the convergence time. It has been in development for around 15 years. PPP is able to acquire high precision as it employs the use of precise products that are made available by the International GNSS Service (IGS). This method reduces the use of costly regional network of reference stations which makes it possible to obtain measurements in areas which have poorly developed ground structure. In PPP, a number of additional data such as the errors of atmospheric model, errors of satellite etc., are considered which helps in obtaining high accuracy of positioning. The analysis was performed using a basic PPP model that used un-differenced dual-frequency code and phase measurements which is converted into ionosphere-free linear combination for eliminating first order ionospheric delay [7] [8].

The basic PPP model [7] is given by

$$\mathrm{PR}_{\mathrm{IF}} = \frac{f_1^2\,PR_1 - f_2^2\,PR_2}{f_1^2 - f_2^2} = \rho_r^s + c(\delta_{r,clock} - \delta_{clock}^s) + T_r^s + \varepsilon_{PR_{IF}} \tag{2.1}$$

$$\Phi_{\mathrm{IF}} = \frac{f_1^2\,\Phi_1 - f_2^2\,\Phi_2}{f_1^2 - f_2^2} = \rho_r^s + c(\delta_{r,clock} - \delta_{rclock}) + T_r^s + \lambda_{IF} N_{IF} + \varepsilon\Phi_{IF} \tag{2.2}$$

where, $\mathrm{PR}_{\mathrm{IF}}$ and Φ_{IF} are ionosphere-free linear combination for code and phase measurements respectively, f_1 and f_2 are the frequencies of the used signals, PR_1 and PR_2 are pseudo ranges for the code measurements at f_1 and f_2 frequencies respectively,

Φ_1 and Φ_2 are pseudo ranges for the phase measurements at f1 and f2 frequencies respectively, ρ_r^s is true geometric range between phase centre of the satellite antenna in the emission time and phase centre of the receiver antenna in the reception time, c is the speed of light, $\delta_{r,clock}$ and δ_{rclock} are the receiver and satellite clock offsets respectively, T_r^s is the slant tropospheric delay, λ_{IF} is the wavelength for the ionosphere-free linear combination, N_{IF} is the real value of ambiguity ionosphere-free linear combination(cycle), εPR_{IF} and $\varepsilon\Phi_{IF}$ are other errors which include noise, multipath receiver and satellite antenna phase centres etc.

J. Rene Vazquez-Ontiveros et al. [9], proposed the utilization of GNSS services in the probabilistic Structural Health Monitoring (SHM) of extension structures utilizing the PPP technique. SHM have been on the expansion for exploring the harm or peculiarity distinguishing proof to characterization and structural assessment. PPP is extremely famous nowadays as it requires only a solitary GNSS receiver to consider the stochastic conduct of displacements and their relating probability density functions. The outcomes indicated that the PPP-GNSS might be considered as a solid option for SHM because of its millimeter displacement.

2.4 Real Time-Kinematic Positioning (RTK)

RTK Positioning is a satellite navigation error correction technique like the PPP. It is used to enhance the precise position of an object by using the data from the GNSS systems such as GPS, GLONASS etc. The phase of the signal's carrier wave is measured in addition to the information derived from the signal and relies on a single reference station that provides up to centimeter-level accuracy. RTK is based on the use of carrier measurements. It consists of a base station whose position is precisely known and a moving rover whose position is to be determined so that the positioning errors can be eliminated [10]. The GNSS signals are received at the base station and this information is used to correct measurement data of the rover. The signal is perturbed by ionosphere, troposphere and imperfections related to ephemerides, clocks and multipath. This results in the calculated position being different from the actual position. This means that the correction data is affected by uncertainties. The factors that affect the uncertainties can be distance-dependent, systematic, random, frequency-dependent etc. [11]

The high accuracy of real time positioning of objects in motion was considered to be a difficult task of engineering geodesy for 10-15 years. A positioning accuracy of 1-3 cm was achievable with the use of geodetic L1/L2 receivers. The measuring instruments consist of rovers with a local reference station (base station). The cost of both the devices was found to be approximately around 30,000 to 50,000 USD [3]. This made the buying of devices difficult for general customers. The use of low-cost devices with single frequency receivers and kinematic real time positioning with the accuracy of centimeter to decimeter levels could be evolved into a mass product. The positioning was determined by means of computing the static raw data with the single frequency receivers using a RTK GNSS Software such as the RTKNAVI. This described the system and the software for the real-time solution of tracking land-based or air-borne objects which provides an efficient and stable positioning [3] [12].

There is an expansion in the utilization of GNSS positioning for surveying and other geomatic applications. Low-Cost Single frequency receivers have been used widely for navigation applications [12]. María S. Garrido-Carretero et al. [13], utilized the ISO 17123-8 for assessing the performance of two GNSS receivers – the geodetic dual frequency Leica GS10 with AS10 antenna using a RTK network solution, and the low-cost single frequency u-blox NEO-M8P using a single-base RTK solution. The ISO 17123-8 standard was pointed toward indicating the field strategies to be embraced for determining and assessing the exactness of GNSS measuring systems in RTK. When considering the different sources of uncertainty and influence quantities, the results indicated that the geodetic receiver gave combined uncertainties of the order ± 2.5 mm for both the horizontal and vertical coordinates which satisfies the limits for high precision applications. In the case of low-cost receivers, the uncertainties were close to ± 5.5 mm and ± 11 mm for horizontal and vertical coordinates respectively. The low-cost receivers provide a competitive positioning performance compared to geodetic receivers in real-time positioning for short baselines.

Domingos Sárvio Magalhães Valente et al. [14], analyzed two low-cost Emlid "Reach RTK" and the NavSpark "NS-HP" GNSS Systems for the positioning accuracy and precision. F-Test statistics was performed, and the rover unit was placed on a field robot which was controlled manually by a remote control on a specifically designed test track in six repeated trials. The value of the accuracy of the two systems was compared with a pre-fixed measurement distance between the GNSS antennas on the robot. Out of the six trials, three trials were in fixed solution status and achieved an expected Root Mean Square Error (RMSE) range of around 50 mm. In two other trials, the GNSS operated in float solution status which achieved the expected range of 1 m. In one trial, the

NavSpark GNSS was working in the false fixed position trial. This can be rectified by making improvements in signal conditioning, software and reducing the noise. The result implied that the Emlid GNSS had better localization performance of about 94% in the fixed solution status while the NavSpark GNSS showed only 71.5%. Artificial Neural Network (ANN) was used for detecting the false fixed solutions for a single GNSS receiver.

In [15] Nang Van Nguyen et al. evaluated and made comparisons on the static performance of low-cost multi-frequency GNSS devices based on RTK corrections. Multi-GNSS receivers which operate on both L1 and L2 frequency satellite signals from all the accessible satellite constellations are currently available in the market. Static measurements were carried out with four different GNSS receivers: low-cost ZED-F9P module, Eclipse P326 OEM module, A325 smart antenna and SPS855 modular receiver. These devices were placed on the roof of a building under open space scenario. The evaluated results show that the low-cost receiver had a horizontal accuracy of 1 cm to 2 cm using RTK positioning which were good enough for stationary applications of these receivers such as soil sampling or machine path planning in agricultural field.

GNSS RTK Base Station:

The base station consists of a GNSS receiver, antenna, radio transmitter and a power supply. The station should be placed at a fixed and known location. The receiver of the base station tracks the satellite signals simultaneously with the rover. The errors that occur in the GNSS systems would be observed at the location of the base station. The base station then sends the correction data to the rover via radio transmission. This data is then used by the rover to correct its real time position, thereby achieving accurate positioning.

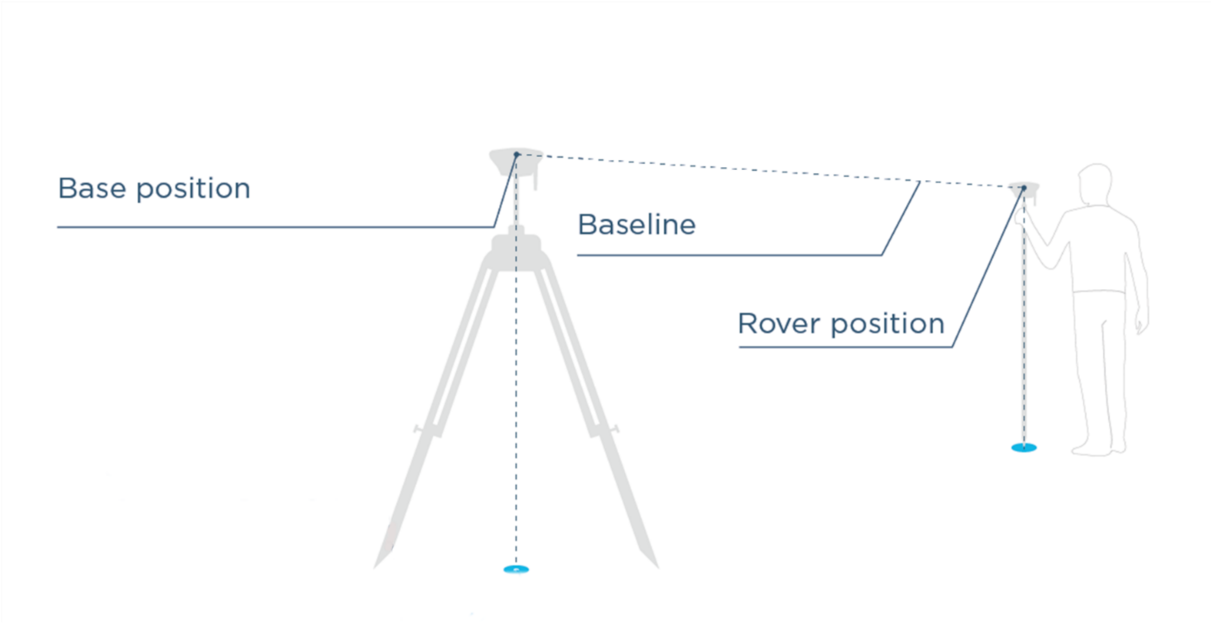

Figure 2.1: GNSS RTK Setup with Base station and rover positions [16]

A basic GNSS-RTK has a GNSS receiver and a GNSS base station. The rovers receive the available signals from the satellites, and the base station sends correction signals to the receiver. These satellite signals contain information such as the latitudes, longitudes, number of satellites available etc.

The GNSS receivers can be fixed on automobiles or load carriers at factories which can provide us information on the location of objects. GNSS systems have primarily found their usage in various fields including navigation, transport, and logistics.

2.5 Satellite Based Augmentation System (SBAS)

Satellite Based Augmentation System were designed to enhance the GNSS services and are mainly used in civil aviation as it provides finer horizontal accuracy and vertical guidance [17]. The architecture of SBAS consists of four segments:

(i) A network of ground reference stations which observes the performance of the GNSS satellites.
(ii) A communication network which is used for transmitting data between the segments.
(iii) A master station which measures the differential corrections to be sent to the user.
(iv) Uplink stations which sends the correction data to the satellites which is then passed on to the user.

The development in advanced satellite navigation enables a new class of aircraft instrument approach procedures. These approaches are based on regional augmentation systems that broadcast corrections over geostationary satellites. Improved GNSS navigation solutions that use corrections from satellite provide the precision and accuracy needed to perform vertical and lateral angular approaches to specific runways. This allows for a cost-effective and a simple process generation even for small airports with low descent. It also supports high-precision route navigation. To further improve the accuracy and reliability of the system, a second frequency in the L-Band radio is being introduced which would correct even the ionospheric errors [18].

The main factor that affects the accuracy of a satellite-based augmentation system is the ionosphere. A study on the accuracy and integrity of European Geostationary Navigation Overlay Service (EGNOS) was done at the EPOD – Olsztyn Airport in Poland by G. Grunwald et al, [19]. The parameters which define the positioning was done in three methods: the original EGNOS ionospheric correction, correction by Klobuchar algorithm and modified ionospheric corrections by Centre of Orbit Determination in Europe (CODE). The results clearly indicate that the original EGNOS ionospheric correction gave the best accuracy and integrity. The CODE and Klobuchar models can be used only for non-precision approach.

Andrzej Banachowicz et al. [20], had done research on monitoring the cargo using GPS/EGNOS multi-GNSS. When the goods are transported within a city, the transport firm needs to have the control on where the goods are at the time of delivery as part of the safety system. Since urban areas are densely built, this may cause signal attenuation due to the reflections and multi-path effects. The EGNOS system superimposes the GPS and GLONASS systems which allow improvements in terms of signal accuracy, reliability etc. Tests have been made at three measuring stations to determine the positioning accuracy of the GPS/EGNOS system in urban areas. The results showed that the implementation of GPS/EGNOS system in practice is not possible for continuous vehicle monitoring in urban areas. The accuracy of a position obtained from this method depends solely on the position of the antenna location. The antenna showed accurate results in open areas while the results in urban areas were far from the true value.

2.6 Other Services and Techniques implemented

X.Luo, S. Schaufler et al. [21], conducted research on the advantages of Galileo to high precision GNSS positioning using the RTK, PPP and post-processing techniques. The estimating conditions, for example, open sky, multipath conditions, and tree shelter were considered. The Galileo GNSS when incorporated with the RTK technique improved the fix accessibility and diminished the errors of decimeter and meter range positioning levels that were caused by uncertainty fixes in testing conditions. The Galileo GNSS with PPP showed the improvement in convergence time and positioning accuracy.

PPP-RTK technique can be applied to the use of multiple GNSS systems. This improves the positioning performance of single-frequency positioning [22]. With the use of a single GNSS system, the convergence time is expected to be very high. However, this can be reduced with the simultaneous use of multiple GNSS, which can reduce convergence time and provide high precision positioning and accuracy. The PPP-RTK technique was applied, and data was processed from GPS, BDS and Galileo separately as well as combined. The receivers used in the research range from expensive multi-frequency geodetic receivers to low-cost single-frequency devices. In the case of GPS, the 90% of the horizontal positioning error was observed to be 5 cm less after a time of 103 mins. In the case of the combined GNSS the time was reduced to 66 mins which can be further reduced by using a single receiver ambiguity resolution [23].

Next to GPS, BDS is highly operational in the Asia-Pacific region [24]. A single GNSS receiver was applied with a dual frequency GPS+BDS data at a distance of 110 km. The satellite clocks and satellite phase hardware biases are acquired from a baseline in Perth, Australia. The precise orbits for GPS and BDS were taken from IGS and Wuhan University respectively. The GPS with PPP needed a convergence time of 1 hour for achieving decimeter level accuracy while the BDS took around 2.5 hours. This problem arises due to the substandard geometry of the BDS satellites. With the combination of both GPS and BDS the convergence time was reduced to an average of 30 minutes. When the satellite phase biases were corrected, the GPS+BDS PPP-RTK solution achieved an accuracy of few centimeter level. The accuracy of components of horizontal position was at 1 cm level while in the vertical position it was at 5 cm level.

BeiDou navigation satellite System (BDS) began providing global positioning, navigation, and timing services from late 2018. Yize Zhang et al. [25], assessed the performance of the new BDS GNSS regarding Single Point Positioning (SPP) and RTK. The Signal in space range error (SISRE) of the BDS-3 satellites were better with an accuracy of 0.71 m whereas the BDS-2 satellites have an accuracy of 0.97 m. In SPP, the RMS value showed an accuracy of 1.1 m in horizontal and 2.2 m in vertical, which is almost as same as the GPS service. The inaccurate BDS ionosphere correction showed a worse positioning precision at B2a frequency band. Increase in the number of new BDS satellites showed an improvement in the baseline accuracy in RTK.

Integration of multi-GNSS has many complications in terms of stochastic modelling. Berkay Bahadur and Metin Nohuctu [22], proposed a filtering approach which is a combination of robust Kalman Filter and variance component estimation for indicating the advantages of multi GNSS observations in single frequency positioning. The robust Kalman filter confronts impact of unexpected outliers by stating an equivalent covariance matrix and the multi GNSS observation variances were determined by the variance component estimation. The proposed filtering approach produced an improved positioning accuracy of 18.5% on single frequency positioning as it determines the variances more rigorously.

Ambiguity Resolution (AR) is a primary issue for high precision positioning using carrier phase measurements. It is the process of resolving the ambiguities which include the changes in the geometry caused by the motion of satellites and receivers. It is a nonlinear integer mixed problem. There are many AR methods that have been used for ground-based positioning system. The known-point-initialization (KPI) calculates the integers ambiguities based on initial measurements measured accurately at the initial coordinates. However, measurements in this method are quite complex in practical applications when there is a loss in the signal [26].

2.7 Summary of Literature

- GNSS systems comprises of three major segments: Space segment, Control Segment and User segment. Every system (GPS, GLONASS etc) consists of its own satellite constellations.
- Different factors like the physical location of the system, the atmospheric conditions and satellite positions and their line-of-sight affects the accuracy of GNSS receivers.
- Many ambiguity resolving methods and filtering approaches have been used to ensure high precision and accuracy. Different positioning techniques like PPP, RTK, and post-processing are available which has its own advantages and disadvantages.

- PPP technique was found to be feasible and more accurate as it uses the products made available by the IGS and has no communication with a receiver station. SF-PPP achieved very accurate results even with the use of low-cost receivers. RTK technique is still being developed to improve accuracy with low-cost receivers which could transform into a mass product.
- RTK technique consists of a reference base station at a known location which observes the satellite signals for any errors. It then sends positional data corrections to the receiver in real time. It was found to be effective in case of tracking positions of various objects in real time.
- Satellite Based Navigation Systems were developed to enhance the accuracy mainly in area of civil aviation. The main factor that affects the accuracy of SBAS is the ionosphere.
- EGNOS is a satellite-based augmentation system developed by European Space Agency (ESA). EGNOS superimposes the GPS and GLONASS systems but failed to achieve accurate results when it was used for tracking continuous movement of vehicles in real time. The signals get affected by the interferences and noises in semi-obstructed areas.
- The conventional GNSS positioning method was difficult to employ in urban areas or inside buildings. This is because there occurs a loss in the GNSS signals due to the reflections and noise.
- The convergence time is too long for some of the services. Several researchers have proposed methods to reduce the convergence time for different GNSS services which can compute the position of the objects in a smaller amount of time.
- Many GNSS services lack in accuracy and precision. For example, implementation of multi-GNSS systems has some complications which can be solved by using a filtering approach.
- The use of combined multi-GNSS systems (e.g.: GPS+Galileo or GPS+BDS+Galileo) achieved better results when compared with the use of single GNSS system.

