NTNU
Norwegian University of
Science and Technology

Investigating Positional Accuracies from unsynchronized independent measurements for autonomous cooperative intelligent transportation systems

David Sebastian Lange

Master of Science in Engineering and ICT
Submission date: June 2018
Supervisor: Hossein Nahavandchi, IBM
Co-supervisor: Trond Arve Haakonsen, Statens Vegvesen

Norwegian University of Science and Technology
Department of Civil and Environmental Engineering

Abstract

Positional real-time data is an important part of a cooperative intelligent transportation system (C-ITS). In order to maneuver and control such an environment safely and efficiently, each autonomous traffic participant must have information about positional accuracy for themselves and their surroundings. There are a variety of ways to determine and compare positional data of vehicles in motion.

GNSS systems are commonly used as a positioning tool in navigation and surveying. With the help of an accurate measurement time, travel trajectories can be predicted. As a result, accurate measurements in both time and position are important in a real-time cooperative system. In order to verify positional accuracies while in motion, a setup where several independent systems are simultaneously tracking the same object might be used.

In this thesis, positional accuracies of GNSS Receivers and Total Stations are investigated, analyzed, and compared. Since the two systems are independent and have unsynchronized internal time references, an interpolation must be performed in order to prepare the data for numerical analysis. A wide theoretical background is provided, including insights into the field of GNSS, ITS and positional accuracies. An experiment, tracking a vehicle in motion in real time, was conducted and is described. The resulting datasets have been analyzed, and an interpolation script has been developed, which is explained in detail. The acquired data value results are investigated to determine possible correlations and inaccuracy sources. Strengths and weaknesses of this general approach to verify positional accuracies are discussed. Finally, a short template for what could be done in the future is given.

Sammendrag

Posisjonell sanntidsdata er en viktig del av et samarbeidende intelligent transportsystem (C-ITS). For å manøvrere og kontrollere et slikt system trygt og effektivt, må enhver selvkjørende trafikkdeltager ha informasjon om posisjonsnøyaktigheter for seg selv og sine omgivelser. Det finnes en rekke måter å bestemme og sammenligne posisjonsdata for kjøretøy i bevegelse.

GNSS-systemer blir ofte brukt som posisjoneringsverktøy i navigasjon og landmåling. Ved hjelp av nøyaktige målinger i tid kan fremtidige posisjoner forutsies. Det betyr at nøyaktige målinger for både tid og posisjon er svært viktige i et sanntidsystem. For å sjekke posisjonsnøyaktigheter for objekter i bevegelse, kan flere uavhengige systemer spore det samme objektet samtidig.

I denne oppgaven undersøkes, analyseres og sammenlignes posisjonelle nøyaktigheter av GNSS-mottakere og totalstasjoner. Siden de to systemene er uavhengige og har en usynkronisert tidsreferanse, må en interpolering utføres for å forberede dataene på numerisk analyse. En bred teoretisk bakgrunn er gitt, som gir innsikt innen GNSS, ITS og posisjonsnøyaktigheter. Et eksperiment der et kjøretøy i bevegelse blir sporet i sanntid ble utført og beskrevet. De resulterende datasettene har blitt analysert, og et interpoleringsskript har blitt utviklet, noe som forklares i detalj. De oppnådde dataverdiene blir undersøkt for å fastslå mulige korrelasjoner og kilder til unøyaktigheter. Styrker og svakheter i denne generelle tilnærmingen for å verifisere posisjonsnøyaktigheter diskuteres. En kort mal for hva som kan gjøres i fremtiden er gitt.

Acknowledgements

This thesis marks the completion of a Master's Degree in Engineering and ICT, with a specialization in Geomatics at NTNU.

I would like to thank my two supervisors, Hossein Nahavandchi and Trond Arve Haakonsen for the help trough-out the duration of the project. The input and comments towards planning, programming and writing were very appreciated.

Appreciations are also extended to Christer Gabrielsen from The Norwegian Public Roads Administration (NPRA, Vegvesenet) and Radoslaw Raczynski at NTNU for the additional help during the conduction of the experiment.

David Lange

Trondheim, June 2018

Contents

List of Figures

List of Tables

Acronyms

C-ITS - Cooperative Intelligent Transport Systems

CQ - Coordinate Quality

EDM - Electronic Distance Meter

ETRS89 - European Terrestrial Reference System

G.E. - Gross Error

GNSS - Global Navigation Satellite Systems

GPS - Global Position System

ITRS - International Terrestrial System

ITS - Intelligent Transport Systems

LIDAR - Light Detection and Ranging

NN2000 - Normal Null 2000

NPRA - The Norwegian Public Roads Administration (Vegvesenet)

NTNU - Norges teknisk-naturvitenskapelige universitet

PDOP - Position Dilution Of Precision.

Radar - Radio Detection and Ranging

RTCM - Radio Technical Commission for Maritime Services

TS - Total Station

Chapter 1

Introduction

1.1 Background

Positional Data, the knowledge of where you are, plays a vital role in navigation and transport. Applications rely on accurate measurements in real time. The confidence in a measurement, including its accuracy, is important in order to use the acquired data correctly. As accuracies and confidence increases, more related real-world applications become available. When speaking of accurate measurements, both position and time are of interest.

Every mobile phone has a receiver for real-time absolute positioning, but their accuracies are up to several meters. For a lot of applications, these accuracies will not be enough. On the topic of self-driving cars, a few meters might be detrimental. Autonomous vehicles being pilot projected today are not solely relying on satellites to determine where they are, but use a variety of sensors to explore and figure out their surroundings. As is discussed in this thesis, a large number of datatypes, including satellite positioning data should be used by autonomous vehicles. In order to use the data confidently and properly, it is important to understand the data's time and positional accuracies.

Since raw data coordinate quality measurements are derived from the receiving instruments themselves, they are not truly independent. In order to verify accuracies, and investigate if accuracies change when in motion, a separate way of detecting positions has to be set up. A total station is a surveying instrument with high accuracy measurements that can be used to confirm the satellite measured positions.

Measuring a single target with two independent methods while in motion, makes a time synchronization issue appear. If the datasets are not in the same time reference system, there is no guarantee that individual point measurements are taken at the same time. In this case, the two time reference systems would be the total station and the satellite receiver, given by their separated internal system clocks. In order to solve this issue, a time interpolation method needs to be developed. Resulting data can be analyzed and evaluated for positional accuracies later on.

The idea for this thesis came naturally trough a project[1] conducted in the fall of 2017, investigating coordinate quality raw data from GNSS measurements by The Norwegian Public Roads Administration (NPRA) in traffic situations. The fall project similarly concerned itself with GNSS accuracies and ITS, giving a great basis for this Master's thesis.

1.2 Objectives

The main objectives of this Master's project are:

1. Investigating the correlation of time and positional accuracies of an object in motion.

2. Analyzing if GNSS accuracies given by internal raw data are reliable and if those can be verified with Total Station data.

3. Testing if accuracies change when measuring while in motion.

4. Developing a method and script to allow not time-synchronized datasets to be analyzed together.

1.3 Approach and limitations

To test accuracies, an experiment is to be conducted where a total station and a satellite receiver are measuring and tracking a target simultaneously. The target will be mounted on a vehicle. This will result in a variety of datasets and data points that can be analyzed. In order to synchronize data, an interpolation script needs to be developed and written, which in turn returns a variety of numerical output data for analysis. The directions of accuracies will be considered individually at first by separating the values into normal (perpendicular) and along (parallel/time) to the travel trajectory.

There are a number of limitations in terms of scale, budget and time. This a Master's project, so there is limited time and resources to test. The entire process is limited to a single university semester. Having a large-scale experiment with big amounts of vehicles, rented specialized locations, large amounts of expensive equipment and personnel is not reasonable. In that sense, collected data might be limited, and conducting the experiment several times, or in specialized locations is not feasible.

1.4 Existing solutions combining total stations and GNSS receivers

The purpose of combining the two systems is usually to help a surveyor during the recording of points. In terms of intention, this is usually not directly related to navigation and/or traffic, but surveying for the construction business. It might sometimes be difficult to get a good overlooking position when using a total station, especially if the construction site is operational at the time of the survey. When recording points with a total station, the line of sight might be interrupted, and the GNSS Receiver is then used to store points instead. That way, the surveyor still has positional information when post-processing, even if the line of sight was broken for a little while. The empirical accuracy of a GNSS Receiver is worse than a total station, but for a lot of surveying application, short stretches with a worse accuracy might still be usable.

Examples of this kind of integration include:

- Leica SmartStation

  A total station with a GNSS receiver on top, a hardware product manufactured by Leica Geosystems[2].

- Land2Map

  A software and controller hardware solution for using GNSS during a total station survey when direct sight to a total station isn't available. Manufactured by SogeLink[3].

Chapter 2

Theory

In this chapter, the fundamentals and background theory are discussed.

2.1 ITS and Autonomous driving

Intelligent Transport Systems (ITS) are systems of sensors and communications that streamline and automate traffic. The technology can be used to help traffic flow, planning, efficiency, security, and risk management. The systems are constantly advanced and have been developed for many decades already[4]. Through the continued advancements within digitalization and accessibility of the Internet, ITS has had great development opportunities for the future. The automation of vehicles and traffic has been a big part of modern ITS progress. ITS systems may contribute and help future autonomous vehicles with a large variety of tasks.

2.1.1 Sensors and equipment

Autonomous vehicles today use a variety of sensors to observe and evaluate their surroundings. Different sensor technology may serve different roles, and help the vehicle gather a large amount of various useful information. Image Recognition uses cameras, and is intended to imitate human sight. Cameras can be positioned strategically trough-out the vehicle, pointing in all directions to help the machine understand its surroundings. Stereo vision cameras can be used to generate depth perception. The picture and video footage is recorded and can be analyzed in real time. Advanced Data Vision algorithms are used to find and recognize objects such as other vehicles, pedestrians and street signs. Recognition algorithms are constantly being improved, but have struggled to compete with a 'human' understanding of surroundings so far. In addition, image recognition might struggle with weather conditions and long-range detection.

Radio Detection and Ranging (Radar) and ultrasound sensors can determine distances to obstacles around the vehicle. Most modern vehicles already have these kinds of 'simple' sensors for functionality like automated parking and Cruise Control.

Light Detection and Ranging (LIDAR) technology uses a laser to map out the vehicle's surroundings. This can create 3D Imagery for everything the laser is able to see. Especially in combination with cameras and image recognition, LIDAR technology can be a very powerful tool for autonomous driving. Many attempts at developing self-driving cars are based on LIDAR, with Tesla being the big exception[5].

In addition to these sensors, satellite signals may be used to determine position. An advantage to using this technology is that found positions are absolute, instead of relative to the vehicle. This aids navigation, and long-term decision-making for routes, lane-changes and can be used as a tool to prepare for upcoming situations along that route. Knowledge of where you are in the world is an essential part of traffic decisions.