2.8 Objective of the Thesis

- Propose a concept of indirect tracking of objects by repeatability tests.
- To evaluate the performance of the Fieldbee GNSS receiver in different scenarios.
- To track different objects in an indirect approach by evaluating the repeatability of positioning.
- The solution would be economically feasible with the Fieldbee device for indirect tracking.

3. METHODOLOGY

3.1 Hardware Part

Several companies were approached by Eliko to find the required equipment for the research which would cost in the range of 1000-3000 euros. The products which provide PPP service had slow convergence time which is not suitable for real-time applications. SBAS and other techniques are less accurate than the RTK and can't guarantee an accuracy of less than 50 cm.

Table 3.1: Cost comparison of different receivers

Companies offering GNSS systems	Quotation Description
Hexagon AB	The price varies depending upon options loaded on the receiver and starting price is around 9000 euros.
Fixposition AG	The cost of the equipment was about 1900 euros excluding the base station.
Oxford Technical Solutions	The initial price starts from 7000 euros.
SBG Systems	There was no base station included and depending upon the performance and features the prices were in the range of 2K to 70K euros per unit approximately.
Septentrio N.V.	The company offered the base station and receiver separately costing around 6000 euros for each.
Fieldbee	The cost of the Fieldbee system is 2900 euros which consists of both the receiver and the base station.

For the scope of the thesis, the Fieldbee GNSS L2 receiver and Fieldbee RTK GNSS L2 Base station is used. According to the use case, the cost can be calculated per forklift as the base station would serve as a common unit for many receivers. The equipment from Fieldbee costs at 2000 euros per forklift which was well within the budget.

3.1.1 Fieldbee GNSS L2 receiver

The receiver works in both L1 and L2 frequency bands and receives the RTK corrections from the base station. They also support local NTRIP (Networked Transport of RTCM via Internet Protocol) providers such as FLEPOS, SAPOS, MoveRTK etc. This receiver was

made to achieve high precision tracking despite mountains and various other signal obstructions and costs around 1850 euros which is economical compared to the receivers mentioned above.
(Device datasheet - https://www.fieldbee.com/wp-content/uploads/2020/10/FieldBee-L2-RTK-GNSS-receiver-Datasheet-26.10.20.pdf)

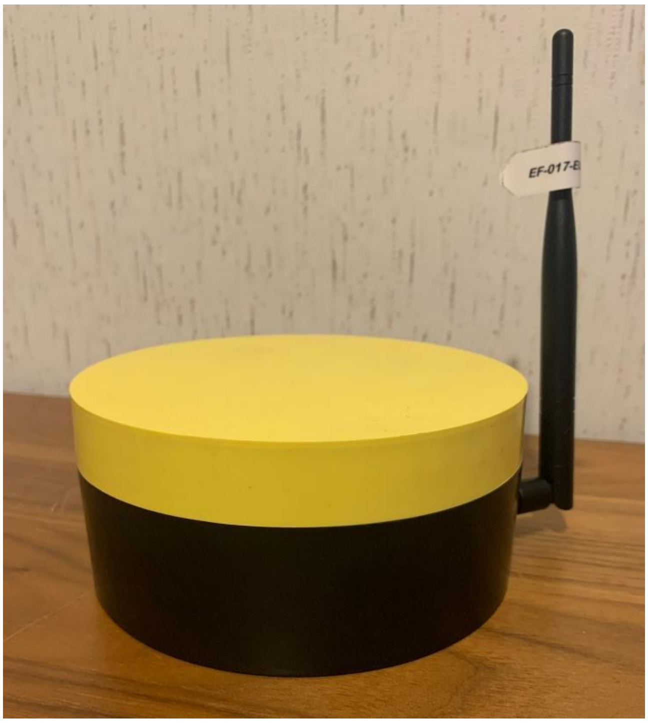

Figure 3.1: Fieldbee L2 Receiver

The characteristics of the Fieldbee L2 RTK receiver are:

- It has a range of 20+ km via Internet and 2.5+ km via Radio.
- It can be combined with various GNSS systems (GPS, GLONASS, BeiDou and Galileo).
- With RTK correction, it can achieve an accuracy of $\pm$ 1 cm.

The FieldBee Receiver outputs coordinates in the form of NMEA messages. Example of NMEA message is $GPGGA,161954.60,5924.6736292,N,02440.7096354,E,1,06,3.77,-1.873,M,18.557,M,,*70

For the purpose of the thesis, we have chosen the $GPGGA message type which outputs the global position of the object.

Table 3.2 shows the format of data acquired with information on latitude, longitude, UTC time etc.

Table 3.2: NMEA Format

Name	Data	Description
Sentence Identifier	$GPGGA	Global Positioning System Fix Data
Time	161954.60	16:19:54 UTC
Latitude	5924.6736292, N	59d 24.6736292 N or 59d 24' 40'' N
Longitude	02440.7096354, E	24d 40.7096354 E or 24d 40' 42'' E
Fix Quality: - 0 = Invalid - 1 = GPS Fix - 2 = DGPS fix	1	Data is from a GPS fix
Number of Satellites	06	6 satellites are in view
Horizontal Dilution of Precision (HDOP)	3.77	Relative Accuracy of horizontal position
Altitude	-1.873M	1.873 meters below mean sea level

3.1.2 Fieldbee GNSS L2 Base Station

The Fieldbee GNSS L2 Base station was designed to achieve very high accuracy and precision including in areas with high hills and places where signal obstructions may occur. The base station serves as a reference point which can monitor the signals received from the satellite and positional corrections in position are made to another antenna in real time over the UHF. It transmits RTK correction over a radius of 25+ km over the internet and a radius of 2.5 km over the radio which helps to achieve an accuracy of ±1 cm. It supports multi-constellation type including GPS, GLONASS, BeiDou etc. The base station has an average of 1 s time to first fix average on an open sky during hot start. The base station receives the first RTK correction under 60 seconds as per technical specifications.

(Device datasheet - https://www.fieldbee.com/wp-content/uploads/2020/10/FieldBee-L2-RTK-GNSS-Base-station-Datasheet-26.10.20.pdf)

3.2 Software Part

3.2.1 Python and R

Python is an interpreted, general programming language which had its initial release in 1991. It is a high-level language which helps the programmers to write a logical and a clear code for various projects. It supports object-oriented and functional programming and is considered to be a popular programming language due to the various inbuilt libraries.

The reason for choosing Python as the programming language was due to the same reasons mentioned above. Many libraries such as *math*, *matplotlib* proved to be exceptional means for performing various calculations and for plotting the required graphs. *os* library in Python was helpful for handling files in the system.

R is a simple and high-level programming language developed initially in 1993 [27]. It is used for analyzing the data and making statistical computations. It is also used for visualizing the data and consists of graphical functions which is used for generating plots as per our design. The programming is done on the interpreter RStudio. For the purpose of this thesis, R language was very useful in visualizing the coordinates on the graph and the mean distance between the points taken with and without the base station.

3.2.2 Collection of Data

data_collect.py is a basic program written by the author for collecting the positional data (latitudes, longitudes) from the GNSS receiver. The user decides an area or location where the performance and accuracy of the receiver is to be determined. The analysis was made to see the behaviour of the system in different scenarios. For example, in good conditions with no buildings around, like empty fields (open space) vs conditions where there are some buildings around but not close to the receiver (semi-obstructed) and bad conditions where there are tall buildings nearby (obstructed). This comparison helps to see how accurately the receiver can receive the GNSS coordinates from the satellite signals.

Initially, *socket* library was used to receive the data from the receiver over Wi-Fi. Because of certain problems such as not knowing the correct port number from the manufacturer, the receiving of data from the receiver was made with the help of serial connection. *serial* library in Python helps to establish a serial connection between the receiver and the software on the computer.

The program is written to receive data from the GNSS receiver, separate the raw coordinates which are then stored in a text file in a particular format. After that the information is imported for the calculation of various parameters.

3.2.3 True Coordinates and Receiver Coordinates

In order to determine the true coordinates of the test location, we chose certain objects such as a lamppost, a pole etc., which were visually seen on the map as the measurement point. This helps us to determine the precise coordinates of that location which is then compared with the coordinates extracted from the receiver. The true coordinates were taken from the X-GIS (Geographical Information System) which gives us all geographical information including geodetic coordinates in Estonia.

The receiver coordinates present the latitudes and longitudes in the form of degree minutes and seconds, while the true coordinates are in the form of decimal degrees. For the purpose of calculation, the written program converts the coordinates from the receiver into the form of decimal degrees.

3.2.4 Calculation of Data

data_calc.py is a program, which enables us to calculate the various parameters for evaluating the accuracy. The receiver coordinates need to be converted into the cartesian coordinate system to have the results in a user readable form.

In the coordinate system:

- The x-axis passes through the longitude and latitude at (0,0)
- The y-axis passes through the longitude and latitude at (0,90)

Thus, the function to convert the latitudes and longitudes into its respective coordinate form is [28]:

$$\text{latitude } (x) = R * \cos(\text{latitude in radians}) * \cos(\text{longitude in radians}) \quad (3.1)$$

$$\text{longitude } (y) = R * \cos(\text{latitude in radians}) * \sin(\text{longitude in radians}) \quad (3.2)$$

3.3 Performance Metrics

3.3.1 Accuracy

The technical definition of accuracy is defined as the degree to which the result of a measurement or calculation is different from the correct or true value. The accuracy of latitudes and longitudes are calculated by finding the difference between the average of all values measured at a point and the true value at that point. This difference value helps us to determine the accuracy of the measured data. If the difference value is low, it means the accuracy on the measured point is high. If the difference is high, it is inaccurate [29].

$$\varepsilon_x = x_{true} - x_{observed} \tag{3.3}$$

$$\varepsilon_y = y_{true} - y_{observed} \tag{3.4}$$

where,

ε_x = error in accuracy of x coordinates

ε_y = error in accuracy of y coordinates

x_{true} = value of true coordinate of x

y_{true} = value of true coordinate of y

$x_{observed}$ = value of observed coordinate of x

$y_{observed}$ = value of observed coordinate of y

3.3.2 Standard Deviation

Standard Deviation is defined as a measurement of the variation of values in a data set. A low standard deviation implies that the values are close to the mean of the data set (i.e., the expected value), whereas a high standard deviation implies that the values are spread out over a larger range [30].

The formula for standard deviation used was

$$\sigma = \sqrt{\frac{\Sigma_{i=1}^{N}(x_i - \mu)^2}{N}} \tag{3.5}$$

where σ stands for Standard Deviation, x_i stands for each value in the coordinate list, μ is the mean value of the list, and N is the total number of values in the list.

3.3.3 Root-Mean-Squared Error (RMSE)

The root mean square error is the measure of difference between the value predicted by a system and the true value. It is the square root of the average of the square of all the error. It is used to compare the accuracy and prediction error of the system [31]. For our thesis, RMSE indicates how close the observed coordinates from the receiver are to the true coordinates of that location. Lower values of RMSE indicate the system is accurate whereas high values of RMSE indicates the system is inaccurate [31].

The formula for RMSE used was

$$RMSE = \sqrt{\frac{\Sigma_{o=1}^{N}(x_t - x_o)^2}{N}} \tag{3.5}$$

where x_t is the true coordinate value, x_o is the observed value from the receiver and N is total number of values.

3.3.4 Distance (Haversine Formula)

Distance is calculated between the true coordinates and the coordinates received from the receiver using the Haversine Formula.

The Haversine Formula is an important equation in the field of GNSS which calculates the distance between two coordinates on the Earth. This formula is found to be accurate as it ignores the effect of ellipsoidal and ignoring high altitude hills and deep valleys. The coordinates should be first converted into radians in order to use the values in the formula. 1 degree = 0.0174533 radians.

The Haversine formula is given by [32]:

$$\Delta_{lat} = lat2 - lat1$$

$$\Delta_{long} = long2 - long1$$

$$a = sin^2 \frac{\Delta_{lat}}{2} + \cos(lat1) * \cos(lat2) * sin^2 \frac{\Delta_{long}}{2}$$

$$c = 2 * atan^2 (\sqrt{a}, \sqrt{1-a})$$

$$d = R * c \tag{3.6}$$

where,

R = Radius of the earth,

Δ_{lat} = difference in latitudes of true coordinates and that measured from the receiver

Δ_{long} = difference in longitudes of true coordinates and that measured from the receiver

c = angular distance between latitudes and longitudes

d = distance

3.3.5 Distribution Plots

The seaborn.displot library in python was used to draw the distribution of the latitudes and longitudes received by the Fieldbee GNSS L2 receiver. This library is used to make statistical graphs which is a high-level interface to matplotlib and represents the overall distribution of the points in a form which is understandable to the reader [33].

The coordinates which are converted to coordinate system (x, y) form are subtracted from their mean which gives us a distribution normalized to 0.

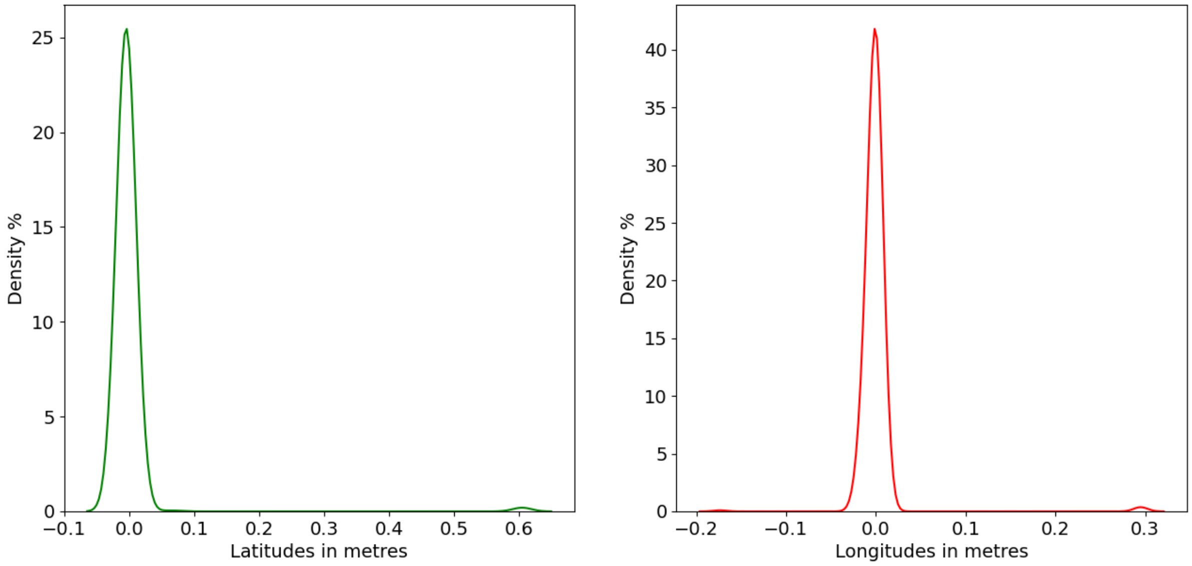

Figure 3.2: Distribution plots of Latitude and Longitude

Figure 3.2 shows the example of a distribution plot of both the latitudes and longitudes in meters. This helps us to understand how accurate the coordinates received from the receiver are distributed. For example, the figure 3.2 shows us that the accuracy of the latitudes obtained are distributed in the range of ± 0.05 m, while the longitudes obtained are distributed in the range of ± 0.04 m.

4. TEST SETUP

In order to evaluate the accuracy of the receiver to be used in Nordwood Sawmill for tracking of objects, several initial tests were done under different scenarios in Tallinn. The different scenarios include open space area, semi-obstructed area and obstructed area.

- The areas where there are no buildings or other signal blocking structures is considered to be Open Space environment. For Example, open fields, playgrounds etc.
- Semi-obstructed areas include places with a few surrounding buildings around which are generally not tall.
- Obstructed spaces are areas where the measurement point is completely surrounded by tall buildings. The places under the roofs can also be considered as an obstructed environment.

Measurements were taken from at different points for each scenario and the data is then fed to the data_calc.py program which helps us to view the results of various parameters.

4.1 Static Testing Setup

The tests were taken at three different locations in Tallinn: Outside Eliko Tehnoloogia Arenduskeskus OÜ (semi-obstructed), Hippodrome (open space) and near the buildings of Tallinn University of Technology (obstructed).

The Fieldbee Base Station and Receiver were fixed on tripods. According to the technical specifications of the Fieldbee GNSS system, the RTK correction mode of the base station has a maximum broadcast radius of 25 km. The base station was set up in the centre of each chosen area at a fixed location. The receivers receive position updates from the satellites for 1 minute. The testing at each point was done twice: once with the base station switched on which gives RTK correction to the receiver and once with the base station switched off so there is no correction present.