Simply knowing your surroundings is not enough to safely traverse in complicated traffic situations. In addition, a powerful, well-trained algorithm needs to interpret the found information and determine the best cause of action depending on the situation. There are still many issues, even with this approach. One possible problematic situation is where two autonomous vehicles are driving towards each other on course to a collision. Both vehicles correctly identify the other vehicle and the fact that they might crash. Both execute an attempt to dodge the approaching vehicle, but because neither knows about the others decision, they attempt to dodge in the same direction, causing an otherwise easily avoidable crash. This example shows that there is another big component missing for a complete autonomous traffic system.

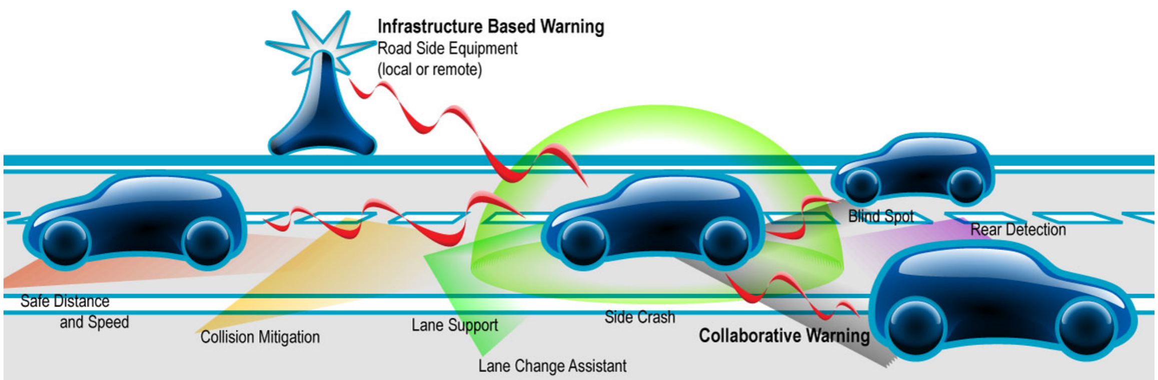

Figure 2.1: ITS Example: European FP6 research project SAFESPOT[6]

2.1.2 Cooperative Intelligent Transport Systems

In an ITS system with autonomous vehicles, there are a large amount of data collected from various traffic participants and permanent road-side installations. Sharing information between each other would create an entirely new level of control, safety, and possible efficiency. This kind of system is known as a Cooperative Intelligent Transportation system (C-ITS). Direct communications between two or more participants and cloud-based cooperation are both needed for a complete solution. An example of this would be a roadblock being signaled to all oncoming vehicles in advance and redirecting traffic efficiently and well coordinated.

Drawbacks to cloud-based communications systems might be the number of connected vehicles. While in an ideal scenario all participants are connected, in reality, the automation of traffic is gradual and slow. There will be the need for non-autonomous and autonomous vehicles to coexist and drive near each other for several decades. Another issue might be timing. Currently, uploading, analyzing and downloading information to a cloud might take a few seconds (depending on various factors including connectivity and server location), which is far too much for an intense troublesome traffic situation with lives on the line. Efficient sending and receiving of information is crucial in an optimized system. In direct communication, the delay would be shorter, and a promising upcoming technology, ITS-G5[7] could be a big step towards fast, efficient and reliable communication between units. In addition, its goal is to incorporate all devices regardless of manufacturer and country. Both vehicle-to-vehicle and vehicle-to-infrastructure communication will be available.

2.2 GNSS

An essential concept within ITS and navigation is real-time positional knowledge. Knowing where you, other traffic participants, and your surroundings are at any time can help create a faster, safer and more reliable transportation system. The Global Navigation Satellite System (GNSS) can provide positional data in real time by means of distance measurements towards satellites in orbit around the earth. The collective term GNSS entails multiple systems, all currently deployed and usable around the world. The most notable one is the Global Position System (GPS), which is an American system. There is also the Russian system GLONASS, the Chinese system Beidou and the European system Galileo. Beidou and Galileo are still under development for global coverage, both expected to achieve full service in 2020[8][9].

Beidou currently only has regional coverage. In addition, both India and Japan have developed their own regional systems as well.

A lot of devices do already contain small GNSS receivers, and the number of devices is expected to rise as more and more technology that relies on positioning is developed. Receivers may vary in size, accuracy, and prize. When recording points, receivers may store metadata, including a timestamp for when the measurement was made.

2.2.1 Error/Accuracy Sources in GNSS

In order to receive satellite signals, the receiver must see the satellite. Weather conditions can be largely be ignored, but the GNSS signals are transmitted with frequencies that have problems getting through a variety of materials, including bedrock and concrete. This means that a receiver may fail to receive the signals where sight is limited or not available, including indoors and tunnels. An additional problem might be multipath, which occurs if the satellite signal bounces of a surface before getting to the receiver. This reflective surface would in urban areas often be a high-rise building. The resulting indirect path causes the signal to have a slight time delay, which in turn results in an incorrect calculated position. High-end receivers and their antennas are generally good at detecting and resolving multipath.

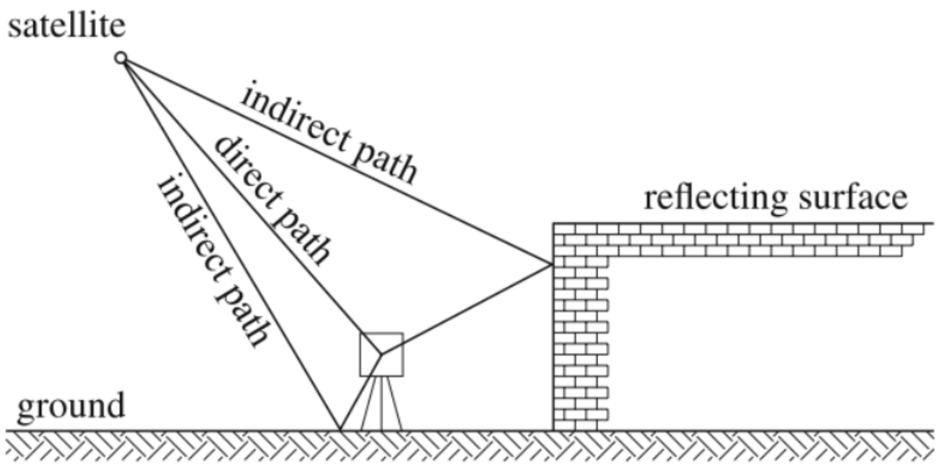

Figure 2.2: Multipath explanation[10]

Another possible source of error and cause of inaccuracy might be selective availability or bad satellite geometry. Sometimes, due to the positioning of the satellites and the receivers surroundings, only a few satellites might be visible to the receiver. If in addition, in case the satellites are not spread out very well, measurements might become less accurate. The quality of the satellite geometry can be measured based on the Position Dilution of Precision (PDOP) value. A low PDOP value would indicate an accurate measurement.

Ionospheric and tropospheric delays on the signal are also issues that need to be corrected. Corrections will be very similar locally, meaning that a network of base station receivers might be able to correctly predict corrections based on differential observation. This practice is known as *Differential GNSS*. Other methods might involve prediction models and use several signal frequencies.

2.2.2 Code Measurements vs. Carrier Phase Measurements

Measurements with GNSS signals for position calculation are based on distances between the satellites and the receiver. There are two main methods for figuring out these distances.

- Code Measurements, based on the pseudorange distances on the code
- Carrier Phase Measurements, based on the carrier wave cycle

The difference between the measurement accuracy of these two methods is significant. Code measurements have an accuracy of up to several meters, while carrier phase measurements operate on centimeter accuracy. It is thus desirable to use phase measurements if possible, but this requires a geodetic receiver. Simple, handheld GNSS receivers are typically based on code measurements. These measurements are based on both recipient and satellite generating the same code at the same time (assuming the times are synchronized correctly). Thus, when the receiver receives the signal from the satellite, the time difference (Δt) between when the signal was sent and when the signal was received is easily calculated. Since the signal speed (speed of light ($c \approx 300000 km/s$)) is known, the distance (D) to the satellite can be calculated based on the formula:

$$D = c \cdot \Delta t + corrections \tag{2.1}$$

For economic reasons, it is not desirable to require all receivers to contain atomic clocks, which means that there must be possibilities for correcting any errors in clock time. This is done by means of pseudo-distances, which means that the computed distance to a satellite does not have the correct value, but equally much offset with all satellites. This creates a linear system of four unknown (three directions and time deviation), which means that there is a minimum of four visible satellite required to find the receiver's position. Since code measurements are made in real time, code measurements do not need to be continuous[11].

Figure 2.3: Code Measurements explanation[12]

Carrier phase measurements are based on distance changes over time between satellites and the receiver antenna. The distances (D) at different times (t) are found using the number of wavelengths (N) and by measuring the phase (Φ) on the carrier signal. Wavelengths of the carrier signal (λ) (for GPS: *L1*, *L2* or *L5*) are known.

$$\begin{cases} D_0 = N_0 \cdot \lambda + \Phi_0 \cdot \lambda + corrections, & \text{at } t_0 \\ D_i = N_0 \cdot \lambda + \Phi_i \cdot \lambda + corrections, & \text{at } t_i \end{cases} \tag{2.2}$$

Since N_0 is an unknown integer number of wavelengths, measurements must be made over time to determine a value. This is because the satellites need to change their position so that N_0 can be determined. Thus carrier phase measurements need continuous signals and will be interrupted by temporarily losing contact with satellites[11].

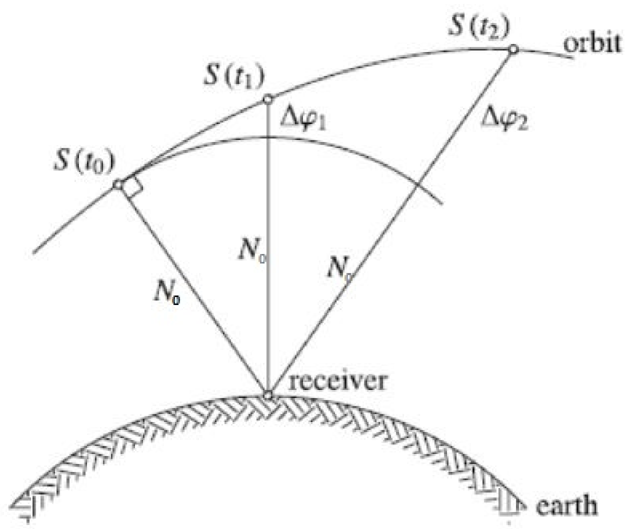

Figure 2.4: Carrier Phase explanation[10]

In order to improve accuracy, accurate corrections need to be found. This can be done by differential point observation. By observing at least two points at the same time, we can find a coordinate difference (usually called a baseline) between the points. If the other point is a known point (base station), the accuracy can be made significantly better. For a receiver in motion, continuous baselines are a bit difficult to achieve since this might require many base stations and fully automatic and efficient switching between these base stations.

2.2.3 CPOS

CPOS is a real-time centimeter accuracy positioning service, which is provided by the Norwegian Mapping Authority (Kartverket)[13]. The service requires continuous internet access, which in these measurements is achieved through the mobile network. The service sends corrections for position to the recipient in real time based on correction data located on their servers. Virtual reference stations are calculated based on data from permanent geodetic base stations. The stations are selected based on the recipient's location. The requirement for constant connection may cause problems in areas with poor mobile coverage.

Correction data from CPOS is delivered in the Radio Technical Commission for Maritime Services (RTCM) format, which provides corrections in the official reference frame EUREF89. EUREF89, sometimes also called *European Terrestrial Reference System 1989 (ETRS89)* is a regional reference frame for the Eurasian tectonic plate, and realizes the Earth's surface as an ellipsoid. For positions in EUREF89, *Universal Transversal Mercator (UTM)* is typically used, which is described with Northing (North-South Value) and Easting (East-West Value). Norway uses EUREF89 UTM as the official coordinate system. The UTM coordinate system is based on a cylinder projection on the Earth that divides the globe into 60 zones, which are again divided into North (N) and South (S). Norway uses the zones UTM32N to UTM35N. In addition, *Normal Null 2000 (NN2000)* is used as the vertical reference frame.

International Terrestrial Reference System (ITRS) is a global reference frame. In 1989, EUREF89 and ITRS were identical, but they move in relation to each other in accordance to the platonic drift. The plate displacement is only a few centimeters a year.

GPS coordinates are given in the current *World Geodetic System (WGS)*, which is WGS84. The coordinates are given in latitude (North-South direction) and longitude (East-West direction). The difference between ITRS and WGS84 is only a few centimeters[14].

Values for accuracies of CPOS depend on the distance between base stations. In addition, GNSS error sources (section 2.2.1) might make the measurements worse.

Table 2.1: CPOS expected accuracies

	Areas with $\approx 35km$ between base stations		Areas with $\approx 70km$ between base stations	
... of values within	66%	95%	66%	95%
2D EUREF89	8mm	16mm	14mm	28mm
Height EUREF89	17mm	34mm	30mm	60mm
Height NN2000	20mm	40mm	36mm	72mm

The values are based on the Norwegian Mapping Authority's own analysis of CPOS data[13]. Fundamentally, these values are given for stationary points, not while in motion.

Figure 2.5: A permanent CPOS base station[15]

2.3 Total stations

A total station (TS) is a piece of surveying equipment, frequently used in the mining, building and construction industry. The system uses an electronic transit theodolite to measure angles and an electronic distance meter (EDM) to measure distances. The idea is to very precisely determine the angle and distance to a target object from the instrument[16].

Optical theodolites, in its simplest form, have been around for centuries. Originally just a mounted telescope with simple protractors to read of vertical and horizontal angles, modern theodolites are a lot more advanced but rely on the same measurement goals. Modern theodolites read the angle measurements electronically with the help of a rotary encoder. Coupled with possible functionality for automated finding and tracking of the target object, electronic theodolites can provide near instantaneous, continuous high-precision angular measurements.

An EDM measures the distance to an object. Total stations generally use a multi-frequency infrared signal. The signal is reflected off the target and the modulation of the signal can then determine the distance to the object. This rangefinding requires direct sight to the target, and the use of a specialized reflector prism as the target object can heavily increase the maximum range of the EDM. Ranges and accuracy may vary based on TS model, prism model, and weather conditions.

When using finding and tracking functionality, a total station needs to be able to recognize its target object. These targets might vary in size and shape, from target reflective stickers for walls to circular glass prisms mounted on a ranging pole. In order to not rely too heavily on hitting the prism dead on, a 360° reflective prism might be used. That way the prism can be rotated and moved without having to worry about the total station losing its target.

With the angles and the distance, the TS can determine a three-dimensional vector representing a coordinate measurement relative to the total station. The

TS does not inherently know where in the world it is positioned, so in order to determine an absolute position, known control points can be used. The TS itself can be positioned in a known point, or known points can be measured from the TS. Once the TS position is known relative to its control points, a geometric transformation can be performed to transform the data points into a known coordinate system. This method is known as *free stationing*[17].

A TS can record and store points, which can later be exported to another device for post-processing and analysis. In addition to coordinates, the station stores some metadata about each data point including a timestamp.

2.4 Internal time reference systems

Independent systems recording data will store data in different time reference systems. Timestamps being recorded and exported in surveying equipment are received from the internal firmware and its internal system clock. This system clock is system dependent, **meaning that clocks in separate systems are not inherently synchronized with each other**. In other words, that means that there is no guarantee that two measurements with the same timestamp from different systems were actually recorded in the same moment. While recording a stationary point with two independent systems might not be a problem, recording a moving target requires very accurate and confident synchronized times in order to compare individual points with each other.

If the two independent system cannot be synchronized, there is an unknown time offset between the two datasets that needs to be found. With high-frequency data, this can be achieved through a time interpolation (section 3.2).

Basic system clocks might experience clock drift, which might mean that the time offset value between two datasets can change slightly over time. In the case of this particular project, the data was collected over a short period of time, so clock drift can be ignored.

Chapter 3

Data Collection and Processing

In this chapter, the entire process from planning to collect data to having post-processed data values to analyze is explained in detail. This includes planning, hardware, collection, transformation, interpolation and error detection.

3.1 Data Collection

3.1.1 Date and Location

The experiment was conducted on March 20th, 2018.

A few things have to be considered in choosing a test area site. Since there are total stations involved, the site should be an open area without any major obstacles that might temporarily obstruct sight to the target prism. The area has to be easily traversable and accessible for vehicles. Obviously, the GNSS signal requires an open air space, and tall surrounding buildings should be avoided for possible multipath reasons. Traffic might interfere when driving a vehicle around as well. A good choice for a test area is an open, level parking lot. For convenience, the parking lot **P25**, south of Høgskoleringen 7B, NTNU Gløshaugen was chosen.

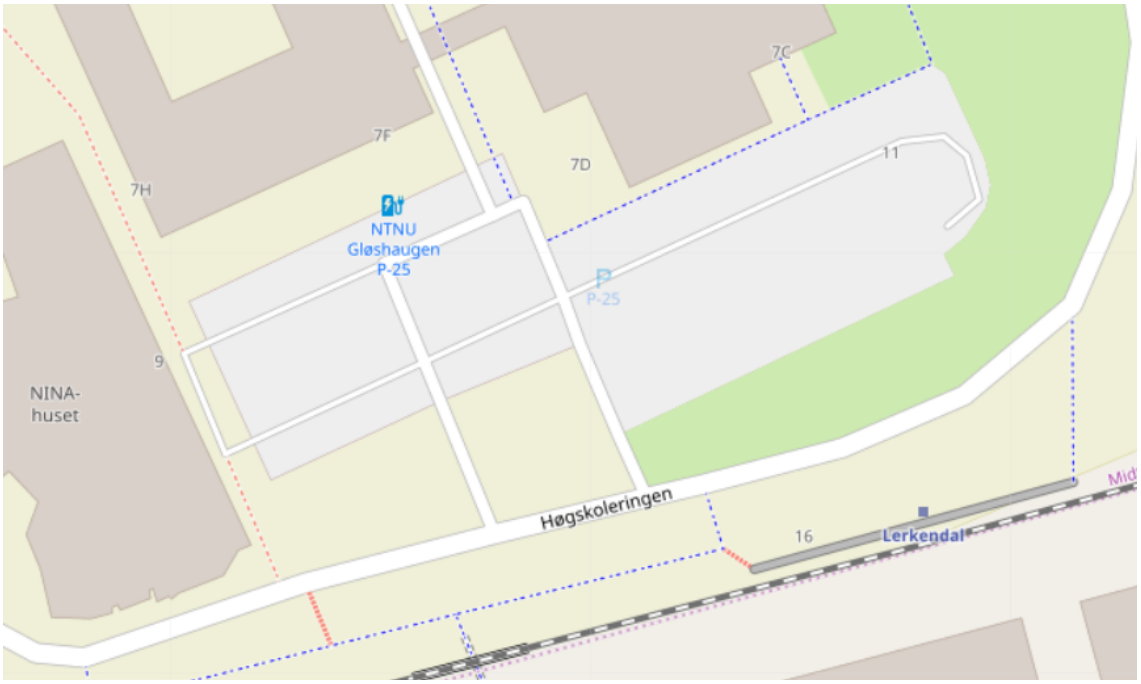

Figure 3.1: Experiment Location[18]

Figure 3.2: Experiment Location (Streetview)[19]

3.1.2 Hardware Equipment

GNSS Receiver: Leica Viva GS16

The Leica Viva GS16 is a GNSS receiver produced by Leica Geosystems[20], which is an independent company under the Leica brand. Leica Geosystems manufactures and sells a variety of surveying equipment.

The receiver returns high-accuracy EUREF89 UTM32N coordinates with the help of CPOS (section 2.2.3).

Important Raw Data Output:

- PointID
- Type of measurement (Code, Phase)
- Positional Data: Northing, Easting, Height (EUREF89 UTM32N)
- Timestamp: YYYY-MM-DD hh:mm:ss.ss (Closest $1/100th$ of a second)
- Coordinate Quality: 3D, 2D, and every direction individually

Reflector Prism: Leica GRZ122 360° Prism

The Leica GRZ122 360° Prism is a reflector Prism produced by Leica Geosystems. It is used by the total station as a target. This prism has a high centering accuracy and has a mount to connect any Leica SmartAntenna (Including the Leica Viva GS16 antenna) on top of it. During the experiment, the GNSS receiver is mounted on the reflector prism, which in turn is mounted on the roof of a vehicle.

Figure 3.3: GNSS Receiver mounted on the prism, mounted on the roof of vehicle

Total Station: Leica TCRP1200+

The TCRP1200+ is the most advanced model of the TPS1200+ series, developed by Leica Geosystems. Additional features compared to the basic TC1200+ include Auto-Lock, which is important in order to follow the prism.

With that in mind, the built-in options to Auto-follow and continuously record points automatically can be used to gather data. The collection frequency can be manually set by a controller unit, with frequencies up to $10Hz$. However, the raw data shows a frequency of about $5-6Hz$. This might be due to hardware limitations.

Two Leica TCRP1200+ Total Stations were used simultaneously to gain more independent data during testing. Dataset can independently be compared for verification and quality assurance.

Important Raw Data Output:

- PointID
- Positional Data: X, Y, Height (Local system)
- Timestamp: YYYY-MM-DD hh:mm:ss.ss (Closest $1/100th$ of a second)
- Coordinate Quality: 3D, 2D, and every direction individually

A device manual that includes a large number of technical details about the product was used to find various amounts of technical information about the product[21].

3.1.3 Setting up and recording data

It is important to set up the total stations to overlook the perimeter properly. The total stations need to be positioned in a way that the radial max velocity is not breached anywhere, and near enough where max ranges are not touched. With the GRZ122 prism, the TS manual lists the minimum range during AutoLock mode to be $1.5 meters$, and the maximum recommended range $500 meters$. These values are assuming good, clear weather conditions. Tangential speed is also limited by the radial max velocity, given by the fact that the TS needs to turn in order to follow the prism. Driving away or towards the total station instead of around it is beneficial to avoid this issue from occurring. With typical parking lot velocities, maximum TS turnrates proved to be no issue during the collection. The total stations cannot be moved until the entire data collection is done.