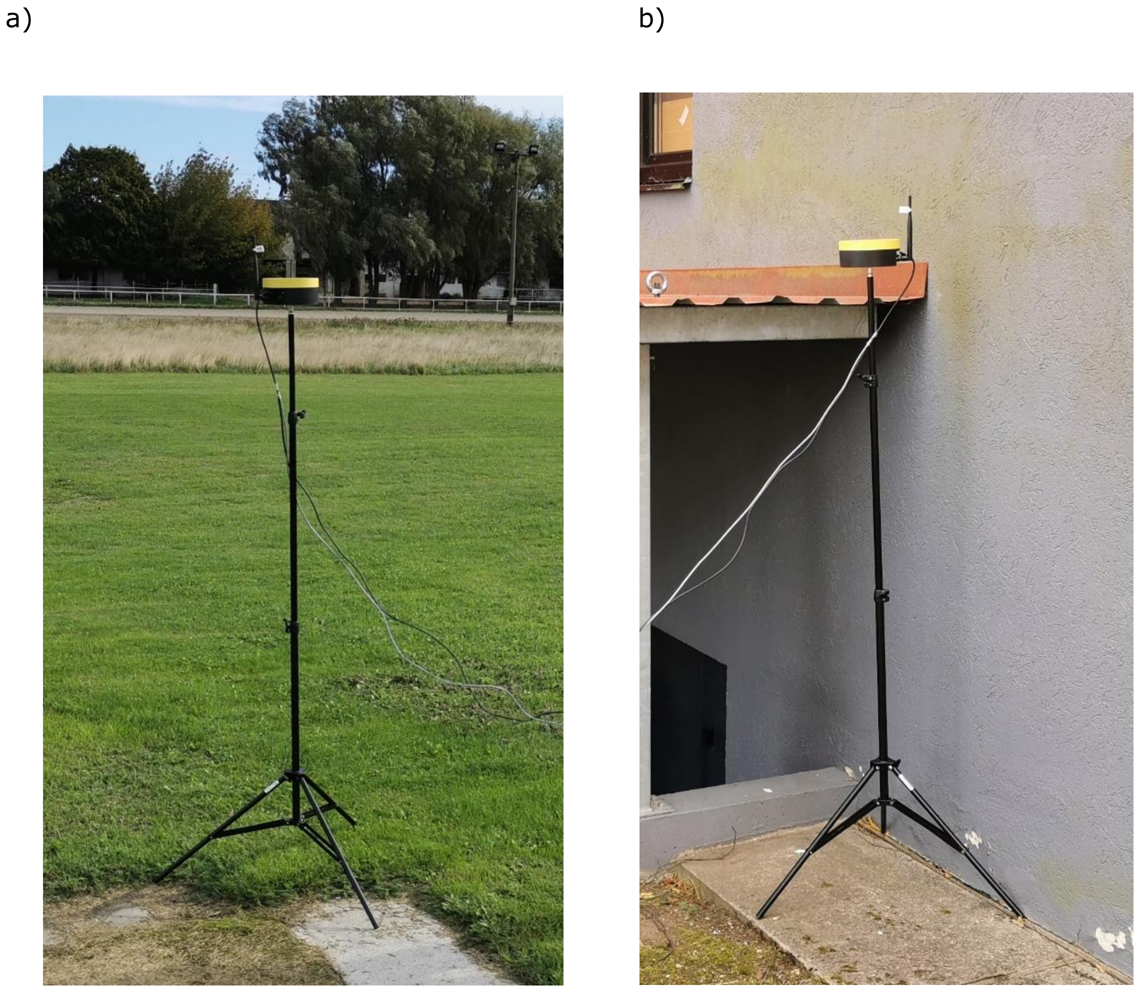

Figure 4.1: Receiver at a) Open Space b) at Obstructed space

Figure 4.1 (a) shows the rover receiving positioning data from the satellite signals under the open space scenario where there are no structures that would block the GNSS signal. Figure 4.2 (b) shows the receiver receiving the data under obstructed areas i.e., near buildings and structures.

4.2 Dynamic Testing Setup

The dynamic testing helps us to visualize how accurate the points are outputted along a certain path. In this test, the base station is placed on a fixed location and the rover is moved along with the tripod on a pre-defined path. The points received are then plotted on a scatter plot to visualize whether they are outputted on the path or if any anomalies are present.

4.3 Static Tests at Nordwood Sawmill

The main use of the GNSS receiver at Nordwood Sawmill is to track the location of the stored wood logs. Since the logs are generally transferred by the forklifts, the test points have been chosen in such a way where the forklifts would travel to pick up, move and then store the logs. The receiver is placed on these locations and the various parameters are evaluated under both conditions: when base station is switched on and when it is switched off.

4.4 Dynamic Tests at Nordwood Sawmill

The dynamic tests are carried out at the sawmill by placing the receiver on a forklift as shown in Figure 4.2. The forklift moves around a certain path while the receiver continuously updates the coordinates in motion.

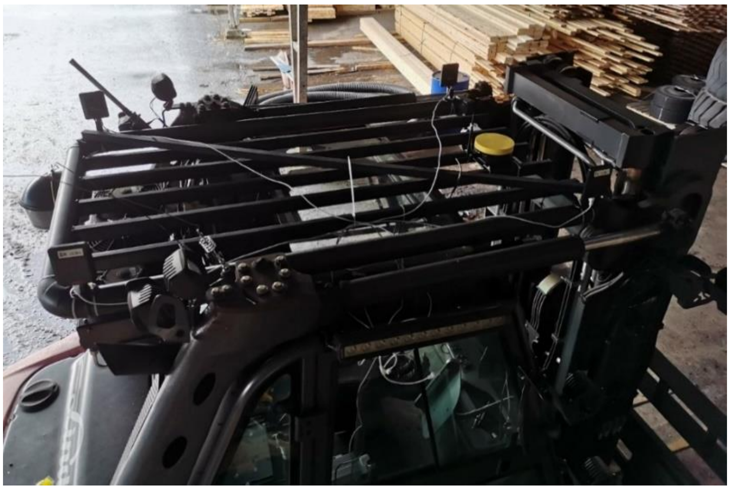

Figure 4.2: Fieldbee GNSS L2 receiver placed on a Forklift

4.5 Concept of Indirect Tracking

As explained in Chapter 1, indirect tracking of objects is carried out by placing receivers on the forklifts instead of on the objects. This would help the industries to track the cargo placed at different locations and to know their position by tracking the forklift. The forklift carries the object and drops it at a particular location. The coordinates at the time of drop down is saved. The vehicle goes to the same location and picks up the object. The coordinates at the time of pick up is compared with the drop-down coordinates. The results were evaluated to see if the rover can output the same coordinates without any errors. The threshold for accuracy value set by Eliko was 50 cm.

5. RESULTS AND DISCUSSIONS

For each measurement points, various parameters including standard deviation, accuracy of latitudes and longitudes, root mean square error were calculated under both conditions: with the base station switched on which provides RTK correction and with the base station switched off which does not provide the corrections.

Due to unavailability of additional high accuracy geodetic devices, the true coordinates were taken from the X-GIS. From the map in Figure 5.1, positioning errors occur due to the differences in pixel and the resolution of the map.

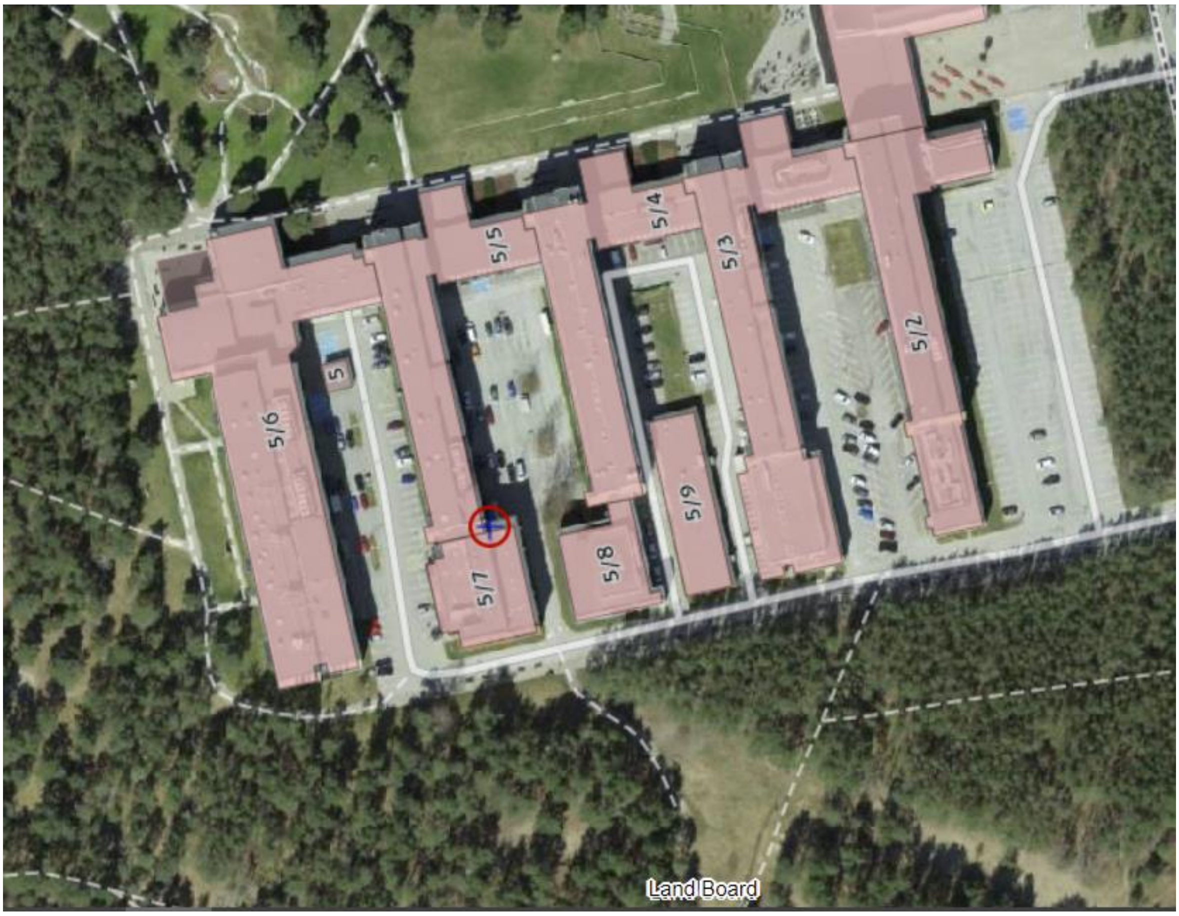

Figure 5.1: Map of X-GIS

The data from the experiments done were plotted on a graph in R-Studio with respect to the true coordinates. From Figure 5.2, it is clearly seen that the current geodetic coordinates taken from X-GIS cannot be used as a reliable source for evaluating the performance. The true coordinates are deviated from the coordinates returned by the receiver by a significant margin of more than 0.5 m.

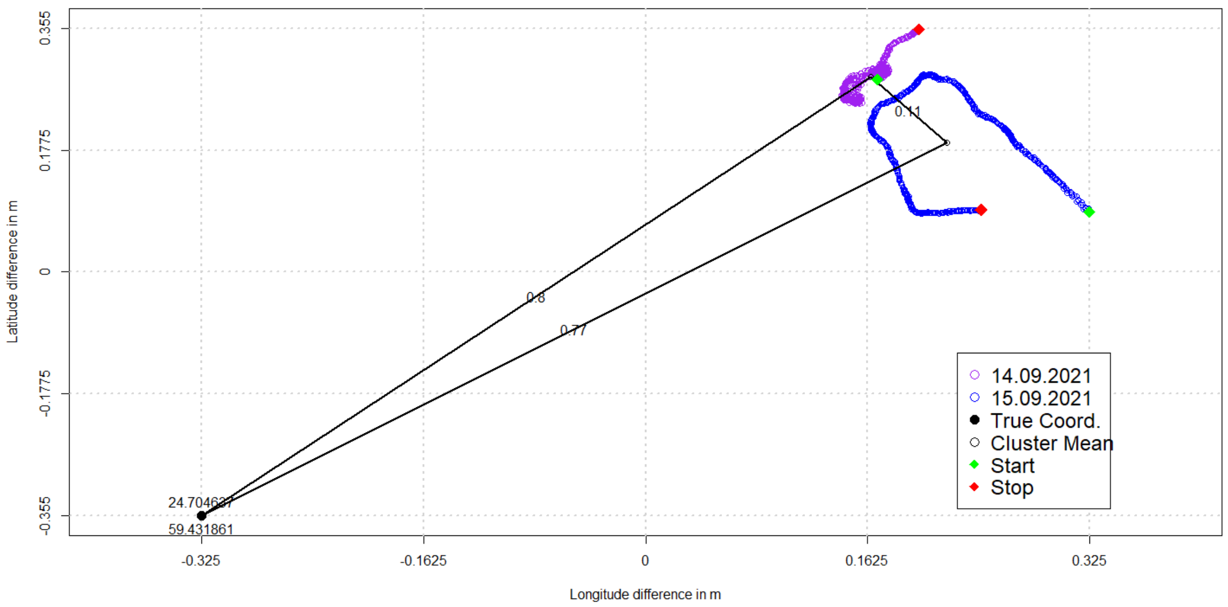

Figure 5.2: Data points with True Coordinates

Measurements were done on 14.09.2021 before noon and on 15.09.2021 afternoon to analyze whether any difference in satellite constellations or atmospheric changes would have an impact on the precision and accuracy of the receiver data.

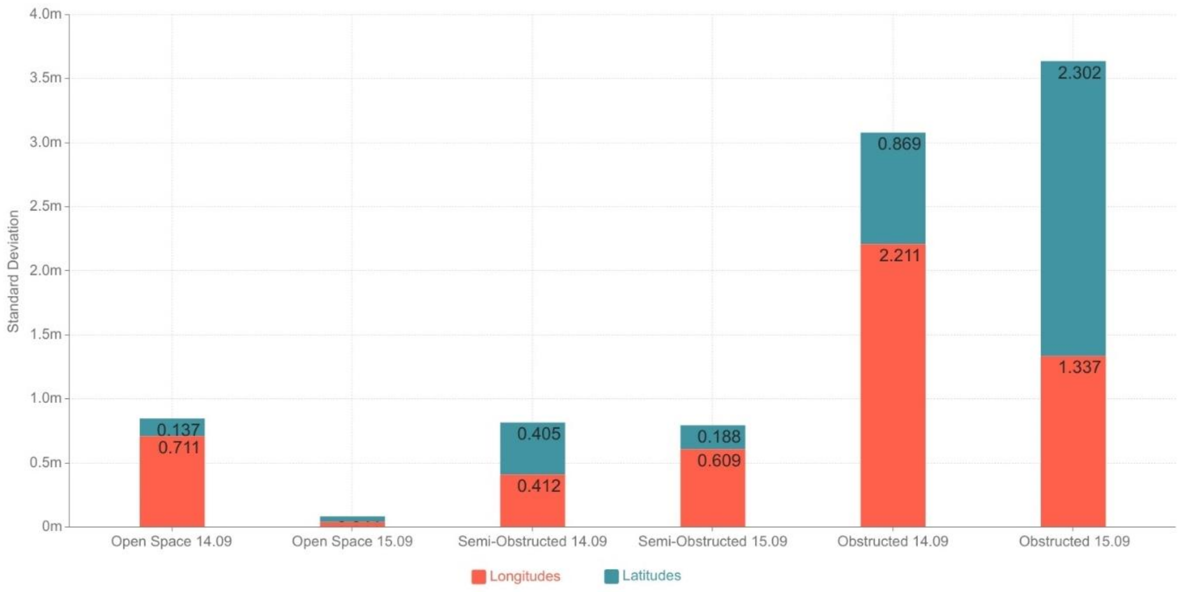

Figure 5.3: Standard Deviation values under different scenarios

Figure 5.3 helps us understand that the resulting standard deviation values of both the latitudes and longitudes measured on different dates under various scenarios does not really affect the results as each scenario gives different deviation values. This implies that the changes in different satellite constellations have less impact on the accuracy of the receiver.

Tests were also done to check the impact of the RTK corrections provided by the base station under different environments. The deviation of both the latitudes and longitudes were plotted on a graph.

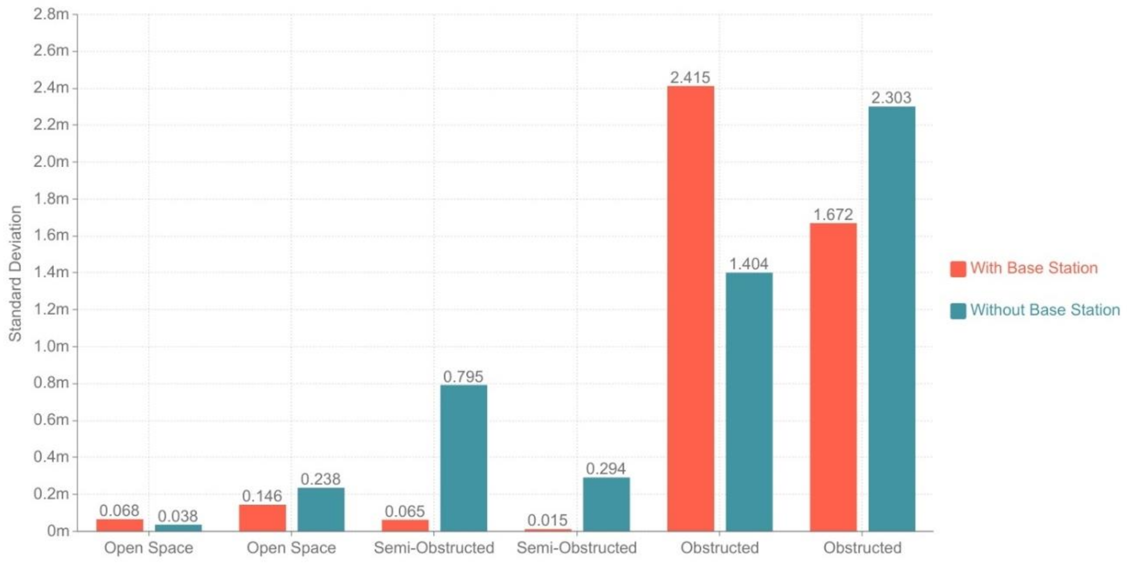

Figure 5.4: Standard Deviation values of Latitudes with and without Base station

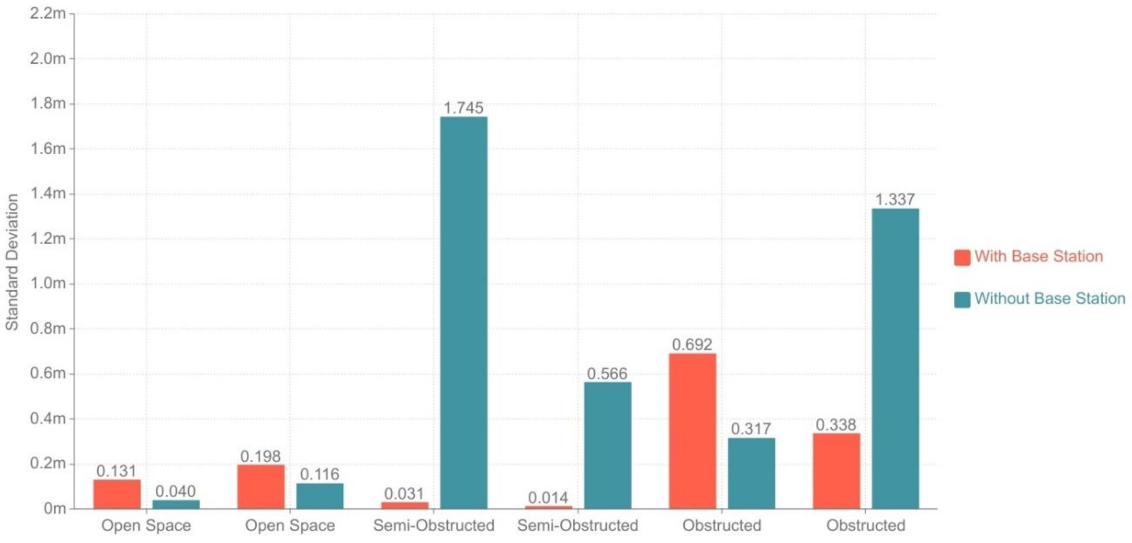

Figure 5.5: Standard deviation values of Longitudes with and without Base station

There was a significant improvement when the base station was used under semi-obstructed conditions (as seen in Figure 5.4 and Figure 5.5). It provided differential corrections to the receiver. It is clearly seen from the plots that with the base station the standard deviation values were smaller and in the absence of it there was an increase in the standard deviation value of the same points under semi-obstructed. In other words, the precision was largely improved with the use of base station. However, there was no impact under open space and obstructed areas. This may be due to the fact that in open space scenario, there is no need of improving the positioning with the base station as the signal quality of the receiver is good. But, in the obstructed area the satellite signals seem to be degraded for both the receiver and the base station. In this case, the base station is unable to provide the RTK corrections to the receiver.