In order to reference the total stations' absolute position, control point must be set up for later transformation. This is a standard procedure in surveying jobs. The procedure is done manually, by placing the GNSS receiver on a marked point and measuring that point for a continuous measurement. Average values over time establish a control point. While measuring and establishing control points, the total stations are pointed towards the prism underneath the GNSS receiver, establishing common points for all data sets, which will be used to transform into a known, unambiguous, common reference system.

Prism and GNSS receiver were screwed tightly on to the roof of the vehicle. It is important that the units do not slide around, or swing back and forth. Slight vibrations are unavoidable.

The total stations are then locked onto the prism and told to auto-follow the prism wherever it moves. Continuous high-frequency measurements are recorded while the vehicle moves. At the same time, the GNSS receiver is recording its own data. The vehicle made two trips driving around the test area, each lasting a few minutes.

Figure 3.4: Setting up the hardware on location

3.2 Data Processing

A transformation needs to be done in order to prepare the data for further analysis. In addition, it is beneficial that the data is exported into file types that are easily read- and writable. Leica's own post-processing software *Leica Infinity Survey Software*[22] is designed to accomplish these tasks.

After importing all raw data (GNSS Receiver dataset and both total station datasets) in Infinity, the total stations' data needs to be transformed into a known, common coordinate system. Since the GNSS dataset is already in EUREF89 UTM32N, that is the obvious choice for the coordinate system. In order to align the total station data with the GNSS data, the three control points are used.

In order to interpolate and analyze the data from the total stations and the GNSS Receiver, timestamps need to be used. The datasets, and their data points, have timestamps for when they were recorded. Unfortunately, every device creates timestamps from their own internal system clock (Section 2.4), meaning that every device essentially operates in its own time reference system. The time systems could, of course, be manually calibrated beforehand, but since accuracy is key here there is no guarantee that the resulting data would be good enough. The system could, in theory, be synchronized automatically by connecting the devices, but if the hardware does not have existing support for this kind of operation, this is not always realistically possible.

The other alternative is a time interpolation between the datasets, looking at the gathered positional data and determining a time offset between the datasets. This means that value that needs to be found is the time difference between the two datasets.

Other useful values, *'velocity'* and *'distance to line'*, are found as by-products through-out the process.

3.2.1 Prerequisites

Raw Datasets are exported from Leica Infinity as .txt-files, and a subsequent import, interpolation, and analysis script is written in Matlab[23]. Matlab is a proprietary scripting language with a heavy focus on numerical computing, developed by MathWorks. The language is a useful tool for this interpolation because of its good ability to handle larger dataset, vector and matrix management, and useful tools for plotting and visualizing data.

There are many other possible programming language options to develop this kind of script. Most modern high-level programming languages could work.

In order to analyse and interpolate different sections in the datasets, a few requirements have to be met.

Dataset requirements for interpolation:

- Continuous Phase measurement in GNSS dataset
- Vehicle in motion
- Continuous high-frequency TS data

3.2.2 Step-by-step methodology for interpolation

The script looks at one GNSS-datapoint at a time. The search method to find the correct closest point in the total station dataset is to find the lowest distance to any total station point. This is done continuously with an iterative loop with all GNSS measurements. However, it is important to limit the search for closest total station point a little, since there needs to be a certainty that there are no overlapping values in the data (since the vehicle drove over similar points several times in a row). Choosing a smaller, smart range of points also decreases the runtime of the script. As the script developed and became more robust, larger sections were used. Using the entire set of data at the same time would require huge amounts of overhead programming, resolving and 'cleaning' data so that the code does not crash, including a bunch of special cases (section 3.2.3) that must be found and logically processed in the code. Since the interpolation is about finding numeric values, using the whole dataset simultaneously is not necessary at all. Several shorter sections with good data at a time provide good results and are significantly easier to work with.

A line is drawn between the measurement points of the total station. In practical terms, 'Lines' in Matlab are handled as euclidean vectors. If a line consists of five point measurements, the resulting multipoint line will be represented by five vectors in the code. This way, a line defined by several points is represented with vectors between each consecutive point. The line is supposed to represent the continuous position of the vehicle (the prism and GNSS receiver on the roof of the vehicle). How exact the line fits the actual path of the vehicle depends on the frequency of points in the total station dataset. If the frequency is high enough, the line will be a good enough representation of the actual path, especially in stretches where the vehicle drives somewhat straightforward.

The target is to find the smallest distance from the GNSS point currently being checked and the line. While doing this, the position of that *'closest point on the line'* also needs to be determined. That point is the projected position of the GNSS point if it were on the line. There are simple methods for finding the distance from a point to a vector, but since there are several vectors representing a line there is a major problem that makes this issue considerably more difficult when working in Matlab.

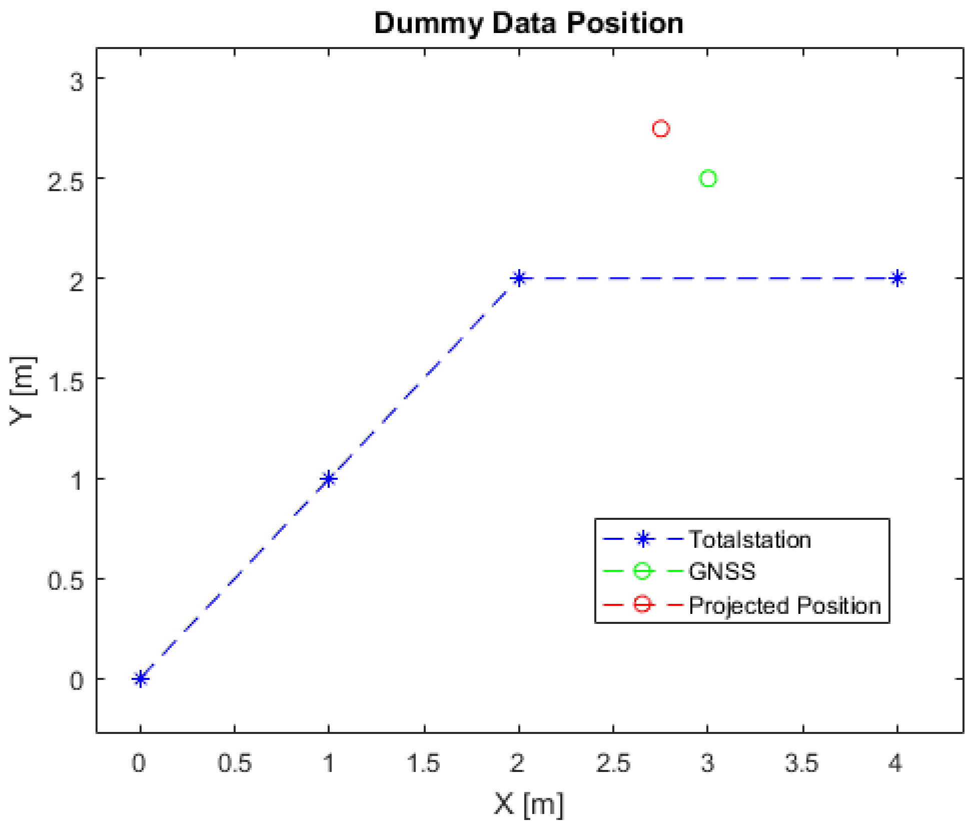

Figure 3.5: Distance to vector problem

As seen in Figure 3.5, the closest point on the line should have been projected in (3,2), directly under the GNSS point. However, the point has wrongly been projected somewhere else. This is caused by the fact that there is a different vector that is closer to the GNSS point.

In order to solve this issue, it is clear that the closest data point to the GNSS point needs to be found, and then the vectors on both side of that point need to be evaluated. It should be noted that there are other possible ways to find the closest point to a line consisting of multiple points. One way would be to use the geometry features of PostGIS, which have support for this kind of operation in their SQL based coding using the *ST_Distance* method[24].

In Matlab, the following methodology can be used to find both the distance to the line and the position on the line:

Compare the selected GNSS point and first total station point in the selected start time. The distance between the points is a simple 2D euclidean distance. The next total station point is selected, and the distances are compared. If the distance is less than before, a new best candidate for closest total station point has been found. If the distance becomes larger then the previous point was the closest point. This method is better for larger datasets (instead of checking all distances and simply finding the minimum) because of significantly lower code runtime. It also avoids rare error cases were a wrong point might be found because the vehicle drove over the same point several times which might cause the script to select the wrong closest point.

Which of the total station measurements is closest to the GNSS measurement is now known. The next step will be to figure out which side of the total station point the GNSS point is on relative to the direction of travel. This is important so that the correct vector (linepiece) is used when determining the distance to the line.

Dummy data to show an example:

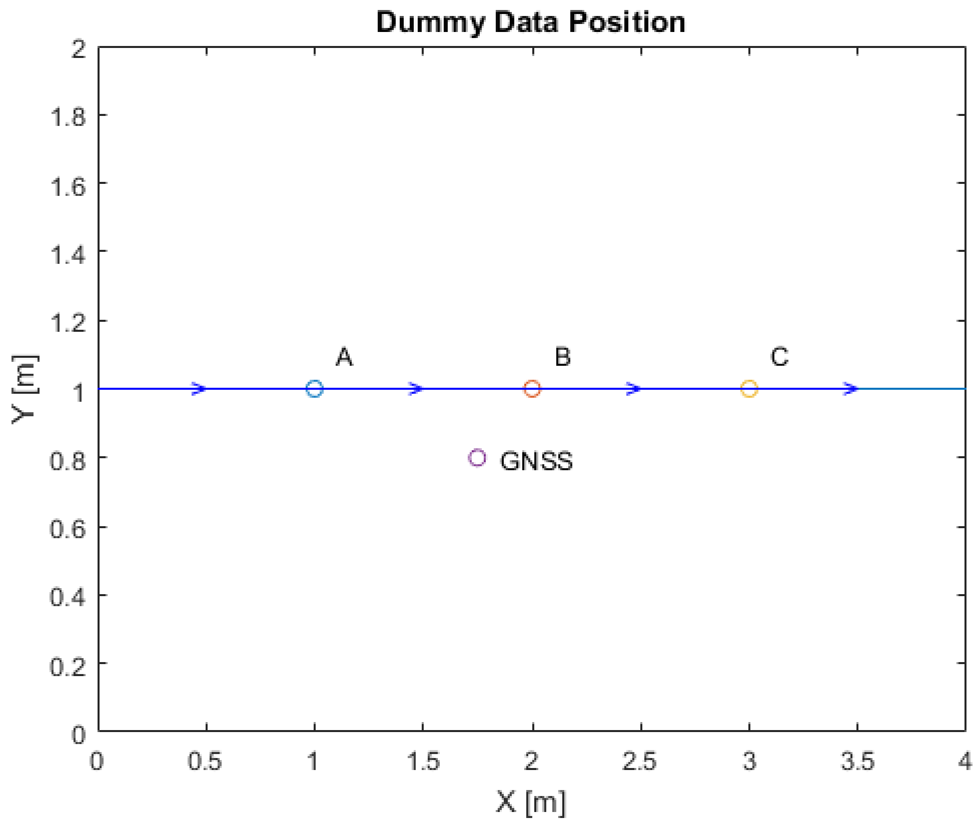

Figure 3.6: GNSS Point is behind closest Point (B)

A GNSS point can appear behind its closest total station point (Figure 3.6) in the travel direction, or in front (Figure 3.7).

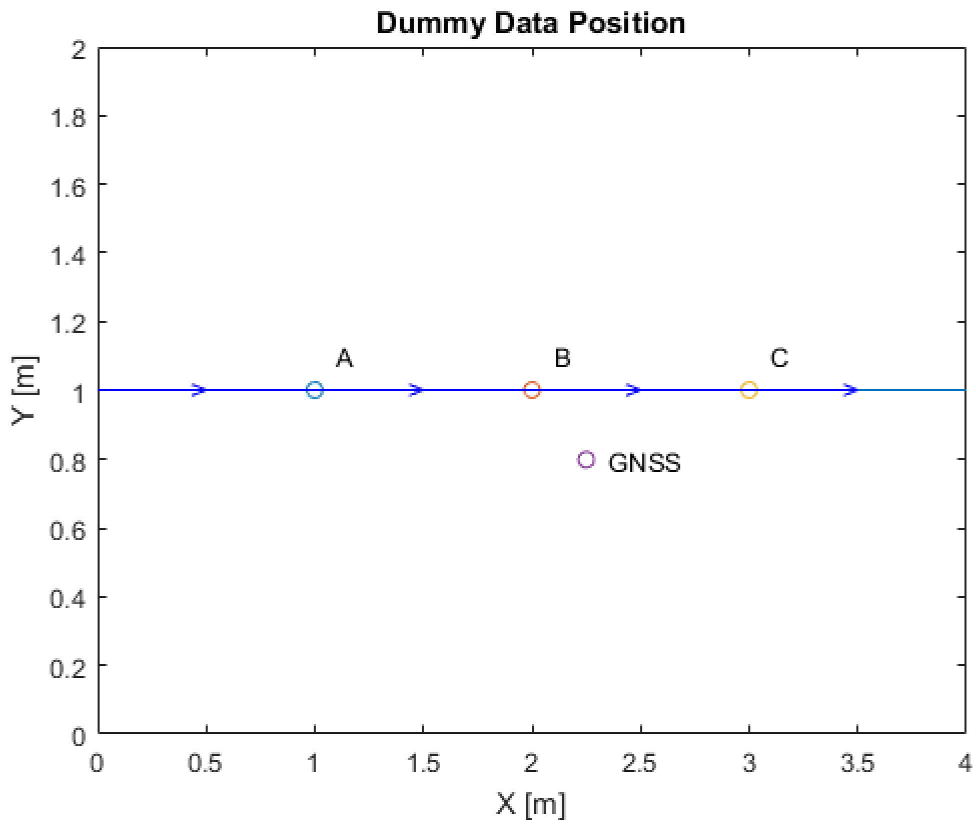

Figure 3.7: GNSS Point is in front of closest Point (B)

The easiest way to determine which side of the closest point the GNSS point is positioned is to compare normalized eigenvectors. A normalized eigenvector represents the direction (given by an x and y value in a 2D environment) of a vector with a magnitude of a unit length. By comparing normalized eigenvectors, the most similar vectors in terms of direction will be found. The normalized eigenvectors that are compared here are:

GNSS-Point to closest total station point(B) versus *Point before closest point(A) to closest total station point(B)*

and

GNSS-Point to closest total station point(B) versus *Point after closest point(C) to closest total station point(B)*

There are several ways to numerically compare two vectors, which should all ultimately result in the same. The chosen way to compare vectors is:

$$\sum |\overrightarrow{AB} - \overrightarrow{GNSSB}| = |\overrightarrow{AB}(x) - \overrightarrow{GNSSB}(x)| + |\overrightarrow{AB}(y) - \overrightarrow{GNSSB}(y)| \tag{3.1}$$

and

$$\sum |\overrightarrow{CB} - \overrightarrow{GNSSB}| = |\overrightarrow{CB}(x) - \overrightarrow{GNSSB}(x)| + |\overrightarrow{CB}(y) - \overrightarrow{GNSSB}(y)| \tag{3.2}$$

Lowest value means the most alike, indicating that the GNSS point is in the corresponding direction. Now the script knows which vector should be used to find the distance to the line from the GNSS point. To find the coordinates for the new point on the line, there are several options. The easiest one is to go from the GNSS point. The distance to the line is known, and the direction will be perpendicular to the vector that is representing that piece of the line. The only thing left to discover is if the vector should be rotated 90° or -90°. Simply checking if the new point is closer or further away from the closest total station point will solve this issue. Figuring out which direction to turn the vector effectively determines if the GNSS point is 'above' or 'below' the line. This information can be used to give the distance to line values their correct direction when mapping the inaccuracies in a 2D plane (section 4.3).

Using the data from Figure 3.6:

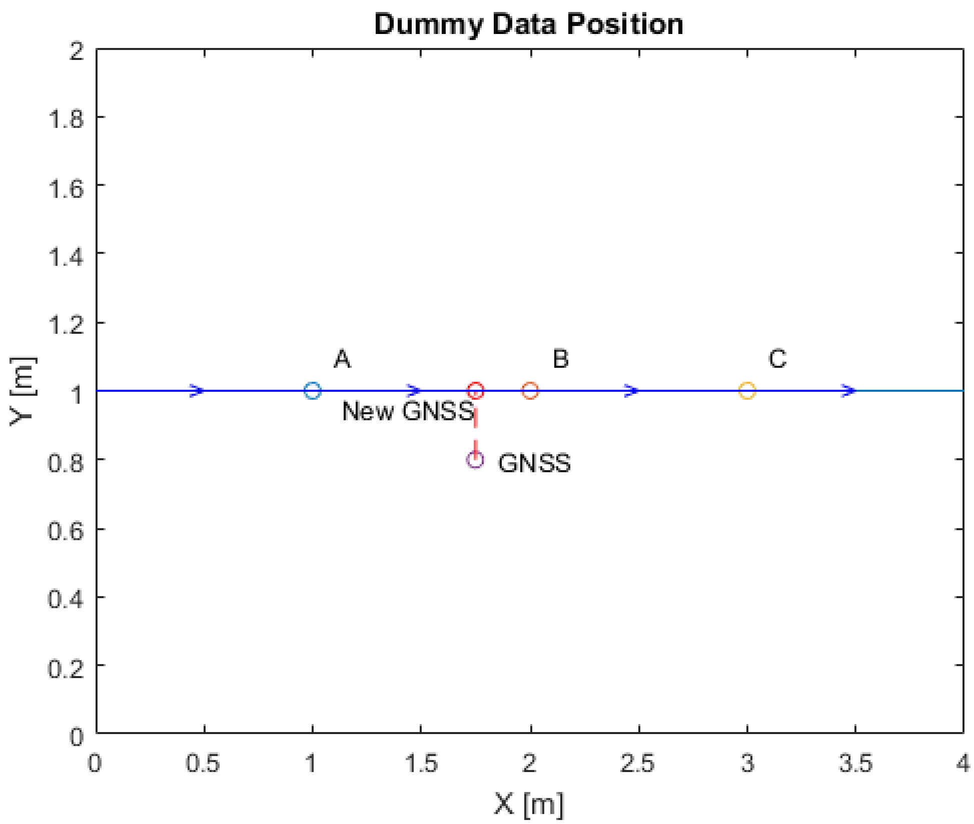

Figure 3.8: Correct vector and projected point (*'NewGNSS'*) are found

The distance between *'NewGNSS'* and *'GNSS'* will be the distance to the line, which will represent the perpendicular offset to the vehicles moving direction. From these values, accuracy perpendicular to the driving direction can be calculated.

At this 'time/moment' on the line, it is expected that the GNSS point was recorded. By looking at the previous and following total station points (and their time values), the expected time of the recording of the GNSS point can be found. **The interpolation script (*'interpolation.m'*) can be found in Appendix A.**

If all GNSS values were perfectly recorded and distributed, they would all have the same time offset to the calculated time value in the total station time system.

E.g.

$$
\begin{aligned}
&\text{Raw Data GNSS Time} &&= 14:02:30.00\\
&\text{Calculated Time in total station system} &&= 14:02:02.75\\
&\text{Time offset} &&= 27.25 seconds
\end{aligned}
$$

Now that the average time offset (based on the 160 data points for each total station) between the two time references is known, any offset from the average can be considered an inaccuracy along the travel trajectory. This time offset multiplied by the vehicles speed at that moment will give the inaccuracy in meters.

3.2.3 Dealing with special cases

Now that a time offset can be found for all GNSS points, it is time to discuss problems that might occur during the execution of the script. Based on the prerequisites for the dataset (See Section 3.2.1), many recorded point cannot be considered for time interpolation.

1. Continuous Phase measurement in GNSS dataset

Not every recorded datapoint is necessarily a carrier phase measurement. Whenever a recorded GNSS point is not a carrier phase measurement, it is either a code measurement, or a setup point for the virtual reference station for CPOS. Code Measurements might appear if the GNSS struggles with the connection to CPOS for a moment. Since Code Measurements inherently have a worse accuracy, and setup points are considered predetermined, neither are usable in analysis. The script is programmed to simply ignore a datapoint whenever it is not a phase measurement.

Larger sections of many phase measurements in a row are present in the dataset, and if a small section of code measurements interrupts the otherwise continuous measurements simply ignoring the unwanted points is unproblematic.

2. Vehicle in motion

In the beginning and the end of a session, points might have been recorded while the vehicle was standing still. This might also occur if the vehicle stops while waiting for another vehicle to pass. Since there is no movement, the points will all be spread around a stationary point, and only their normally distributed estimated accuracies will distinguish the many TS points in a row. For this reason, the initial part of the interpolation (finding the closest TS data point) will not work properly. If the interpolation were to be performed the interpolation would still find time offset values, but because the closest point chosen is arbitrary, the time offset will be random. This issue can be fixed by adding a check into to the code that verifies that the point has moved a minimum distance since the previous iteration. The minimum distance should be more than the largest inaccuracies found. Ignoring a small section where the vehicle is temporarily stopped in the middle of a session is unproblematic.

3. Continuous high-frequency TS data

The interpolation requires somewhat consistent data points in both datasets. Since every individual GNSS point is compared to all TS points, it is important that the TS points are continuous as well. Larger gaps in the TS points would cause the selected GNSS point to be compared to the beginning and the end of the gap, which might be far apart. This would not necessarily be a problem if the vehicle is driving perfectly straight with constant speed, but since this is not realistically the case, having a gap will cause a miscalculated time offset. The TS dataset might have a gap whenever it loses sight of the prism.