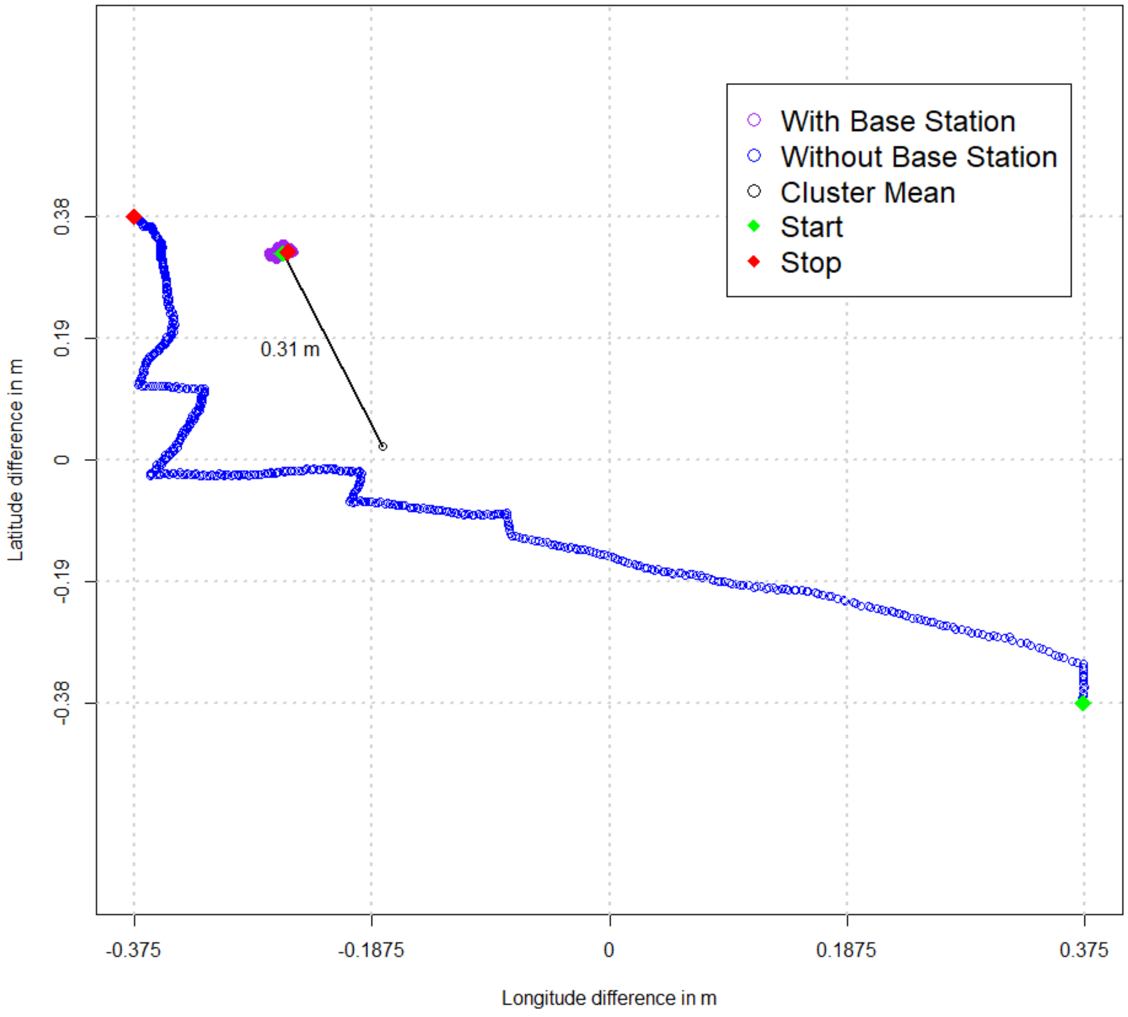

Figure 5.6: Data points with and without Base station (Semi-Obstructed Scenario)

We can see the precision of the points from the Figure 5.6 in the semi-obstructed area. The points are distributed close to each other with the base station (violet dots), while without the base station, the points are distributed with a distance of 30-40 cm-s (blue dots).

5.1 Open-Space Scenario

The initial open-space test was taken in Tallinn at Hippodrome.

Table 5.1: Results of Open Space

	With Base Station	Without Base Station
Average distance between true coordinates and receiver coordinates	0.793m	1.027m
Accuracy of latitudes	0.693m	0.904m
Accuracy of longitudes	0.215m	0.162m
Standard Deviation of latitudes	0.039m	0.137m
Standard Deviation of Longitudes	0.032m	0.071m
RMSE of latitudes	0.694m	0.914m
RMSE of longitudes	0.218m	0.177m

Table 5.1 shows the results evaluated under open space scenario. We can see that with the base station providing corrections, the results do not vary much under open space. In this case the receiver is able to achieve an accuracy of less than 1 m. The results can be taken as an approximate value because of human errors such as identifying the exact true coordinate of the point where the measurements were taken (as explained in the beginning of chapter 5).

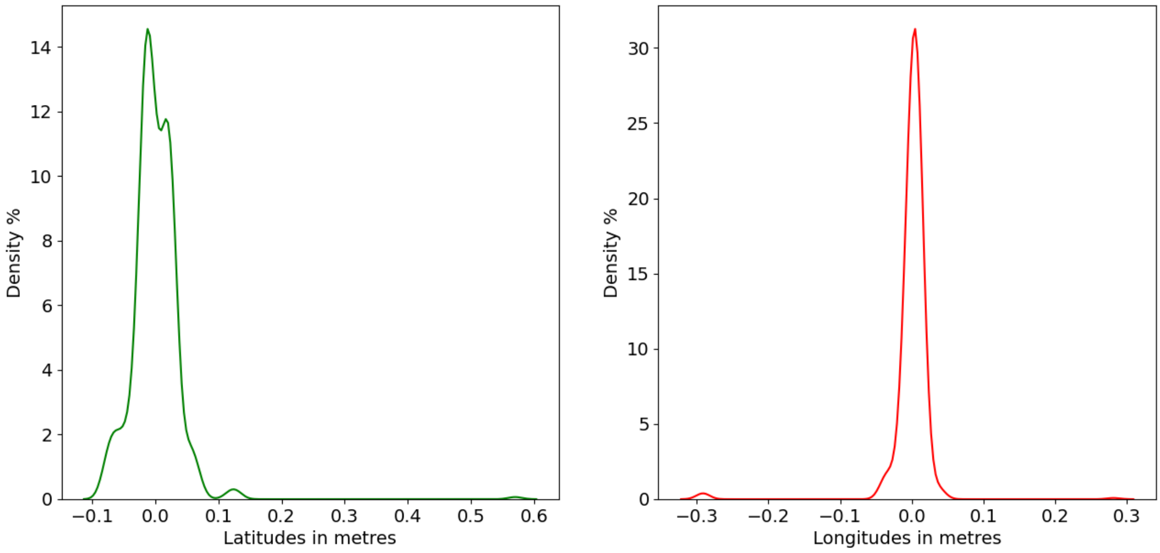

Figure 5.7: Distribution plot with Base Station (Open Space)

Figure 5.7 and figure 5.8 show the distribution plots of both the latitudes and longitudes in the presence and absence of the base station. In the presence of the base station, most of the values are under ± 0.1 m for both the latitudes and longitudes. The distribution obtained was similar to a Gaussian Distribution. In the absence of the base station, most of the values are in the range of ± 0.2 m for both the latitudes and longitudes.

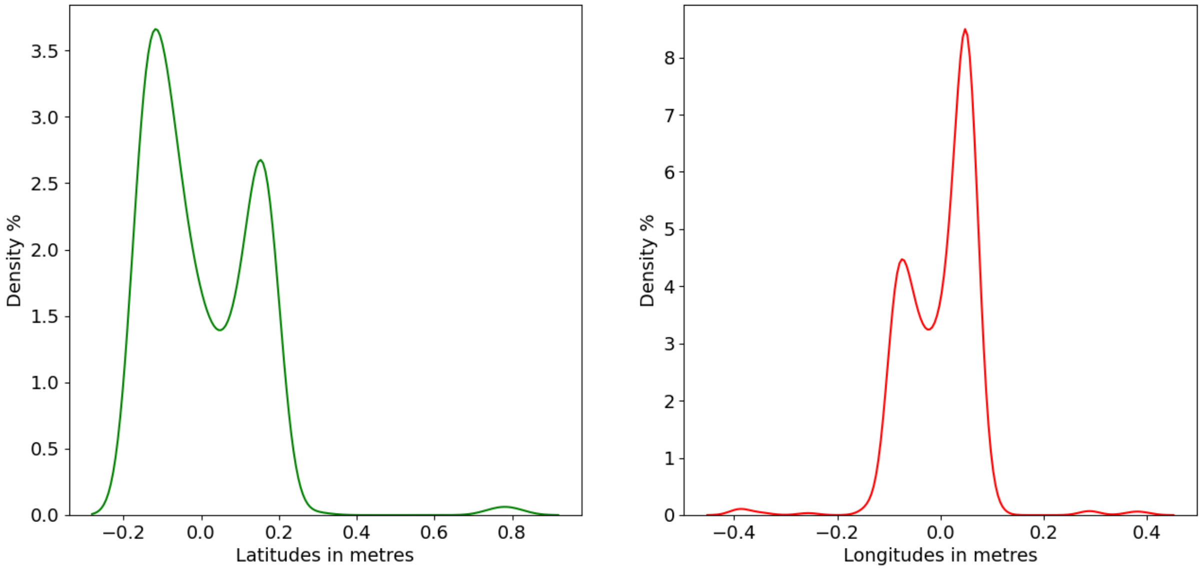

Figure 5.8: Distribution Plot without Base station (Open Space)

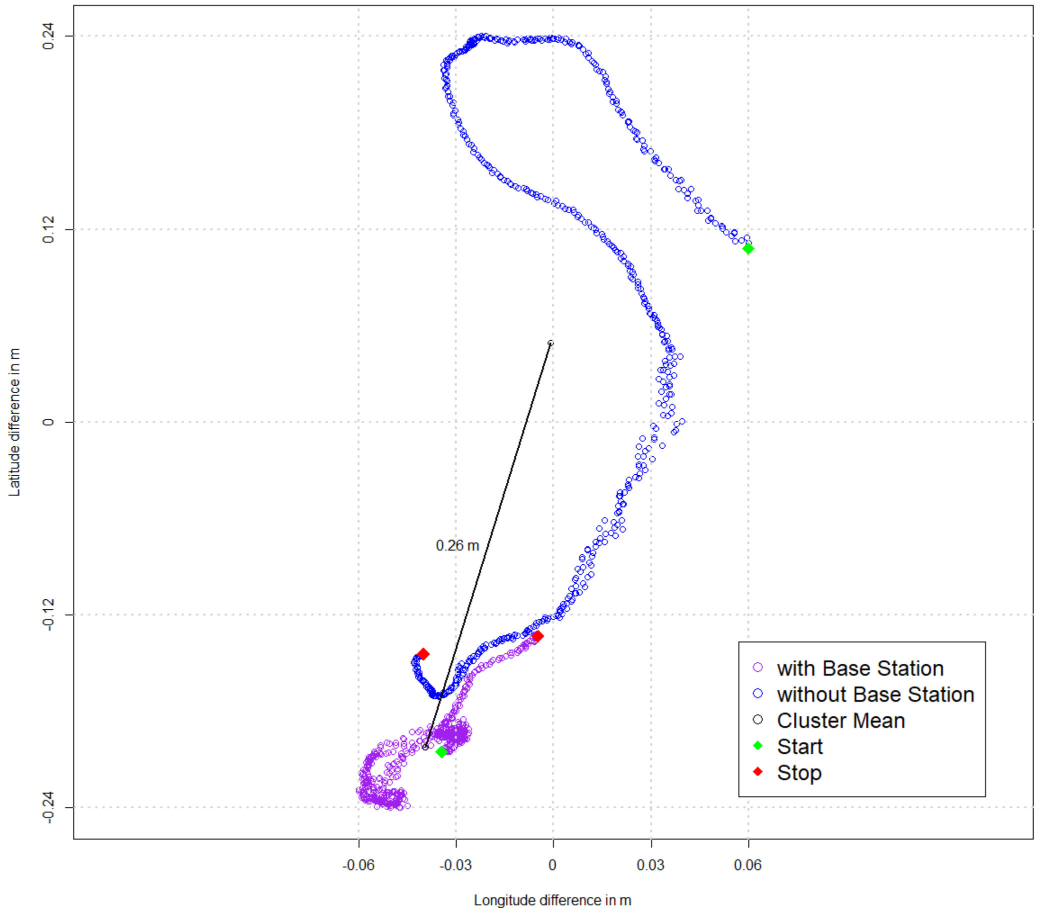

Figure 5.9: Data points collected (Open Space)

Figure 5.9 shows the graphs of the points received from the rover with and without base station respectively. The green dots indicate the first initial reading of the receiver while the red dots indicate the final reading. The blue and violet lines indicate the distribution of the points on the same area with and without the base station. In open space the deviation of the latitudes and longitudes were found to be around 12 cm and 3 cm respectively with the base station.

5.2 Semi-Obstructed Scenario

The initial tests for the semi-obstructed scenario were done outside Eliko Tehnoloogia Arenduskeskus OÜ.

Table 5.2: Results of Semi-Obstructed

	With Base Station	**Without Base Station**
Average distance between true coordinates and receiver coordinates	1.430m	1.834m
Accuracy of latitudes	1.226m	1.300m
Accuracy of longitudes	0.197m	1.232m
Standard Deviation of latitudes	0.064m	0.794m
Standard Deviation of Longitudes	0.031m	1.745m
RMSE of latitudes	1.228m	1.524m
RMSE of longitudes	0.200m	2.136m

Table 5.2 shows the results evaluated under semi-obstructed scenario. Under this scenario, the base station is found to be essential as it provides much better performance compared to without it. Without the base station the accuracy was found to be approximately around 2 meters.

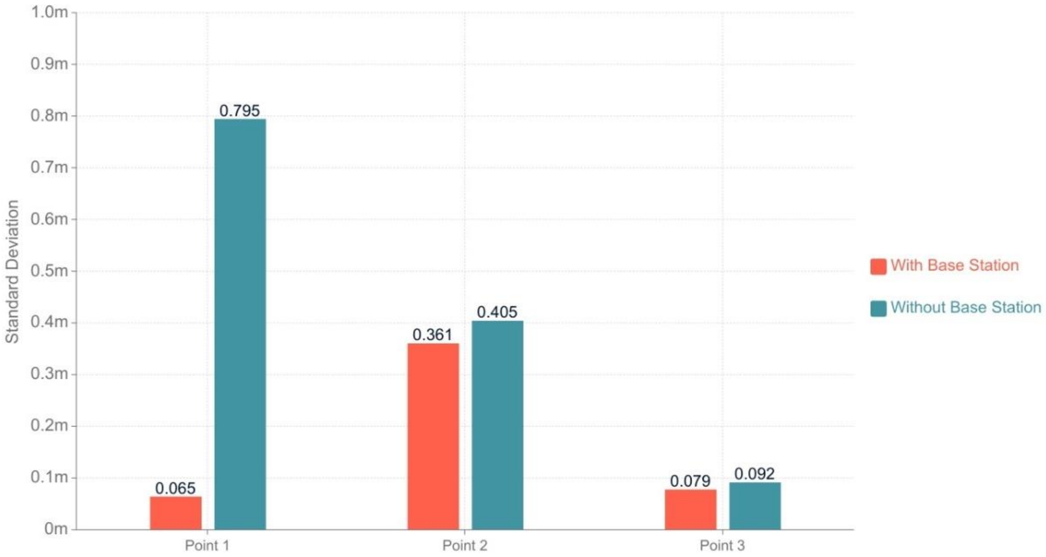

Figure 5.10: Standard Deviation of Latitudes (Semi-Obstructed)

Figures 5.10 and 5.11 shows the deviation of points taken at different locations under the semi-obstructed conditions. The base station proved to be more useful in this scenario. We can see from the graphs clearly that the deviation values of latitudes and longitudes is smaller with the base station compared to the values without it. Thus, it can be stated that the accuracy and precision is much more improved with the help of base station under the semi-obstructed scenario than the open-space or obstructed.

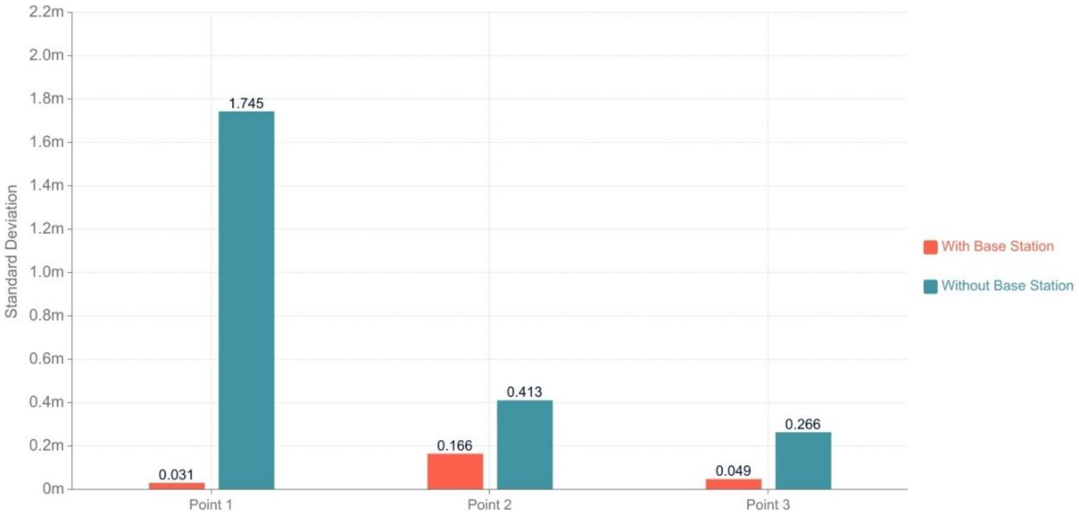

Figure 5.11: Standard Deviation of Longitudes (Semi-Obstructed)

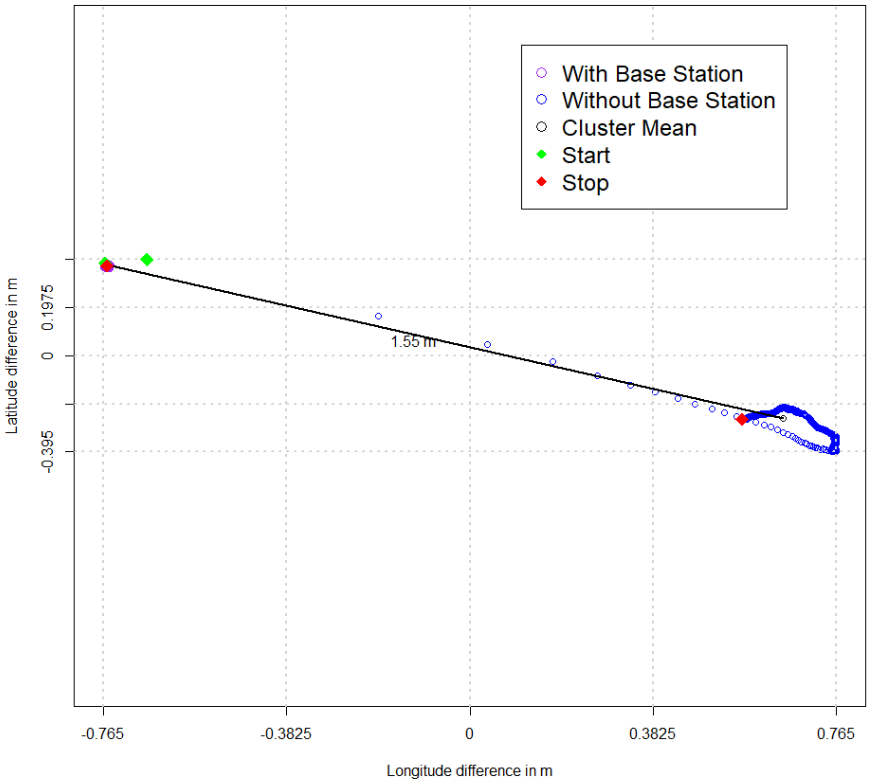

Figure 5.12: Data points collected (Semi-Obstructed)

Figure 5.12 clearly implies that in semi-obstructed area, the RTK corrections provided by the base station improved the accuracy and precision of the coordinates. A lot of anomalies were found in the received data when the measurements were done without the base station. The deviations are found to be less than 10 cm for both the latitudes and longitudes under semi-obstructed scenario when it is used with the base station. Without the base station, the deviations of the coordinates are more than 60 cm.

5.3 Obstructed Scenario

The high buildings near Tallinn University of Technology were considered for measuring the performance of the receiver under obstructed conditions.

Table 5.3 shows the results evaluated under obstructed scenario.

Table 5.3: Results of Obstructed Scenario

	With Base Station	Without Base Station
Average distance between true coordinates and receiver coordinates	2.992m	3.575m
Accuracy of latitudes	0.979m	1.304m
Accuracy of longitudes	2.577m	3.000m
Standard Deviation of latitudes	0.558m	0.457m
Standard Deviation of Longitudes	0.469m	0.266m
RMSE of latitudes	1.126m	1.382m
RMSE of longitudes	2.619m	3.011m

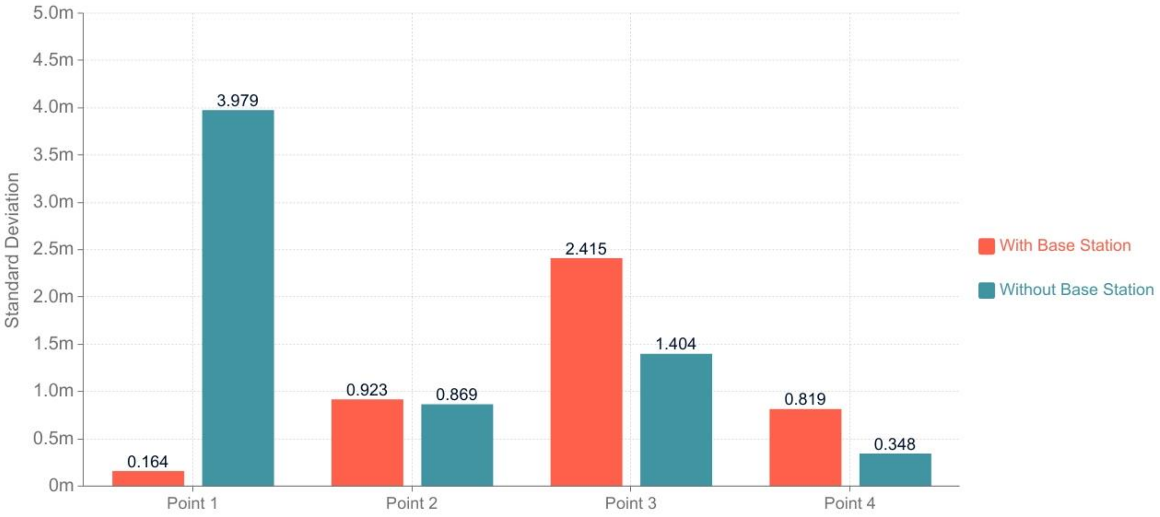

Figure 5.13: Standard Deviation of Latitudes (Obstructed)

Figures 5.13 and 5.14 shows the standard deviation of latitudes and longitudes at different points in obstructed scenario. Due to possible loss of signal and non-line-of-sight conditions in receiving the satellite signals, the deviation of the coordinates was found to be large with or without the base station. As stated earlier, the base station is of minimal use under obstructed scenario. The standard deviation value was around 4 m for the latitudes and around 3 m for the longitudes which is very inaccurate.

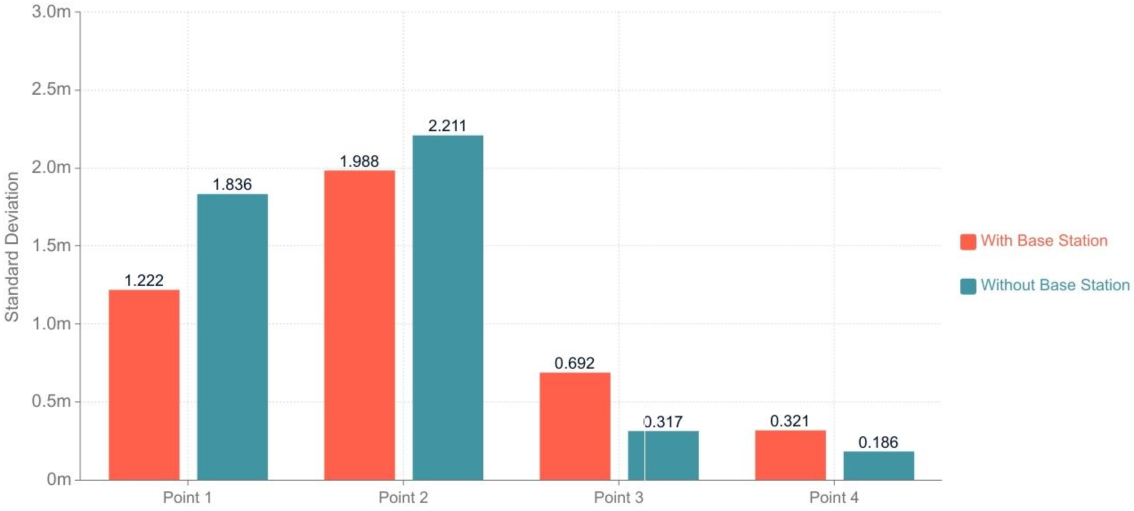

Figure 5.14: Standard Deviation of Longitudes (Obstructed)

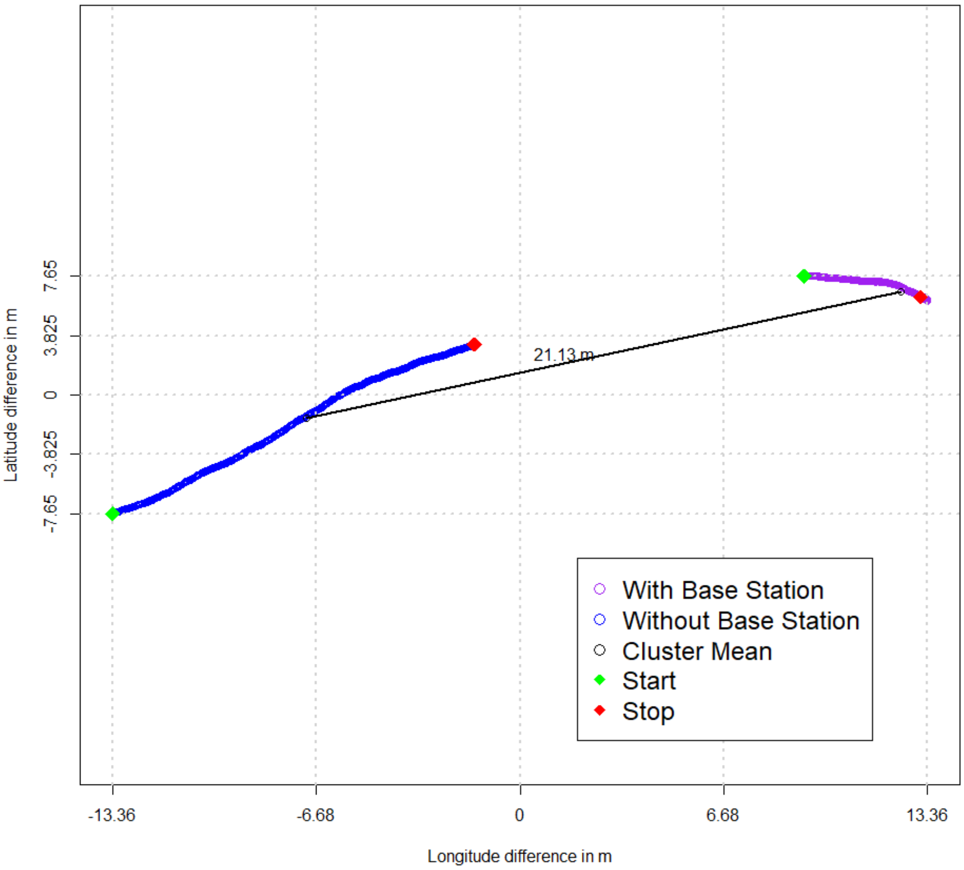

Figure 5.15: Data Points collected (Obstructed)

From figure 5.15, we can see there is a mean difference of 21.13 m between the points recorded with the base station and the points recorded without the base station. The system lacked accuracy and precision despite RTK corrections and cannot be used in places where the signals can be obstructed.

5.4 Dynamic Test Results

The figure 5.16 shows the results of the path travelled by the rover with and without the base station. The path was chosen under obstructed scenario near the buildings of Tallinn University of Technology. It also includes a section where the base station is not in line of sight with the GNSS receiver.

Figure 5.16 (a) displays the points plotted on the R-plot and Figure 5.16 (b) shows the pre-defined path travelled by the rover on the X-GIS map.

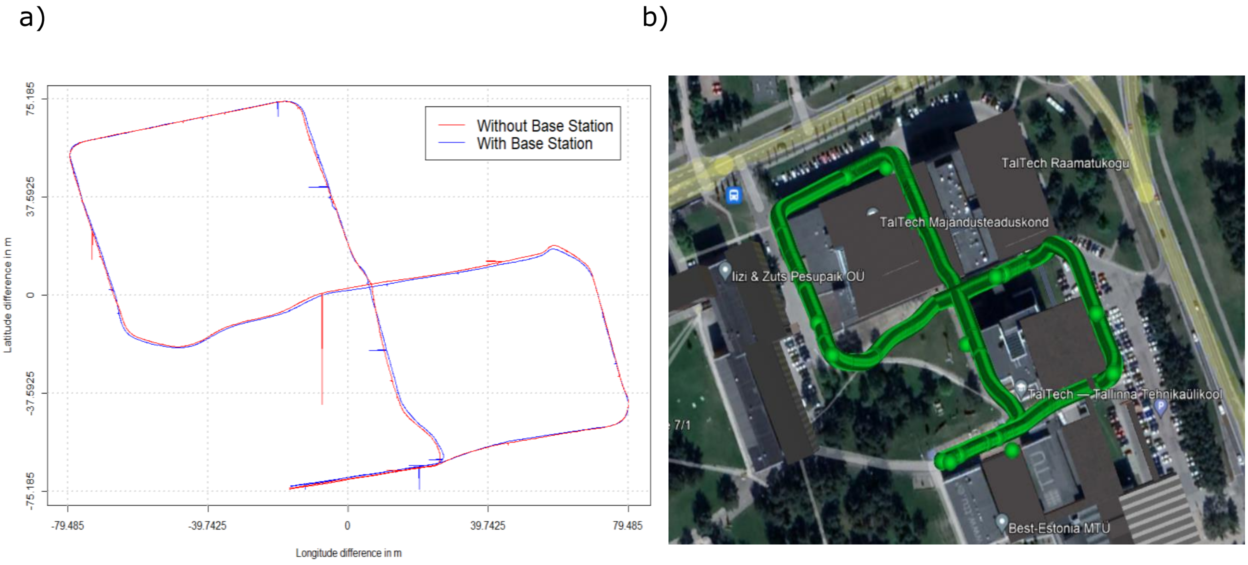

Figure 5.16: Dynamic movement of receiver a) on R-Plot b) on X-GIS Map

When the base station was switched on, the receiver almost followed the pre-defined path with certain errors at places where it was not in line-of-sight with the base station (shown by blue lines). In the absence of the base station, there was major deviation from the path which can be seen by the red spike in figure 5.16 (a). This indicates the receiver presented coordinates which were at a certain distance from the path. We can conclude that when the rover is in motion, the base station is able to give corrections in real time. Thus, this system can be used for many applications though a small number of outliers are present.

5.5 Nordwood Sawmill at Aegviidu

After determining the performance of the receiver under different environments, it was tested at the Nordwood Sawmill at Aegviidu. The idea is to use positioning information in locating the objects around the sawmill territory. In that sense, tests were taken at different points under different scenarios similar to the ones taken in Tallinn.

Figure 5.17 shows the layout of the sawmill territory. The different points where the measurements were taken are marked in numbers.

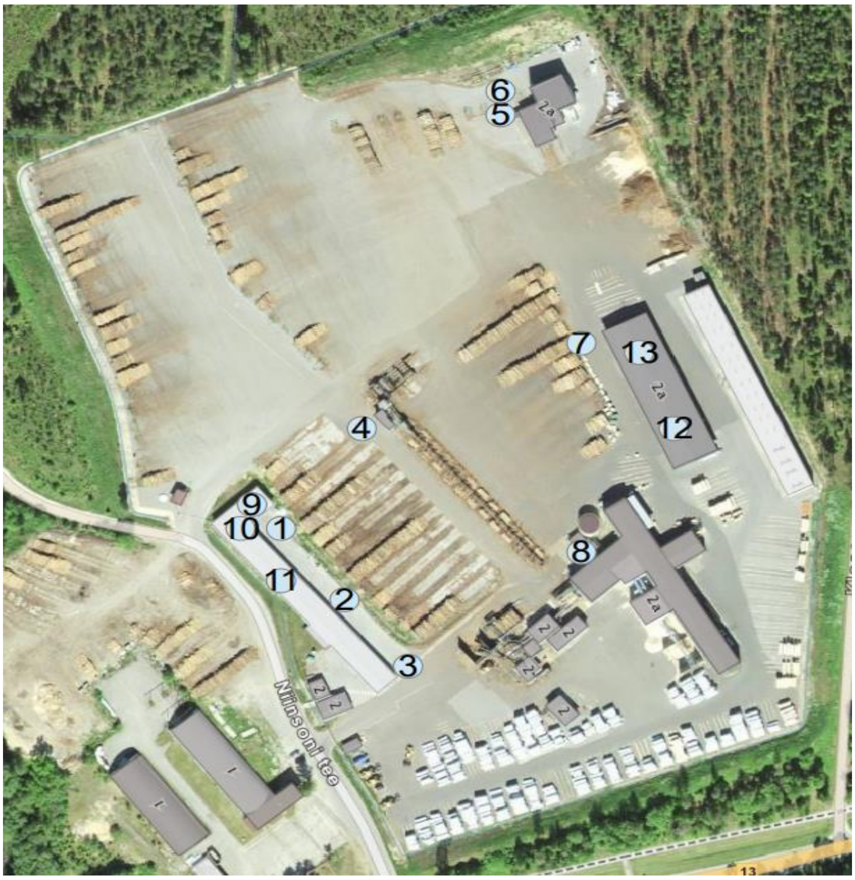

Figure 5.17: Layout of Nordwood sawmill territory

The results at Aegviidu indicate that the performance of the receiver under obstructed areas is worse as expected from the preliminary tests. The test points (9-13) for the obstructed scenarios were taken under the metal roofs where the cargo is stored.