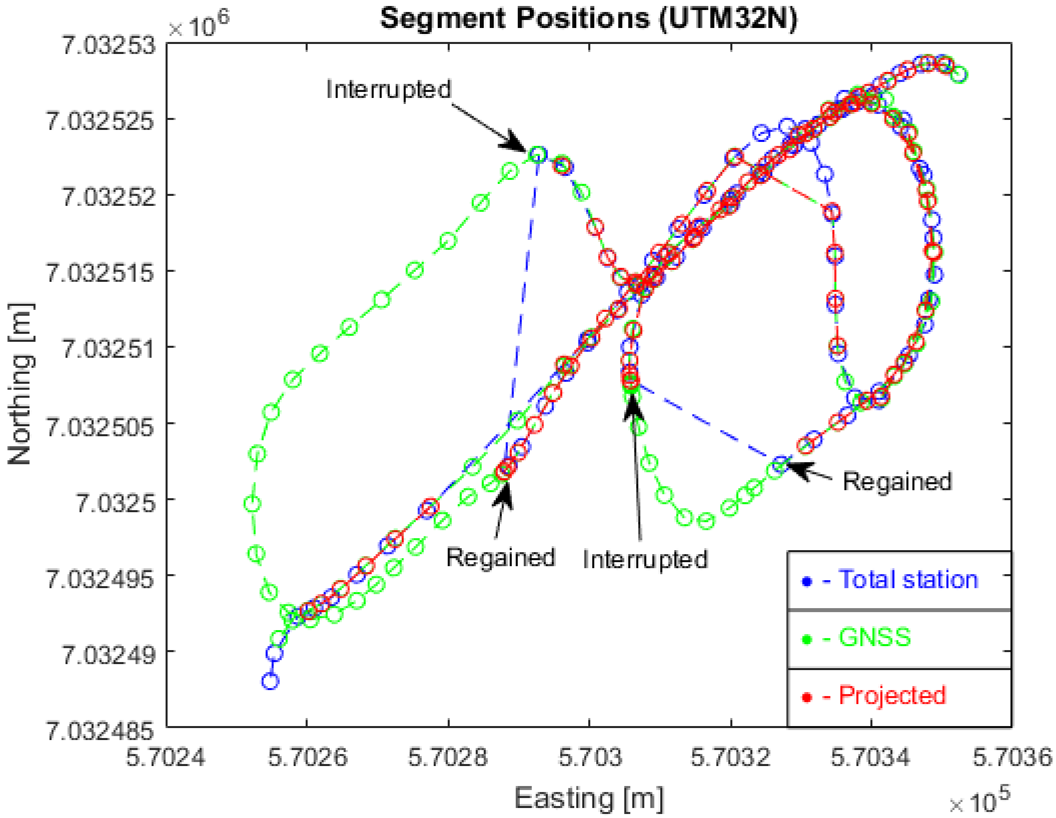

Figure 3.9: First Trip Position (Total Station 1)

As seen in Figure 3.9, total station data was temporarily interrupted twice. This is the first trip, recorded by Total Station 1.

With the stations PowerSearch function, it will most likely regain the line of sight to the prism once it reappears in the field of view and continue the capturing of points. The GNSS points that are missing their counterpart TS datapoints have to be found and subsequently ignored in the script. Similarly to dealing with code measurements, ignoring a small section of GNSS points is unproblematic. **The frequency of TS points should be at least as high as the frequency of GNSS points to ensure accurate time offset values.**

4. Special Alignment and unusual geometry cases

In addition to the three prerequisite-cases, there is also a fourth issue that needs to be addressed.

While stationary points might technically also classify as a special alignment, there are some other special cases that the script needs to be able to handle correctly. As mentioned in the interpolation, driving over the same spot several times might create issues by defining an even closer TS point to the currently selected GNSS point. As carrier phase measurements have empirical accuracies in the centimeter range, and TS points even better, this is not very likely to happen frequently, but nonetheless possible. The issue can be solved by looking at the expected following TS points first, and by tweaking the criteria of how the closest point is determined (Section 3.2.2), which is also good for runtime reasons. Another possible issue would arrive in very rare geometry cases like the one shown in Figure 3.10 (shown with dummy data):

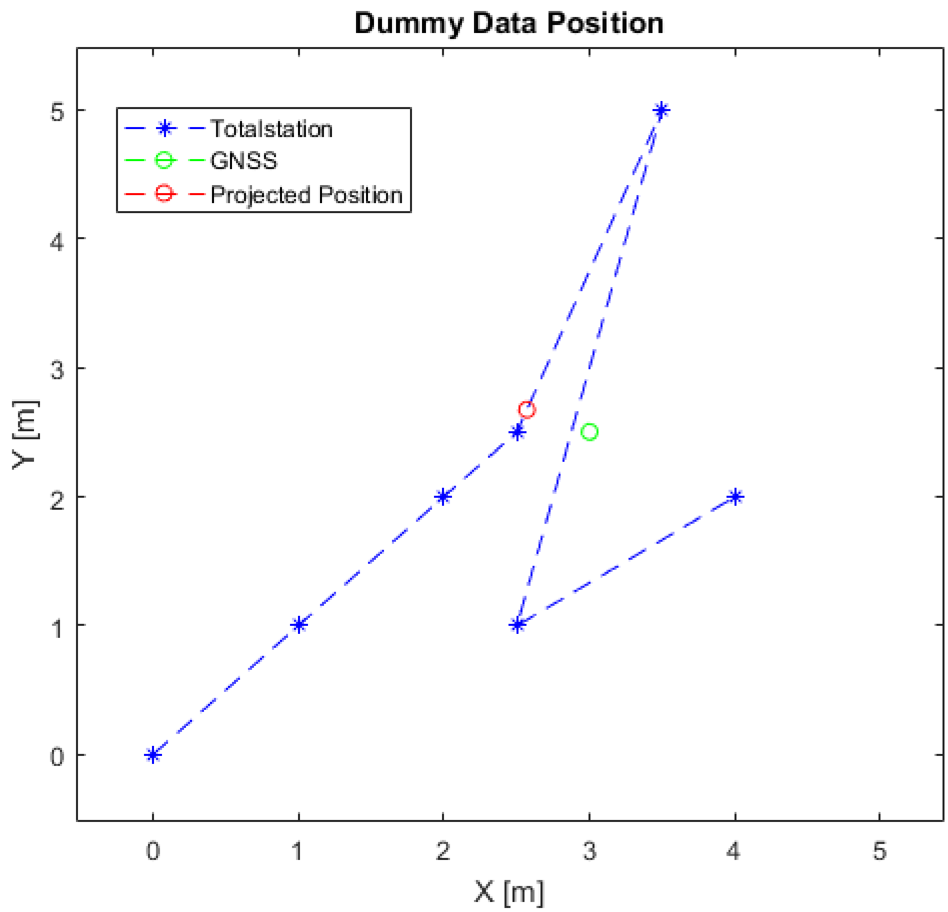

Figure 3.10: Unusual geometry causing problems

In this particular case, the script would (correctly) identify that the closest point is in (2.5,2.5), and would look at the two adjacent vectors and as a result find a projected position on the line towards (3,5). However, there is a line that is closer to the GNSS point. This causes the program to determine a presumably incorrect projected position which causes an incorrect time offset to be calculated. This example is rather extreme because these TS points do not represent a normal trajectory of a driving vehicle. However, in a large enough dataset, especially if the

TS datapoint frequency is low enough and the car is doing sharp turns, this issue might arrive. It would be incredibly difficult to think of and find a good fix for every possible geometry issue. As a result, not every such issue is addressed in the interpolation itself. However, the resulting wrong time offset values can easily be detected in the analysis later on by means of gross error detection.

3.2.4 Gross Error Detection

The handling of the special cases should find most of the systematic error values. The rest will appear as large spikes in the time offset data and are easily found. A small section of data points run through the time interpolation script with manually identified error causes is shown in Figure 3.11.

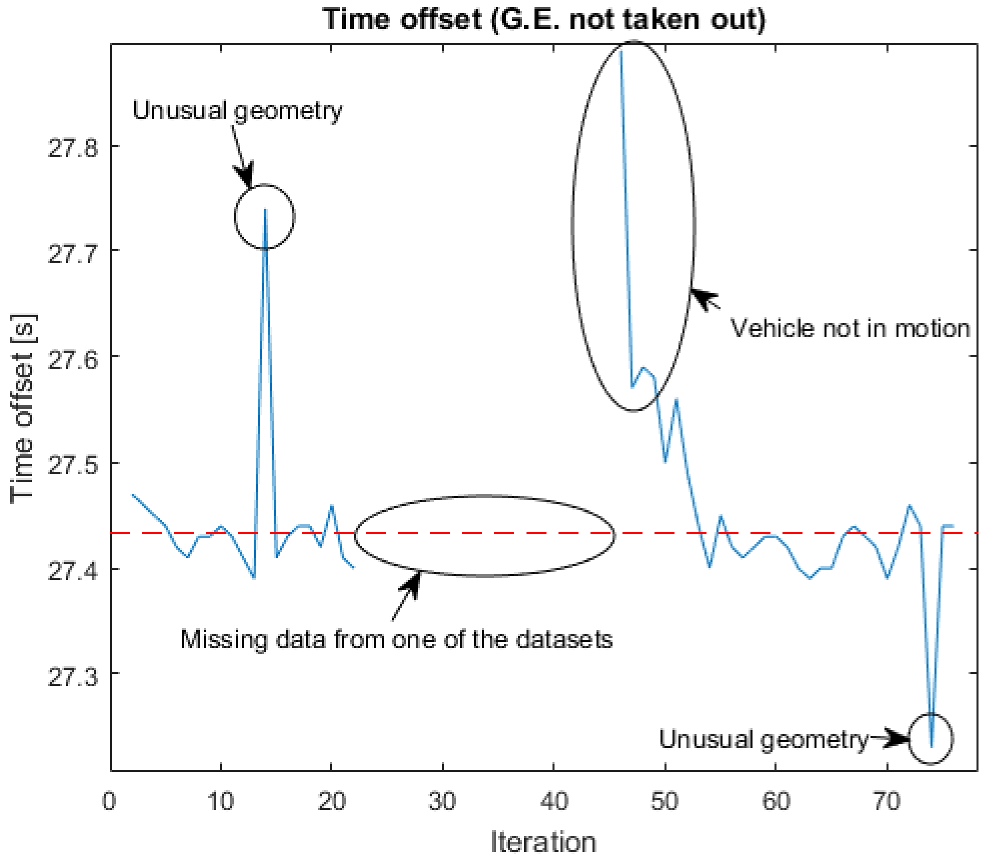

Figure 3.11: How Gross Errors might appear in time offset data

As shown, bad values are easily detectable and can be taken out of the dataset. The script might struggle with reliably identifying the correct causes, but the result is the same. The remaining data is expected to be distributed with a normal distribution around the average value (red line). Values over four times the standard deviation away from the average must be considered outliers and are most likely due to undetected systematic errors.

Chapter 4

Results

In this chapter, the numerical values and results of the interpolation are presented, analyzed and discussed. This chapter looks at Time Offset, the accuracies perpendicular and along the travel trajectory, their combined 2D accuracy, and looks at potential correlations between values. In addition, possible sources for inaccuracies and findings are determined and discussed.

4.1 Time Offset

There are a number of numerical investigations to perform on the interpolated data. The calculated average time offset from each TS dataset is important for the time accuracy. As described in Section 2.4, datasets are expected to be in their own internal time reference, and thus to have a time offset between them. Since two different independent total stations were used during data collection, their time offsets are also expected to be independent.