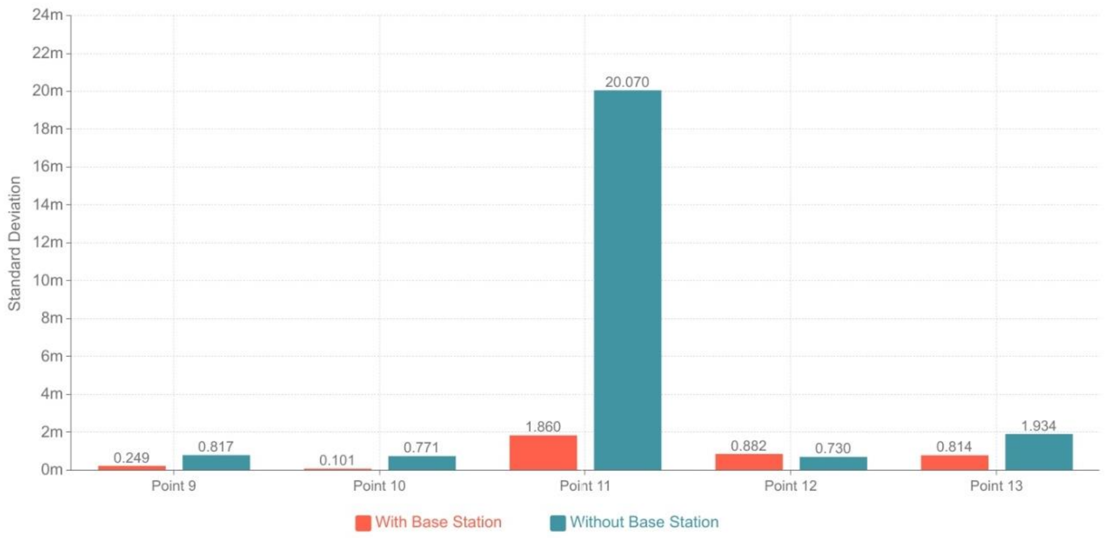

Figure 5.18: Standard Deviation of Latitudes at the sawmill

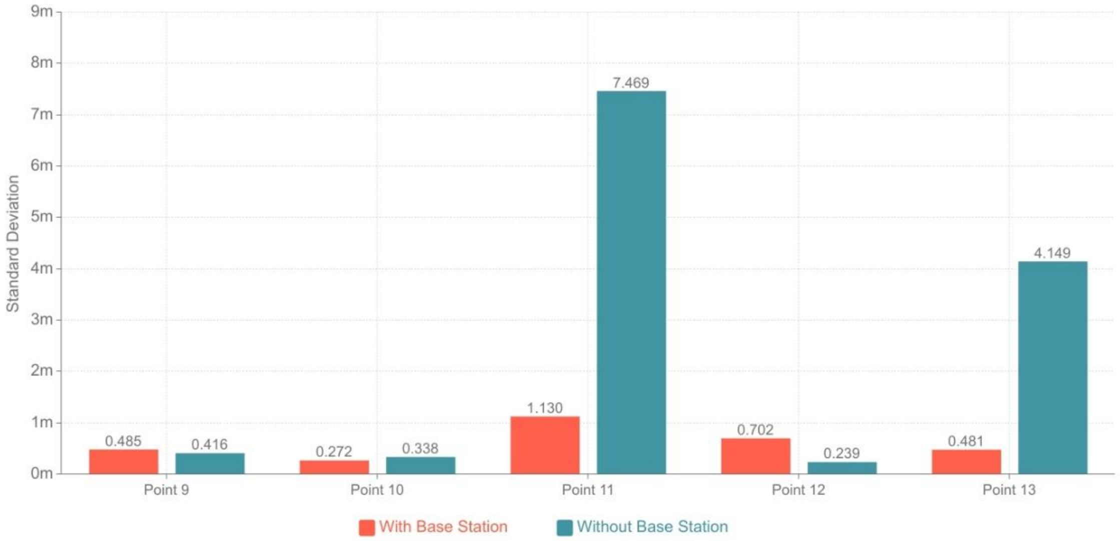

Figure 5.19: Standard Deviation of Longitudes at the sawmill

The measurements done both under high roof as well as low roof had a high amount of deviation. For example, point 11 (from figures 5.18 and 5.19) was taken under a small roof. The standard deviation value was about 7.4 m for longitude and 20 m for latitude. This makes it difficult for locating the position of objects in obstructed areas.

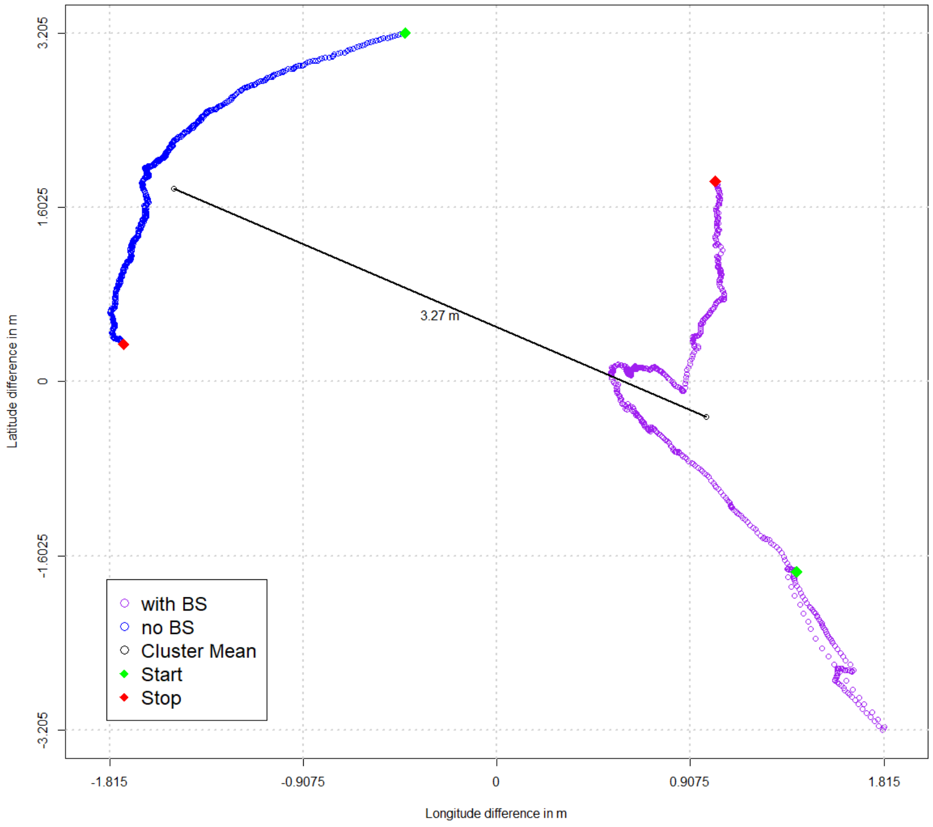

Figure 5.20: Data points collected at Aegviidu (Obstructed)

The figure 5.20 displays the data points collected under a low roof structure (obstructed). The graph shows a large deviation in the points taken with and without the base station, which indicates the precision is very low. When the cargo is present under obstructed areas, the receiver outputs coordinates with errors. The mean distance between both the points collected were found to be 3.27 m.

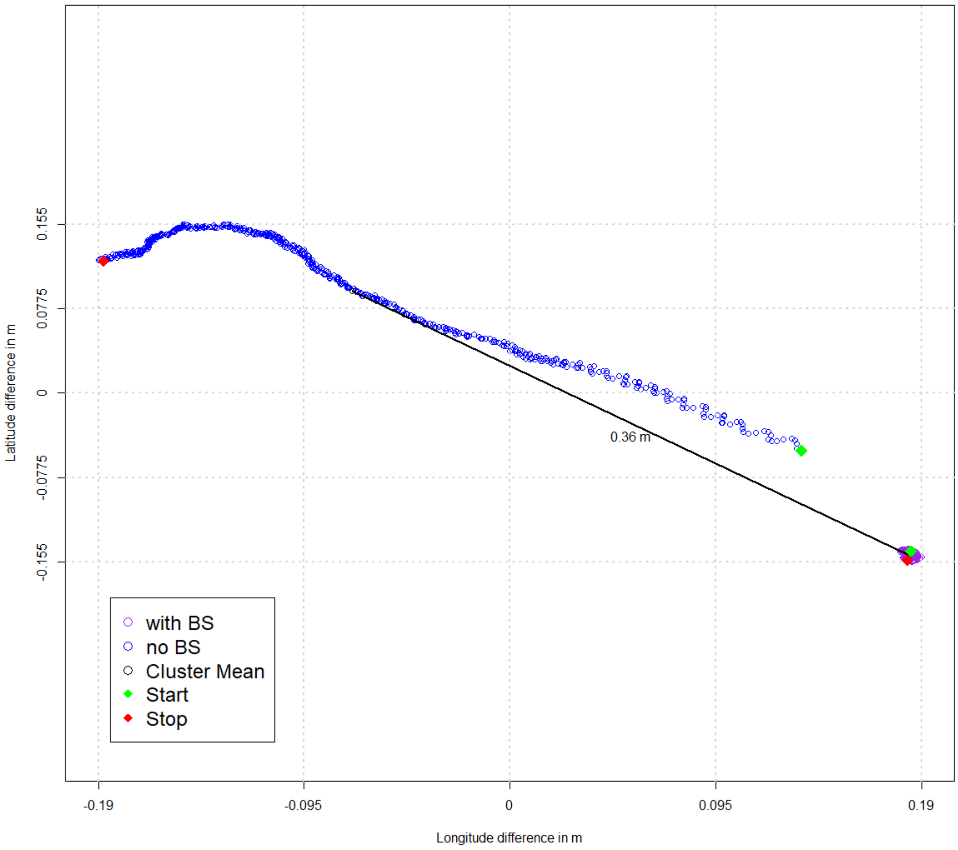

Figure 5.21: Data points collected at Aegviidu (Open Space)

The figure 5.21 displays the data points collected under the open space scenario. The precision was high with the Base station. Without the base station, the points are distributed over a wide area.

5.6 Indirect Tracking:

The application of indirect tracking is based on the concept of repeatability (as discussed in section 4.5). The measurements of the same points taken on both dates are compared with each other.

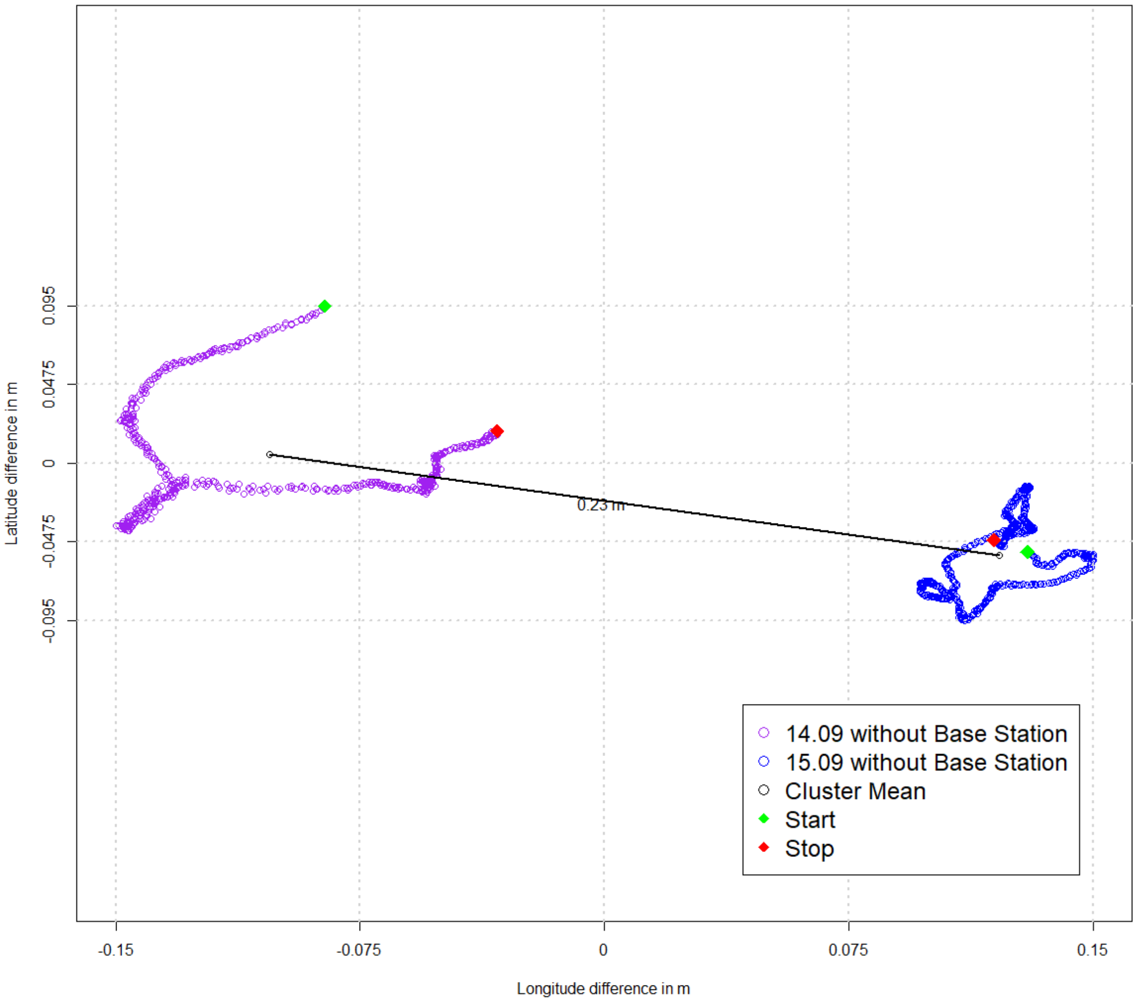

Figure 5.22: Repeatbility results (Open Space)

Under open space scenario, the rover is able to receive the signal accurately even without the use of base station. Figure 5.22 displays the collected points on both the dates. The deviation can be seen to be smaller, and the accuracy was around 6 cm.

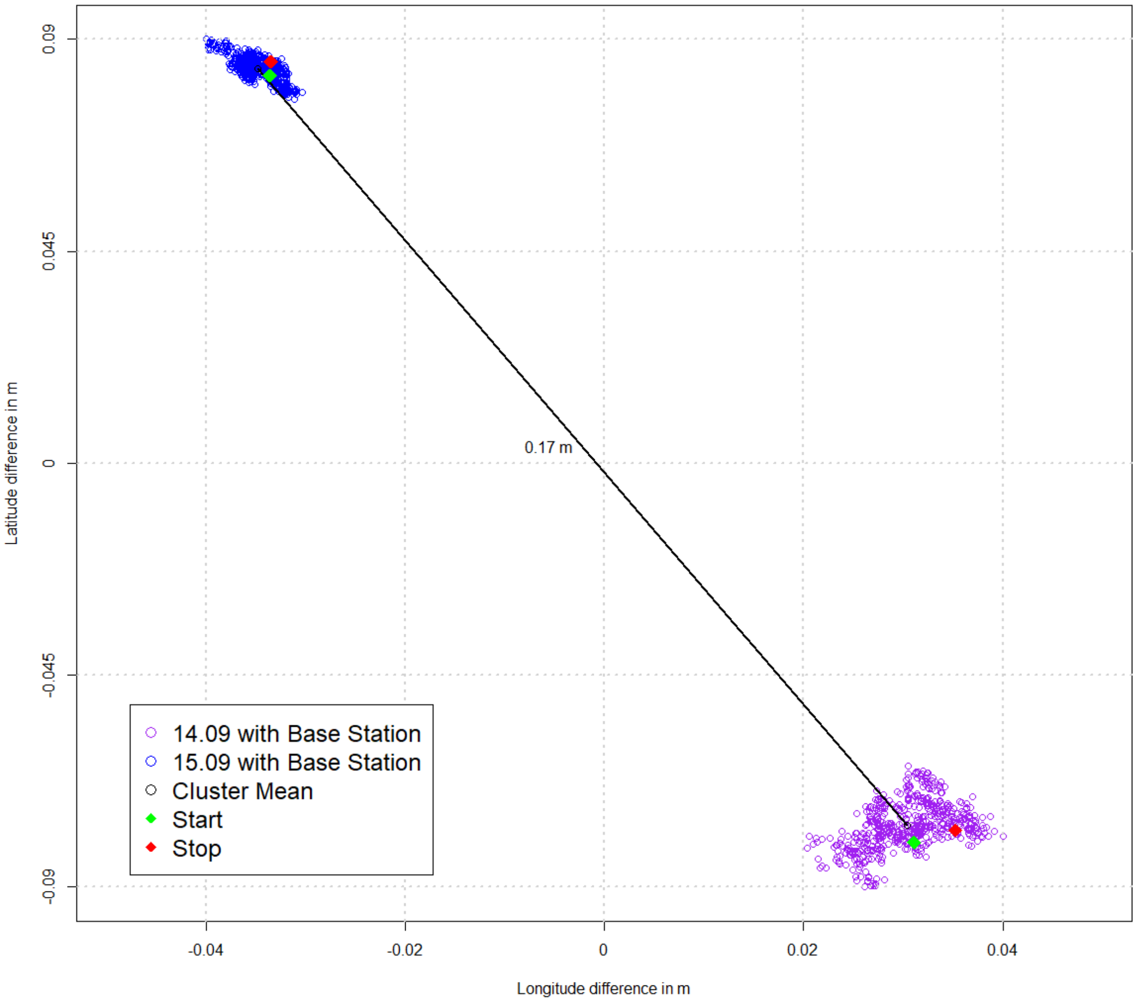

Figure 5.23: Repeatability results (Semi-Obstructed)

Figure 5.23 shows the collection of points under semi-obstructed taken on both the days. Since the base station is required to increase the accuracy under this scenario, the deviation value was found to be around 15 cm.

In both the scenarios the rover was able to output the same longitudinal position with a slight difference in the latitudes. These results conclude that the rover can be used for the purpose of indirectly tracking the objects with centimeter level accuracy under open space and semi-obstructed areas.

5.7 Summary of Results:

The results evaluated showed that the RTK technique provided more accurate results in semi-obstructed areas with the use of a base station. The base station was able to provide corrections to the receiver under semi-obstructed areas which increased the accuracy of the positional data. There was minimum benefit of the base station under open space area. There was no degradation in the satellite signals acquired by the receiver and base station in open space area. In the case of obstructed scenario, this method is not suitable as the standard deviation values were around 20 m. This makes the tracking of objects difficult in obstructed areas using the GNSS RTK technique. The dynamic tests showed the rover was able to output the path travelled without any errors when it was used with the base station. There were some anomalies present in the absence of base station. The results of repeatability showed the device was usable under open space and semi-obstructed area with the use of a base station. The accuracy of the results was within the acceptable range of 50 cm (as stated in section 4.5).

6. SUMMARY

GNSS is a satellite navigation system which determines the geographical location of a receiver on Earth with the help of satellite signals. Currently there are four major GNSS systems used in the world: GPS, GLONASS, BeiDou, Galileo. The topic of this Master's thesis was aimed at indirect tracking of cargo. For indirect tracking, the receivers need to be robust and accurate. Some GNSS receivers lack the positioning accuracy. To achieve high positioning accuracy using GNSS, currently two main techniques are being used: PPP and RTK. The author has used the RTK technique for this purpose as it is beneficial and gives correction in real-time. The GNSS RTK system consists of a rover which receives the satellite signals and a base station for sending corrections to the receiver.

The performance of the Fieldbee L2 GNSS RTK system used was evaluated by conducting tests under different scenarios such as open space, semi-obstructed and obstructed. Measurements were made with and without the base station to analyze the impact of RTK corrections sent to the receiver. In semi-obstructed scenario, the use of the base station improved the accuracy of positioning. There was minimal benefit of the base station in the open space and obstructed areas. The open space scenarios saw the receiver having good line of sight with the satellites and better signal strength in both receiver and base station. In the case of obstructed scenario, those conditions were violated and the Fieldbee system didn't provide acceptable accuracy. Similar tests were done by taking measurements in Nordwood Sawmill at Aegviidu where the concept of indirect tracking could be used for locating goods on the territory of the mill. The results evaluated under different scenarios confirmed that indirect tracking in obstructed areas is not possible and needs a different approach for the positioning of the cargo.

Future improvements may include identifying a solution to provide positioning information in closed or obstructed areas. One solution would be to use positioning solutions like Ultra-Wideband (UWB) or inertial measurement unit (IMU) sensors. A sensor fusion can be performed between these technologies and the GNSS systems to acquire positioning data in both outdoors and GNSS-denied areas. Future research could be done on using the constantly evolving SBAS method to provide information on positioning. However, the current literature stated that this method doesn't provide accurate results in semi-obstructed areas. There were problems regarding the evaluation of true coordinates in the thesis. Initially, the true coordinates were taken from the X-GIS map application. It can be concluded that due to the map's finite resolution, the absolute coordinates were found to be unreliable. An additional instrument like a precise geodetic device can be used to estimate the true coordinates of a location accurately. A front-end application can be created using this system wherein the user can store the location of different objects in a data base. The application would tell the user where each object is located on the map. This can be used in the industries or warehouses for keeping track of their goods.

6. KOKKUVÕTE

GNSS on satelliitnavigatsiooni süsteem, mille abil määratakse satelliitsignaali vastuvõtja asukoht maa peal. Hetkel on kasutusel 4 GNSS süsteemi: GPS, GLONASS, BeiDou ja Galileo. Käesoleva magistri töö teemaks oli objektide kaudne jälgimine GNSS abil. Selle all on mõeldud näiteks kaupa kandvate masinate asukoha määramist. Kauba jälgimise puhul sõltuks infrastruktuuri suurus ja hind kauba kogusest. Seevastu masinate jälgimise seadmepargi kulu sõltuks masinate arvust. Asukoha määramisel peavad GNSS vastuvõtjad olema ka robustsed ja täpsed. Kõrge positsioneerimistäpsuse saavutamiseks on kasutusel kaks peamist meetodit: PPP ja RTK. Autor kasutas RTK meetodit, kuna see võimaldab saada positsioneerimistäpsuse korrektsioone reaalajas. GNSS RTK süsteem koosneb vastuvõtjast ja baasjaamast, mille ülesanne on edastada vastuvõtjale positsioneerimise parandused.

Fieldbee L2 GNSS RTK süsteemi jõudlust hinnati välitestidega erinevate stsenaariumite korral nagu avatud, poolkinnine ja kinnine ala. Mõõtmised teostati nii baasjaamaga kui baasjaamata, et hinnata positsioneerimise korrektsioonide täpsust ja kasulikkust. Baasjaamast oli kõige rohkem kasu poolkinnisel alal, kuna parandas seal asukoha määramise täpsust. Avatud ja kinnisel alal oli baasjaamast vähe kasu. Esimesel juhul oli vastuvõtjal ja baasjaamal otsenähtavus satelliitidega ning seetõttu ka väga hea vastuvõetud signaali tugevus nii vastuvõtjas kui baasjaamas. Kinnisel alal need tingimused puudusid ning GNSS ei taganud piisavat asukoha määramise täpsust. Sarnaseid teste tehti ka Aegviidu Nordwood saeveskis, kus idee oli määrata puidukoormate asukohad. Tulemused erinevate stsenaariumite korral kinnitasid, et GNSS kasutamine kinnisel alal ei ole jätkusuutlik ning vajab teistsugust asukoha määramise lahendust.

Tuleviku lahenduste osas tuleks kaaluda meetodeid, mis tagaks positsioneerimise ka kinnistel aladel. Üks lahendus oleks näiteks rakendada ultra-lairiba või inertsiaalanduri põhist positsioneerimist ning liita need sensorid koos GNSS vastuvõtjaga ühtsesse süsteemi. Lisaks võiks tulevikus uurida pidevalt areneva SBAS süsteemi võimekust, sest hetke teaduskirjanduse järgi on see poolkinnistel aladel liiga ebatäpne. Käesoleva töö käigus esinesid probleemid absoluutsete koordinaatide määramisel. Esialgu määrati absoluutkoordinaadid X-GIS kaardirakenduse alusel kuid kaardikihi resolutsioonist sõltuvalt ei ole võimalik neid koordinaate absoluutse täpsusega määrata. Nende täpsemaks hindamiseks võiks tulevikus kasutada suurema täpsusega geodeetilist maamõõte seadet. Samuti võiks koostada rakenduse, mille abil kasutajad saaksid salvestada laos paiknevate kaupade asukohti, kasutades töös kajastatud kaudse positsioneerimise meetodit. Interaktiivne positsioneerimine säästaks aega vajaliku kauba ülesleidmisel nii välis- kui sisetingimustes.

REFERENCES

[1] T. Wang, Y. Zheng ja M. Lu, „On-The-Fly Ambiguity Resolution Based on Double-Differential Square Observation,‟ *Sensors,* nr 2495, p. 18, 2018.

[2] A. A. Bin Ariffin, N. H. Abdul Aziz ja K. Azura Othman, „Implementation of GPS for Location Tracking,‟ *IEEE Control and System Graduate Research Colloquium,* 2011.

[3] W. D. Dr. Neil ja V. Prof Dr. Schweiger, „Cost Effective GNSS Positioning Techniques,‟ International Federation of Surveyors (FIG), Helsinki, 2014.

[4] An Introduction to GNSS, Alberta, Canada: NovAtel Inc., 2015.

[5] T. E. Commission ja T. E. G. A. (GSA), „GNSS User Technology Report,‟ Publications Office of the European Union, Luxembourg, 2018.

[6] P. de Bakker ja C. Tiberius, „Real-time multi-GNSS single-frequency precise point positioning,‟ *GPS Solutions,* kd. 21, 2017.

[7] D. Kiliszek ja K. Kroszczynski, „Performance of the precise point positioning method along with the development of GPS, GLONASS and Galileo systems,‟ *Measurement,* 2020.

[8] M. Malinowski ja J. Kwiecien, „A Comparative study of Precise Point Positioning (PPP) Accuracy using Online Services,‟ *Reports on Geodesy and Geoinformatics,* kd. 102, pp. 15-31, 2016.

[9] J. Vazquez-Ontiveros, G. Vazquez Becerra ja J. e. a. Quintana, „Implementation of PPP-GNSS measurement technology in the probabilistic SHM of bridge structures,‟ *Measurement,* 2020.

[10] „PPP-RTK Market and Technology Report,‟ European GNSS Agency, 2019.

[11] R. Emardson, P. Jarlemark, J. Johansson, S. Bergstrand, M. Lidberg ja B. Jonsson, „Measurement Accuracy in Network-RTK,‟ *Bollettino Geod. Sci. Affini.*

[12] W. Stempfhuber ja M. Buchholz, „A precise, low-cost RTK GNSS system for UAV applications‟.

[13] M. Garrido-Carretero, M. Cobos, M. Borque-Arancón, A. M. Ruiz-Armenteros, R. Moreno-Guerrero ja A. Gil, „Low-cost GNSS reciever in RTK positioning under the standard ISO-17123-8: A feasible option in geomatics,‟ *Measurement,* kd. 137, pp. 168-178.

[14] D. Valente, M. Momin, T. Grift ja A. Hansen, „Accuracy and precision evaluation of two low-cost RTK global navigation satellite systems,‟ *Computer and Electronics in Agriculture,* kd. 168, 2019.

[15] N. V. Nguyen, W. Cho ja K. Hayashi, „Performance evaluation of a typical low-cost multi-frequency multi-GNSS device for positioning and navigation in agriculture –Part 1: Static testing,“ *Smart Agricultural Technology,* kd. I, 2021.

[16] „Emlid - Placing the base,“ Innovelec, 24 09 2020. [Võrgumaterjal]. Available: https://innovelec.co.uk/emlid-placing-the-base/. [Kasutatud 11 11 2021].

[17] T. Walter, Position, Navigation and TIming Technologies in the 21st Century, 2017.

[18] T. Dautermann, „Civil air navigation using GNSS enhanced by wide area satellite based,“ *Progress in Aerospace Sciences,* pp. 51-62, 2014.

[19] G. Grunwald, M. Bakula, A. Ciecko ja R. Kaźmierczak, „Study of EGNOS accuracy and integrity in eastern Poland,“ *The Aeronautical Journal,* kd. 120, pp. 1275-1290, 2016.

[20] B. Andrzej, R. Bober, T. Szewczuk ja A. Wolski, „Monitoring of Cargo Transport in Urban Areas Using GPS/EGNOS Technologies as Part of the Safety System,“ *Procedia - Social and Behavioral Sciences,* kd. 151, pp. 6-15, 2014.

[21] X. Luo, S. Schaufler, M. Branzanti ja J. Chen, „Assessing the benefits of Galileo to high-precision GNSS positioning - RTK, PPP and post-processing,“ *Advances in Space Research,* 2020.

[22] B. Bahadur ja M. Nohutcu, „Integration of variance component estimation with robust Kalman filter for single-frequency multi-GNSS positioning,“ *Measurement,* 2020.

[23] N. Nadarajah, A. Khodabandeh, K. Wang, M. Choudhury ja P. Teunissen, „Multi-GNSS PPP-RTK: From large-to Small-Scale networks,“ *Sensors (Basel, Switzerland),* kd. 18, 2018.

[24] D. Odijik, B. Zhang ja T. J. Peter, „Multi-GNSS PPP and PPP-RTK: Some GPS+BDS Results in Australia,“ kd. 341, pp. 613-625, 2015.

[25] Y. Zhang, N. Kubo, J. Chen, J. Wang ja H. Wang, „Initial Positioning Assessment of BDS New Satellites and New Signals,“ *Remote Sensing,* kd. 11, p. 1320, 2019.

[26] Q. Li, R. Li, K. Ji ja W. Dai, „Kalman Filter and Its application,“ *2015 8th International Conference on Intelligent Networks and Intelligent Systems,* pp. 74-77, 2015.

[27] D. Wallach, D. Makowski, J. W. Jones ja F. Brun, The R Programming Language and Software. Working with Dynamic Crop Models, Academic Press, 2019.

[28] D. Peterson, „Distances on Earth 1:The Cosine Formula,“ *Formulas, Real life,* March 2021.

[29] P. Weber, „Measurement Error,“ *The International Encyclopedia of Communication Research Methods,* pp. 1-2, 2017.

[30] L. Blank ja A. Tarquin, Engineering Economy, McGraw-Hill, 2012.

[31] P. N. Simon ja H. M. Reza, Ocean Modelling for Resource Characterization. Fundamental of Ocean Renewable Energy, Academic Press, 2018.

[32] H. R. Hatta, M. H. Suroso, I. F. Astuti, D. M. Khairina ja S. Maharani, „Application of Haversine Formula in Education Game "Landmark Nusantara",“ *Advances in Engineering Research,* kd. 203, 2020.

[33] M. L. Waskom, „Seaborn: Statistical Data Visualization,“ *The Journal of Open Source Software,* 2021.

7. APPENDICES

Appendix 1 – Python script for collecting data from receiver

```
import os
import serial
from time import time

from datetime import datetime

path_to_script = os.path.dirname(__file__)

def receiver_coordinates():

    test_no_counter = 1  # Keeps track of number of tests conducted at a specific spot
    start_time = time()
    seconds = 60

    incoming_msg = 0


    while True:  # Ensure that input for location_no is an integer
        try:
            location_no = input("Enter the number of the spot currently being tested: ")
            location_no = int(location_no)
            break
        except ValueError:
            print("Please enter an integer value!")

    files_in_folder = os.listdir(f"{path_to_script}/Collected Data/")  # Checks the file in
the specified folder

    # Number of tests conducted
    for file_name in files_in_folder:

        if "_" in file_name:
            splitted_file_name = file_name.split("_")
            if splitted_file_name[1] == str(location_no):
                if test_no_counter <= int(splitted_file_name[2]):
                    test_no_counter += 1


# Serial connection with the GNSS receiver

    gnss = serial.Serial(port="COM3", baudrate=115200, bytesize=8, timeout=2,
stopbits=serial.STOPBITS_ONE)  # Serial connection with the receiver
    serialString = ""
    now = datetime.now().strftime("Date_%m_%d_%Y_Time_%H_%M_%S")


    while True:
        line = gnss.readline().decode("utf-8").strip("\r\n")
        data = line.split(',')
        current_time = time()
        elapsed_time = current_time - start_time

        if data[0] == "$GPGGA":

            with open(f"{path_to_script}/Collected
Data/{now}_Location_{str(location_no)}_{str(test_no_counter)}.txt",
                      "a") as f:  # Create the file name based on user input and calculated
test iteration

                f.write(f"{data[2]} \t {data[4]} \n")

                if elapsed_time > seconds:
                    break

receiver_coordinates()
```

Appendix 2 – Python script for calculation of data

```
from math import cos,sin,asin,sqrt
from scipy.stats import skew
from scipy.stats import kurtosis

from statistical_measures import *
import statistics
import seaborn as sns
import matplotlib.pyplot as plt

R = 6371

# Specify the input file path
collected_coordinates = open("")
lines = collected_coordinates.readlines()
coordinates = lines[-1].split("\t")

temp_out = [el.strip("\t\n") for el in lines]
list_of_coords = [el.split("\t") for el in temp_out]

x_list, y_list = [], []
sys_x_deg_list, sys_y_deg_list = [],[]
sys_x_rad_list, sys_y_rad_list = [] , []
sys_x_list, sys_y_list = [] , []
error_x_list, error_y_list = [],[]
dist_x_list,dist_y_list = [],[]

for element in list_of_coords:
    x_list.append(float(element[0].strip()))
    y_list.append(float(element[1].strip()))


for el in x_list:
    x_receiver = el/100

    # Conversion to Degrees Latitude
    split_x_receiver = str(x_receiver).split('.')
    x_receiver_degree = int(split_x_receiver[0])
    x_receiver_minutes = int(split_x_receiver[1]) // 60

    sys_x_deg = (float(f"{x_receiver_degree}.{x_receiver_minutes}"))
    sys_x_deg_list.append(sys_x_deg)

    sys_x_rad = sys_x_deg * math.pi / 180  # Converting degrees to radians
    sys_x_rad_list.append(sys_x_rad)

avg_sys_x_rad = statistics.mean(sys_x_rad_list)

for el in y_list:
    y_receiver = el/100

    # Conversion to Degrees Longitude
    split_y_receiver = str(y_receiver).split('.')
    y_receiver_degree = int(split_y_receiver[0])
    y_receiver_minutes = int(split_y_receiver[1]) // 60
    sys_y_deg = (float(f"{y_receiver_degree}.{y_receiver_minutes}"))
    sys_y_deg_list.append(sys_y_deg)

    sys_y_rad = sys_y_deg * math.pi / 180  # Converting degrees to radians
    sys_y_rad_list.append(sys_y_rad)

avg_sys_y_rad = statistics.mean(sys_y_rad_list)

"""
Function to convert the coordinate to x,y in metres
sys_x = R * cos(lat) * cos(lon) * 1000
sys_y = R * cos(lat) * sin(lon) * 1000

"""

for el in range(len(sys_x_rad_list)):

    # Values for calculation
    sys_x = R * cos(sys_x_rad_list[el]) * cos(sys_y_rad_list[el]) * 1000
    sys_y = R * cos(sys_x_rad_list[el]) * sin(sys_y_rad_list[el]) * 1000

    sys_x_list.append(sys_x)
    sys_y_list.append(sys_y)

avg_sys_x = statistics.mean(sys_x_list)
avg_sys_y = statistics.mean(sys_y_list)

# Creation of Distribution list
for el in range(len(sys_x_list)):
    dist_x = sys_x_list[el] - avg_sys_x # Each Value - Mean of the list of latitudes
    dist_x_list.append(dist_x)

for el in range(len(sys_y_list)):
    dist_y = sys_y_list[el] - avg_sys_y # Each Value - Mean of the list of longitudes
    dist_y_list.append(dist_y)

# Input Coordinates
true_x_deg = float(input("Enter the latitude in degrees:"))
true_y_deg = float(input("Enter the longitude in degrees:"))

true_x_rad = true_x_deg * math.pi/180
true_y_rad = true_y_deg * math.pi/180

true_x = R * math.cos(true_x_rad) * math.cos(true_y_rad) * 1000
true_y = R * math.cos(true_x_rad) * math.sin(true_y_rad) * 1000

# Creation of Error List
for el in sys_x_list:
    error_x_list.append(el - true_x)

for el in sys_y_list:
    error_y_list.append(el - true_y)

# Distance
# Haversine Formula
dlon = abs(avg_sys_y_rad - true_y_rad)
dlat = abs(avg_sys_x_rad - true_x_rad)
```

```
a = sin(dlat / 2) ** 2 + cos(true_x_rad) * cos(avg_sys_x_rad) * sin(dlon / 2) ** 2
c = 2 * asin(sqrt(a))
dist = c*R*1000
distance = format(dist,".4f")
print("Distance:",distance,"m")

# Accuracy
accu_x = abs(true_x - avg_sys_x)
accu_y = abs(true_y - avg_sys_y)
accuracy_x = format(accu_x,".5f")
accuracy_y = format(accu_y,".5f")
print("Accuracy of Latitude:",accuracy_x,"m")
print("Accuracy of Longitude:",accuracy_y,"m")

# Standard Deviation
std_x = calc_STDEV(sys_x_list)
std_y = calc_STDEV(sys_y_list)
stdev_x = format(std_x,".5f")
stdev_y = format(std_y,".5f")
print("Standard Deviation of Latitude:",stdev_x,"m")
print("Standard Deviation of Longitude:",stdev_y,"m")

# Average Mean
avg_x = calc_AVG(sys_x_deg_list)
avg_y = calc_AVG(sys_y_deg_list)
average_x = format(avg_x,".3f")
average_y = format(avg_y,".3f")
print("Mean of Latitude:",average_x,"°")
print("Mean of Longitude:",average_y,"°")

#Median
med_x = statistics.median(sys_x_deg_list)
med_y = statistics.median(sys_y_deg_list)
median_x = format(med_x,".3f")
median_y = format(med_y,".3f")
print("Median of Latitude:",median_x,"°")
print("Median of Longitude:",median_y,"°")

# Skewness
skew_x = skew(sys_x_list, bias= False)
skew_y = skew(sys_y_list, bias= False)
skewness_x = format(skew_x,".5f")
skewness_y = format(skew_y,".5f")
print("Skewness of Latitudes:",skewness_x,)
print("Skewness of Longitudes:",skewness_y)

# Kurtosis
kurt_x = kurtosis(sys_x_list, bias= False)
kurt_y = kurtosis(sys_y_list, bias= False)
kurtosis_x = format(kurt_x,".5f")
kurtosis_y = format(kurt_y,".5f")
print("Kurtosis of Latitudes:",kurtosis_x)
print("Kurtosis of Longitudes:",kurtosis_y)

#RMSE
root_x = calc_RMSE(error_x_list)
rmse_x = format(root_x,".5f")
```

```
root_y = calc_RMSE(error_y_list)
rmse_y  = format(root_y,".5f")
print("RMSE  of Latitudes:",rmse_x,"m")
print("RMSE  of Longitudes:",rmse_y,"m")

# Plotting the Distribtution

parameters = {'xtick.labelsize':  14, 'ytick.labelsize':  14,
          'axes.labelsize':  14,}
plt.rcParams.update(parameters)

fig, axes = plt.subplots(1, 2, figsize=(17,  7), sharey=False,  dpi=100)
p = sns.distplot(dist_x_list, color="green", ax=axes[0], axlabel='Latitudes in
metres',hist=False)
q = sns.distplot(dist_y_list, color="red", ax=axes[1], axlabel='Longitudes in
metres',hist=False)
axes[0].set_xlim(auto=True)
axes[1].set_xlim(auto=True)
p.set_ylabel("Density  %")
q.set_ylabel("Density  %")
plt.show()
```