Table 4.1: Average Time Offsets between datasets

From	To	Average Time Offset
GNSS Data	Total Station 1	-27.43seconds
GNSS Data	Total Station 2	-2.65seconds

Both Total Stations were not calibrated in any way beforehand, so time offset values are non-zero, as expected. The numerical values themselves do not really matter, but this shows that there is, in fact, a time offset between the datasets. These values are used in the code do determine time accuracy. Larger values do not mean worse or less accurate results, just an internal time reference with more offset. The fact that both values are negative means that both time references are behind the GNSS time reference. This means that a point recorded at Time A in the GNSS dataset is expected to have been recorded at $A - 27.43 seconds$ in the Total Station 1 dataset.

4.2 Numerical Positional Inaccuracies

When talking about positional inaccuracies for a vehicle in motion, the 2D Accuracy can be represented by values along the trajectory and perpendicular to the trajectory. The alternative would be to either look at Northing and Easting separately or look at 2D accuracy overall. Looking at Northing and Easting individually is tricky, since depending on the vehicles travel trajectory, inaccuracy sources might be different in different directions. Looking at and analyzing the error sources independently can be a useful tool to spot causes individually for each direction.

4.2.1 Perpendicular to travel trajectory

The accuracy perpendicular to the travel trajectory is represented by the *'distance to line'* value found during the interpolation.

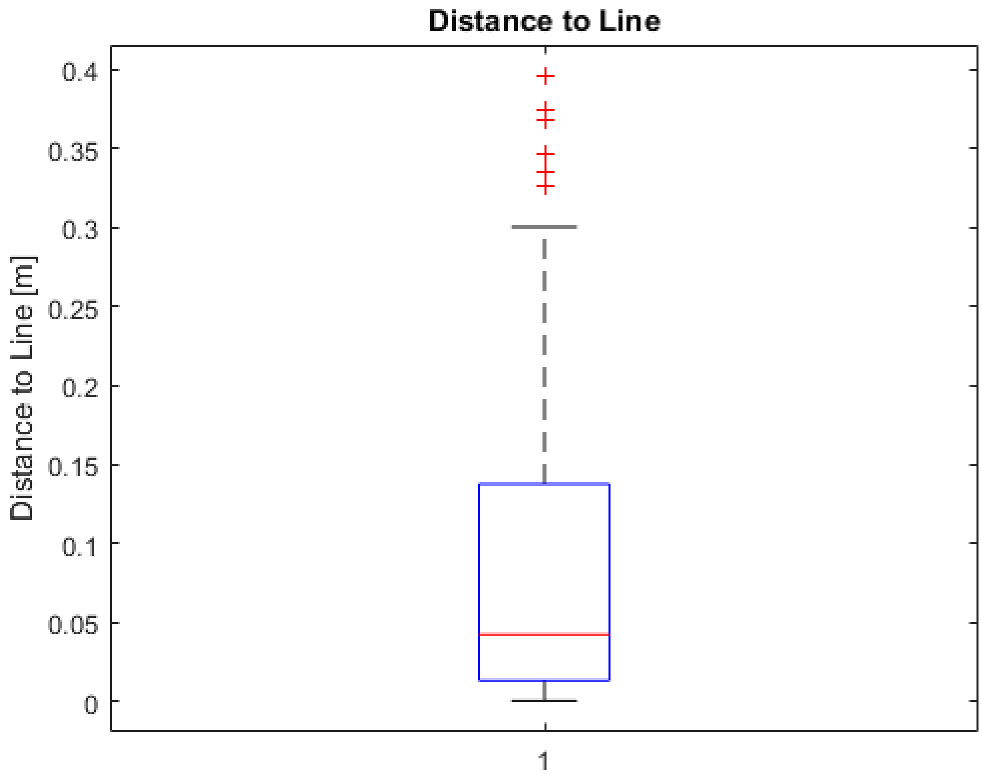

Figure 4.1: First Trip Boxplot Distance from Line

Figure 4.1 shows a boxplot of all the calculated *'distance to line'* values from Total Station 1. Since these are distances, the values will always be positive, regardless of direction. Assuming the data has a normal distribution as expected, these values should be given in a half-normal distribution, which is a special case of a folded normal distribution[25].

In a boxplot, a red line shows the median value of the data set. The box around indicates numeric values from the lowest quartile to the highest quartile. This means that 1/4 of the values are under the box, and 1/4 of the values are above the box. The outer black lines indicate ± 2.7 standard deviations that statistically should equal 99.3% of all values. The red crosses outside the outer lines show extreme values (Outliers). Matlab has a built-in plotting tool for boxplots[26].

4.2.2 Along travel trajectory/ Time inaccuracy

Terminology:

1. Average Time Offset - Average Time Offset between the datasets
2. Time Offset - Time Offset between the time references in two datasets (in the projected position)
3. Time Offset From Mean - Offset From the Average Offset in time reference between datasets

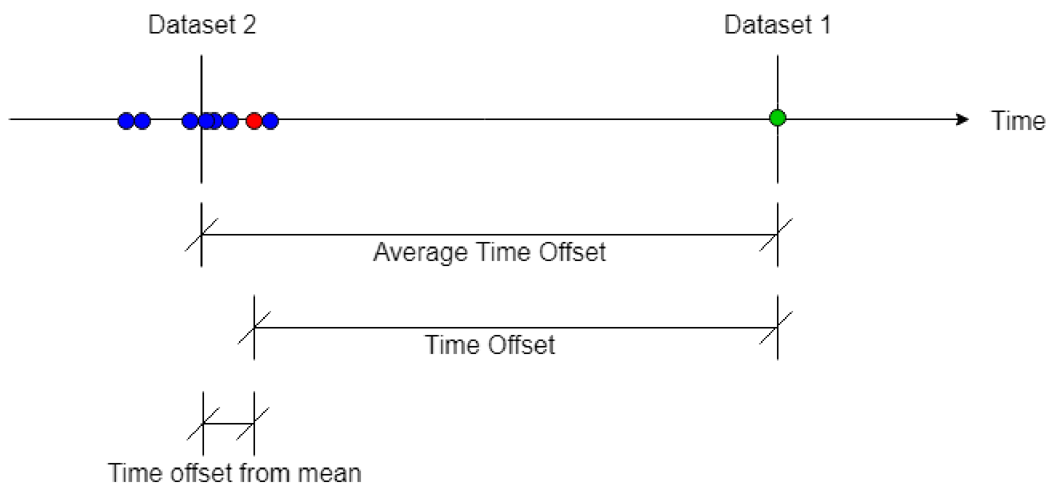

Figure 4.2: Time offset concept

This mean that:

$$TimeOffsetFromMean = TimeOffset - AverageTimeOffset \tag{4.1}$$

The accuracy along the travel trajectory can be represented by the time accuracy in the dataset.

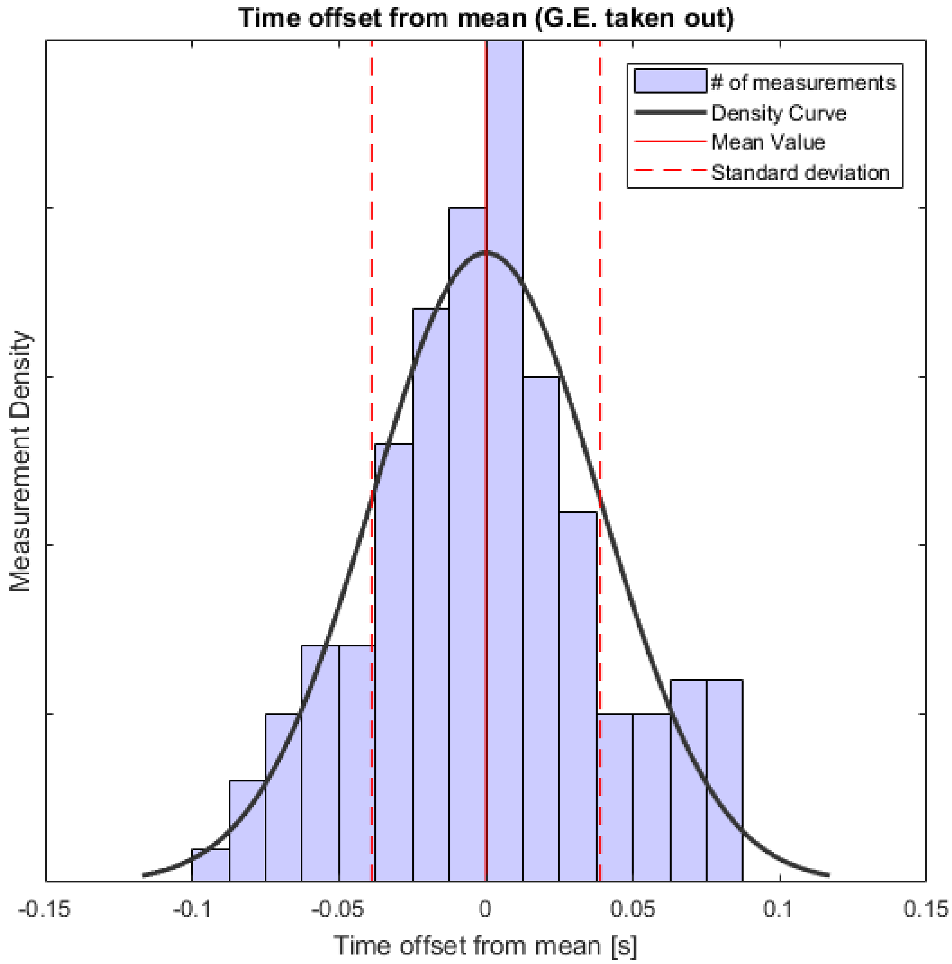

Figure 4.3: Time offset from mean in datasets

The values in Figure 4.3 are distributed with a normal distribution:

$$TimeOffsetFromMean \sim N(\mu, \sigma^2), \qquad \text{where } \mu = 0 \text{ and } \sigma = 0.039s \tag{4.2}$$

Using the *Time Offset From Mean* value **instead of**

$$TimeOffset \sim N(AverageTimeOffset, \sigma^2) \tag{4.3}$$

is ideal because this allows the use of both datasets at once. This is the case because datasets from different total station have a different 'Average Time Offset' (Section 4.1).

Similarly to the accuracy perpendicular to the trajectory, these values can also be represented by their absolute value, subsequently resulting in a half normal distribution. Once again, this can be shown in a boxplot.