Appendix 3 – R Script

```
rm(list=ls())
graphics.off()

#Set the directory where data is located
setwd("D:/Master Thesis/Trial2/Tallinn/Collected Data")

#|--------------------------------------------------------------------------------|

# Semi-obstructed conv. test (w Basestation)
#my_data1                                                                       <-
read.delim("Date_10_21_2021_Time_10_48_48_Location_1_1.txt",header=F)
#my_data2                                                                       <-
read.delim("Date_09_14_2021_Time_10_45_05_Location_1_1.txt",header=F)

#|--------------------------------------------------------------------------------|

# Open space pt1 (w Basestation)
#my_data1                                                                       <-
read.delim("Date_09_14_2021_Time_11_16_34_Location_7_1.txt",header=F)
#my_data2                                                                       <-
read.delim("Date_09_15_2021_Time_15_22_01_Location_27_1.txt",header=F)
#TC <- matrix(c(59.431479,24.705915,3),ncol=3)

# Open space pt2 (w Basestation)
#my_data1                                                                       <-
read.delim("Date_09_14_2021_Time_11_18_37_Location_8_1.txt",header=F)
#my_data2                                                                       <-
read.delim("Date_09_15_2021_Time_15_24_16_Location_28_1.txt",header=F)
#TC <- matrix(c(59.43205,24.705995,3),ncol=3)

# Open space pt3 (w Basestation)
my_data1                                                                        <-
read.delim("Date_09_14_2021_Time_11_21_04_Location_9_1.txt",header=F)
#my_data2                                                                       <-
read.delim("Date_09_15_2021_Time_15_26_29_Location_29_1.txt",header=F)
TC <- matrix(c(59.431861,24.704637,3),ncol=3)

#|--------------------------------------------------------------------------------|
#|--------------------------------------------------------------------------------|

# Open space pt1 (no Basestation)
#my_data1                                                                       <-
read.delim("Date_09_14_2021_Time_11_23_17_Location_10_1.txt",header=F)
#my_data2                                                                       <-
read.delim("Date_09_15_2021_Time_15_28_42_Location_30_1.txt",header=F)
#TC <- matrix(c(59.431479,24.705915,3),ncol=3)

# Open space pt2 (no Basestation)
#my_data1                                                                       <-
read.delim("Date_09_14_2021_Time_11_25_30_Location_11_1.txt",header=F)
#my_data2                                                                       <-
read.delim("Date_09_15_2021_Time_15_30_57_Location_31_1.txt",header=F)
#TC <- matrix(c(59.43205,24.705995,3),ncol=3)

# Open space pt3 (no Basestation)
```

```
#my_data1                                                          <-
read.delim("Date_09_14_2021_Time_11_27_29_Location_12_1.txt",header=F)
my_data2                                                           <-
read.delim("Date_09_15_2021_Time_15_33_04_Location_32_1.txt",header=F)
#TC <- matrix(c(59.431861,24.704637,3),ncol=3)

#|----------------------------------------------------------------------------|
#|----------------------------------------------------------------------------|

# Semi obstructed pt1 (w Basestation)
#my_data1                                                          <-
read.delim("Date_09_14_2021_Time_10_45_05_Location_1_1.txt",header=F)
#my_data2                                                          <-
read.delim("Date_09_15_2021_Time_14_51_53_Location_21_1.txt",header=F)
#TC <- matrix(c(59.411583,24.678773,3),ncol=3)

# Semi obstructed pt2 (w Basestation)
#my_data1                                                          <-
read.delim("Date_09_14_2021_Time_10_46_55_Location_2_1.txt",header=F)
#my_data2                                                          <-
read.delim("Date_09_15_2021_Time_14_53_32_Location_22_1.txt",header=F)
#TC <- matrix(c(59.411585,24.679227,3),ncol=3)

# Semi obstructed pt3 (w Basestation)
#my_data1                                                          <-
read.delim("Date_09_14_2021_Time_10_49_03_Location_3_1.txt",header=F)
#my_data2                                                          <-
read.delim("Date_09_15_2021_Time_14_55_14_Location_23_1.txt",header=F)
#TC <- matrix(c(59.411913,24.679281,3),ncol=3)

#|----------------------------------------------------------------------------|
#|----------------------------------------------------------------------------|

# Semi obstructed pt1 (no Basestation)
#my_data1                                                          <-
read.delim("Date_09_14_2021_Time_10_51_27_Location_4_1.txt",header=F)
#my_data2                                                          <-
read.delim("Date_09_15_2021_Time_14_57_22_Location_24_1.txt",header=F)
#TC <- matrix(c(59.411583,24.678773,3),ncol=3)

# Semi obstructed pt2 (no Basestation)
#my_data1                                                          <-
read.delim("Date_09_14_2021_Time_10_53_02_Location_5_1.txt",header=F)
#my_data2                                                          <-
read.delim("Date_09_15_2021_Time_14_59_31_Location_25_1.txt",header=F)
#TC <- matrix(c(59.411585,24.679227,3),ncol=3)

# Semi obstructed pt3 (no Basestation)
#my_data1                                                          <-
read.delim("Date_09_14_2021_Time_10_54_37_Location_6_1.txt",header=F)
#my_data2                                                          <-
read.delim("Date_09_15_2021_Time_15_01_08_Location_26_1.txt",header=F)
#TC <- matrix(c(59.411913,24.679281,3),ncol=3)

#|----------------------------------------------------------------------------|
#|----------------------------------------------------------------------------|
# Obstructed pt1 (w Basestation)
#my_data1                                                              <-
read.delim("Date_09_14_2021_Time_11_52_28_Location_13_1.txt",header=F)
#my_data2                                                              <-
read.delim("Date_09_15_2021_Time_15_55_51_Location_33_1.txt",header=F)
#TC <- matrix(c(59.393963,24.669711,3),ncol=3)

# Obstructed pt2 (w Basestation)
#my_data1                                                              <-
read.delim("Date_09_14_2021_Time_11_54_44_Location_14_1.txt",header=F)
#my_data2                                                              <-
read.delim("Date_09_15_2021_Time_15_58_23_Location_34_1.txt",header=F)
#TC <- matrix(c(59.394579,24.669447,3),ncol=3)

# Obstructed pt3 (w Basestation)
#my_data1                                                              <-
read.delim("Date_09_14_2021_Time_11_56_42_Location_15_1.txt",header=F)
#my_data2                                                              <-
read.delim("Date_09_15_2021_Time_16_00_08_Location_35_1.txt",header=F)
#TC <- matrix(c(59.39461,24.669969,3),ncol=3)

# Obstructed pt4 (w Basestation)
#my_data1                                                              <-
read.delim("Date_09_14_2021_Time_11_59_02_Location_16_1.txt",header=F)
#my_data2                                                              <-
read.delim("Date_09_15_2021_Time_16_02_17_Location_36_1.txt",header=F)
#TC <- matrix(c(59.393967,24.67019,3),ncol=3)

#|--------------------------------------------------------------------------|
#|--------------------------------------------------------------------------|

# Obstructed pt1 (no Basestation)
#my_data1                                                              <-
read.delim("Date_09_14_2021_Time_12_00_45_Location_17_1.txt",header=F)
#my_data2                                                              <-
read.delim("Date_09_15_2021_Time_16_03_48_Location_37_1.txt",header=F)
#TC <- matrix(c(59.393963,24.669711,3),ncol=3)

# Obstructed pt2 (no Basestation)
#my_data1                                                              <-
read.delim("Date_09_14_2021_Time_12_03_02_Location_18_1.txt",header=F)
#my_data2                                                              <-
read.delim("Date_09_15_2021_Time_16_06_02_Location_38_1.txt",header=F)
#TC <- matrix(c(59.394579,24.669447,3),ncol=3)

# Obstructed pt3 (no Basestation)
#my_data1                                                              <-
read.delim("Date_09_14_2021_Time_12_06_50_Location_19_1.txt",header=F)
#my_data2                                                              <-
read.delim("Date_09_15_2021_Time_16_08_11_Location_39_1.txt",header=F)
#TC <- matrix(c(59.39461,24.669969,3),ncol=3)

# Obstructed pt4 (no Basestation)
#my_data1                                                              <-
read.delim("Date_09_14_2021_Time_12_09_05_Location_20_1.txt",header=F)
#my_data2                                                              <-
read.delim("Date_09_15_2021_Time_16_10_32_Location_40_1.txt",header=F)
```

```
#TC <- matrix(c(59.393967,24.67019,3),ncol=3)

#|--------------------------------------------------------------------------------|
#|--------------------------------------------------------------------------------|

options(digits=8)

# Switch 1st and 2nd column
my_data1 <- my_data1[,c("V2","V1")]
my_data2 <- my_data2[,c("V2","V1")]
TC <- TC[,c(2,1,3)]

# Add labels to input data
L <- matrix(1,nrow(my_data1))
my_data1 <- cbind(my_data1,L)
L <- matrix(2,nrow(my_data2))
my_data2 <- cbind(my_data2,L)

# Combine the data sets
my_data <- rbind(my_data1,my_data2)

# Earths radius
R <- 6371*10^3

# Coordinate conversion
Conv <- function(input_data){
  # Data conversion for Latitudes
  Lat_deg <- as.integer(input_data[,1] / 100)
  Lat_mm <- input_data[,1] %% 100
  Lat_conv <- Lat_deg + (Lat_mm / 60)

  # Data conversion for Longitudes
  Lon_deg <- as.integer(input_data[,2] / 100)
  Lon_mm <- input_data[,2] %% 100
  Lon_conv <- Lon_deg + (Lon_mm / 60)

  A <- cbind(Lat_conv,Lon_conv,my_data[,3])

  return(A)
}

# Haversine distance
H_distance <- function(a,b) {
  phi1 <- a[2]*(pi/180)
  phi2 <- b[2]*(pi/180)
  dphi <- (b[2]-a[2])*(pi/180)
  dlam <- (b[1]-a[1])*(pi/180)

  g <- (sin(dphi/2)^2)+cos(phi1)*cos(phi2)*(sin(dlam/2)^2)
  c <- 2*atan2(sqrt(g),sqrt(1-g))
  d <- c*R
  return(d)
}

# Convert input data
C <- Conv(my_data)
```

```
# Calculate Mean for both clusters
MC <- matrix(0,nrow=2,ncol=3)
for (i in 1:2){
  cluster <- C[C[,3] == i, 1:2]
  MC[i,1] <- mean(cluster[,1])
  MC[i,2] <- mean(cluster[,2])
  MC[i,3] <- 4
}

# Haversine distance between means
MeanD <- round(H_distance(MC[1,],MC[2,]),2)

# Haversine distance between 14.09 mean and True Coord.
Dist1 <- round(H_distance(MC[1,],TC),2)

# Haversine distance between 14.09 mean and True Coord.
Dist2 <- round(H_distance(MC[2,],TC),2)

# Separate clusters start/stop pt
c1 <- C[C[,3] == 1, 1:2]
c1_1 <- c1[1,1:2]
c1_2 <- c1[length(c1[,1]),1:2]

c2 <- C[C[,3] == 2, 1:2]
c2_1 <- c2[1,1:2]
c2_2 <- c2[length(c2[,1]),1:2]

# Bind True coord. and two clusters and their means
C_TC <- rbind(C,MC,TC)

# X and Y axis conversion
w <- C_TC[,1:2]

Xmin <- min(w[,1])
Xmax <- max(w[,1])
Ymin <- min(w[,2])
Ymax <- max(w[,2])

XL <- cbind(Xmin,Ymin)
XU <- cbind(Xmax,Ymin)
YU <- cbind(Xmin,Ymax)

Xlength <- round(H_distance(XL,XU),2)
Ylength <- round(H_distance(XL,YU),2)

nrticks <- 5
xseq <- seq(Xmin,Xmax,length.out=nrticks)
yseq <- seq(Ymin,Ymax,length.out=nrticks)

# Master Plot
plot(C_TC,col=c("purple","blue","black","black")[C_TC[,3]],lwd=1,xaxt="n",yaxt="n",
xlab="Longitude difference in m",ylab="Latitude difference in m",cex=1,pch=1,asp=1)

axis(1,at=xseq,labels=c((-1)*Xlength/2,(-1)*Xlength/4,0,Xlength/4,Xlength/2))
axis(2,at=yseq,labels=c((-1)*Ylength/2,(-1)*Ylength/4,0,Ylength/4,Ylength/2))

grid(lwd = 2, nx = NA, ny = NA)
```

```
abline(v = xseq, lwd = 2, lty = 3, col = "lightgray")
abline(h = yseq, lwd = 2, lty = 3, col = "lightgray")

# Plot lines
segments(x0 = MC[1,1], y0 = MC[1,2], x1 = MC[2,1], y1 = MC[2,2], col =
"black",lwd=2)
segments(x0 = TC[1], y0 = TC[2], x1 = MC[1,1], y1 = MC[1,2], col = "black",lwd=2)
segments(x0 = TC[1], y0 = TC[2], x1 = MC[2,1], y1 = MC[2,2], col = "black",lwd=2)

# Plot text
text(x = (MC[1,1]+MC[2,1])/2,  y = (MC[1,2]+MC[2,2])/2,sprintf("%s",MeanD))
text(x = (TC[1]+MC[1,1])/2,  y = (TC[2]+MC[1,2])/2,sprintf("%s",Dist1))
text(x = (TC[1]+MC[2,1])/2,  y = (TC[2]+MC[2,2])/2,sprintf("%s",Dist2))
text(x = TC[1], y = TC[2],sprintf("%s",TC[1]),pos=3)
text(x = TC[1], y = TC[2],sprintf("%s",TC[2]),pos=1)

# True coordinate plot
points(TC[1],TC[2],col="black",cex=1.5,pch=16)

# Start/stop points
points(c1_1[1],c1_1[2],col="green",cex=2,pch=18)
points(c1_2[1],c1_2[2],col="red",cex=2,pch=18)

points(c2_1[1],c2_1[2],col="green",cex=2,pch=18)
points(c2_2[1],c2_2[2],col="red",cex=2,pch=18)

# Legend of all items
legend("bottomright",    # Position
      inset = 0.05, # Distance from the margin  as a fraction of the plot region
      legend = c("14.09.2021","15.09.2021","True Coord.","Cluster
Mean","Start","Stop"),
      col = c("purple",  "blue","black","black","green","red"),
      pch = c(1,1,16,1,18,18),cex=1.5)

# Standard deviation conversion and calculation
rad_c1_lon <- c1[,1]*pi/180
rad_c1_lat <- c1[,2]*pi/180

rad_c2_lon <- c2[,1]*pi/180
rad_c2_lat <- c2[,2]*pi/180

c1y <- R * cos(rad_c1_lat) * cos(rad_c1_lon)
c1x <- R * cos(rad_c1_lat) * sin(rad_c1_lon)

c2y <- R * cos(rad_c2_lat) * cos(rad_c2_lon)
c2x <- R * cos(rad_c2_lat) * sin(rad_c2_lon)

c1_lat_sd <- round(sd(c1x),4)
c1_lon_sd <- round(sd(c1y),4)

c2_lat_sd <- round(sd(c2x),4)
c2_lon_sd <- round(sd(c2y),4)

#text(x = c1_1[1], y = c1_1[2],sprintf("%s",paste("Lat SD:
",round(c1_lat_sd,2),"m")),pos=1)
#text(x = c1_1[1], y = c1_1[2],sprintf("%s",paste("Lon SD:
",round(c1_lon_sd,2),"m")),pos=3)
```

```
#text(x = c2_1[1], y = c2_1[2],sprintf("%s",paste("Lat  SD:
",round(c2_lat_sd,2),"m")),pos=1)
#text(x = c2_1[1], y = c2_1[2],sprintf("%s",paste("Lon  SD:
",round(c2_lon_sd,2),"m")),pos=3)

BP <- matrix(c(c1_lat_sd,c1_lon_sd,c2_lat_sd,c2_lon_sd),ncol=4)
colnames(BP) <- c("Longitude SD","Latitude  SD","Longitude  SD","Latitude  SD")
tab <- as.table(BP)
dev.new()
xx  <-  barplot(tab[1,],col=c("purple","purple","blue","blue"),ylab="Standard   deviation
in m",ylim=c(0,1))

## Add text at top of bars
text(x = xx, y = tab[1,], label = paste(tab[1,],"m"),  pos = 3, cex = 2, col = "red")
```