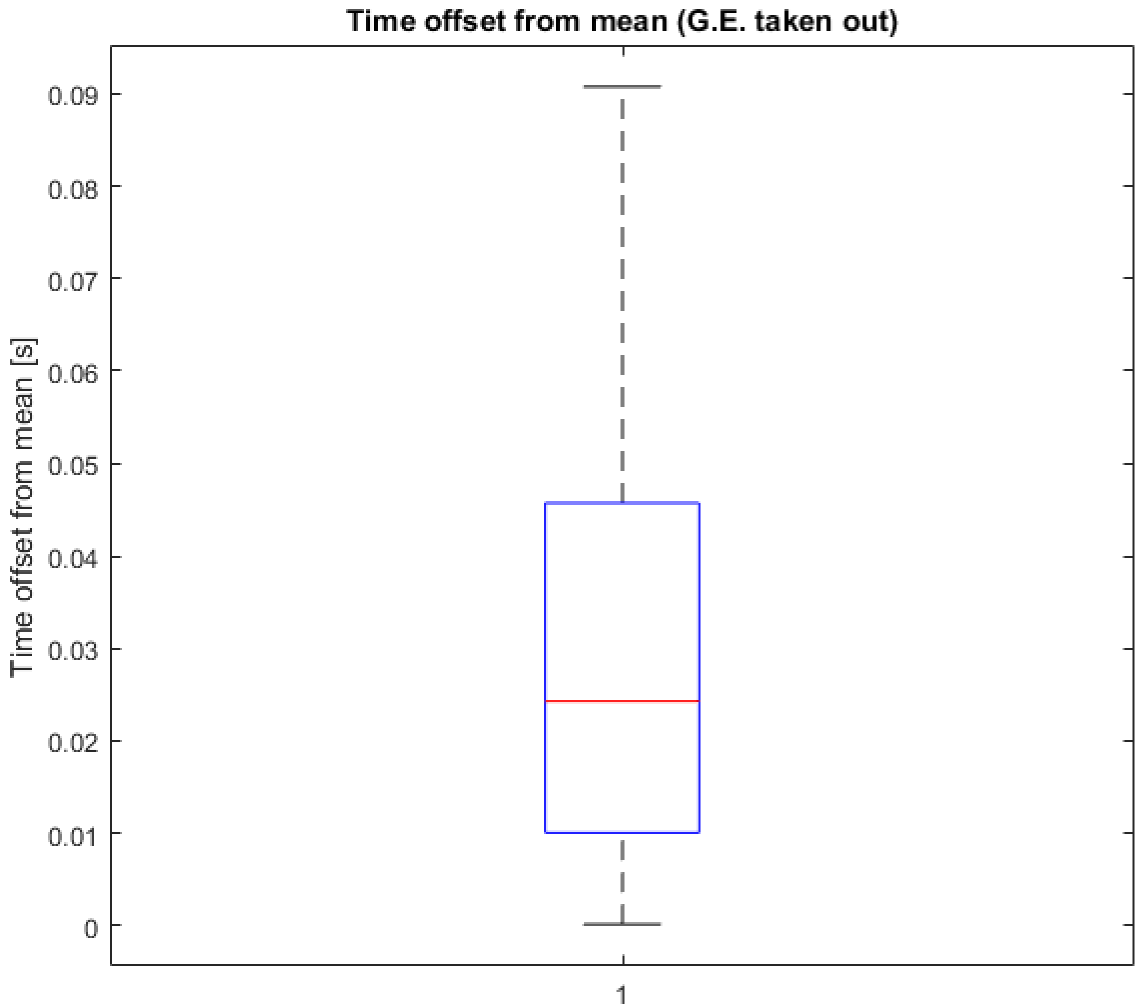

Figure 4.4: Boxplot Time Offset from mean (absolute values)

The values in Figure 4.3 and Figure 4.4 are given in a time reference, seconds [s], not in meters [m]. The way the values are obtained during interpolation, finding a time reference value makes more sense. To obtain the positional accuracy in meters, the velocity of the vehicle will need to be considered. A time offset from the expected mean value will have a larger effect depending on the current velocity. The positional offset and the time offset from mean are linearly correlated.

$$PositionalOffset = TimeOffsetFromMean * Velocity \tag{4.4}$$

E.g.: If a point has a time offset of $0.05s$ from the mean value at a velocity of $2m/s$, the expected positional offset is $10cm$. With the same time offset from mean at $10m/s$, that positional offset is half a meter.

The average velocity during the experiment was about $4m/s$, but values do vary as the vehicle drove around the testing area.

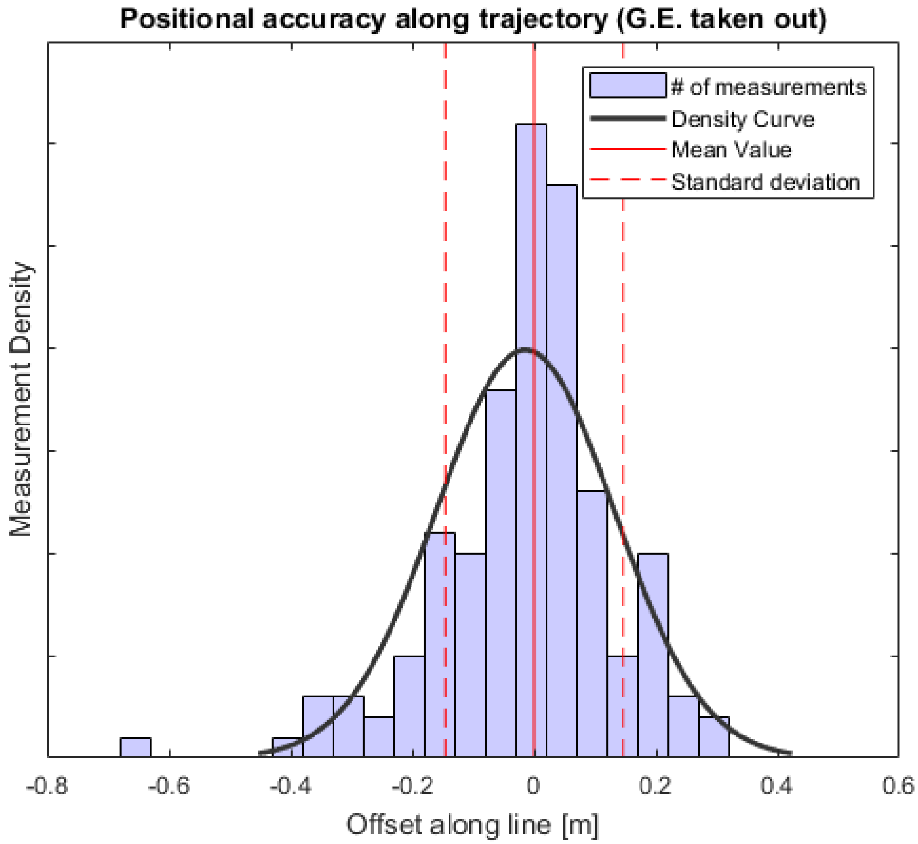

Figure 4.5: Positional Accuracy along travel trajectory

The values in Figure 4.5 appear to be distributed with a normal distribution:

$$OffsetAlongLine \sim N(\mu, \sigma^2), \qquad \text{where } \mu = 0 \text{ and } \sigma = 0.146m \tag{4.5}$$

4.3 2D Accuracy

When considering both directions, all recorded points can be mapped into a 2D plane. Accuracy perpendicular and along travel trajectory can be considered as distances from their expected value. Analyzing a 2D plot of two independent normal distributed values can be done with an error ellipse.

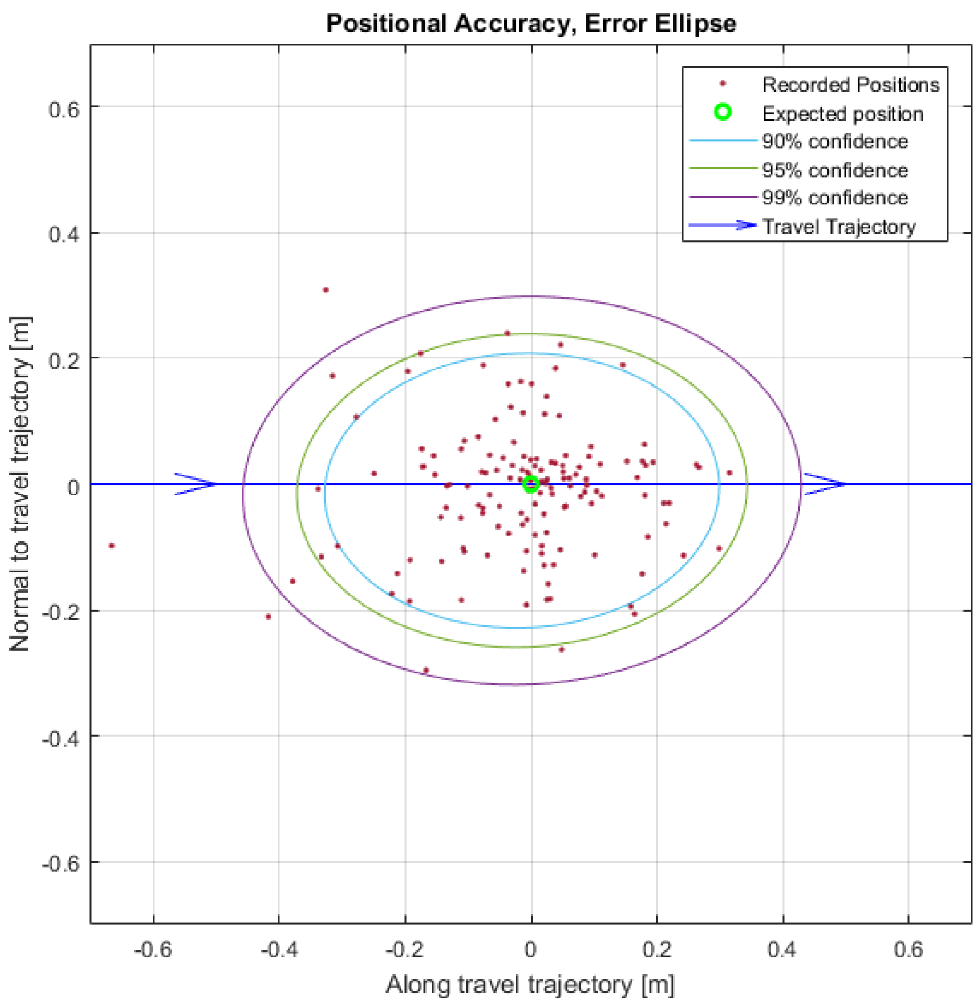

Figure 4.6: Error Ellipse for Positional Accuracy

Figure 4.6 shows an error ellipse with the recorded points, their expected position, the travel trajectory and several confidence intervals. The distribution of the sum of two squared normally distributed values is known as a Chi-Square distribution with two degrees of freedom[27]. The magnitude of the ellipse axes is dependant on the variance for the corresponding value.

It is interesting to note that the accuracy along the travel trajectory is worse than perpendicular to the travel trajectory. Several reasons can be discussed:

- The measurement method for time interpolation and recording points in this direction is less accurate

  and/or

- The GNSS accuracy when in motion is worse in the direction of movement.

4.4 Other analysis

It might be of interest to look at the raw data given value for the coordinate quality. The GNSS Receiver gives outputs about how accurate the values are based on perceived GNSS error values. The values are given as raw data for every GNSS point in the output file. Mapping the 2D Coordinate Quality (CQ) against each respective corresponding calculated accuracy value (euclidean distance between the recorded point and expected point) can show possible correlations between the two values.

Correlation in Matlab is based on the built-in function *'corrcoef(A,B)'*. The function calculates the Pearson correlation[28] between N lists of values, returning a NxN correlation matrix. In this case, there are only 2 lists, and the lists will be the raw data 2D coordinate quality and the calculated 2D accuracy value. A Pearson correlation is a linear correlation coefficient based on the formula

$$\rho(A,B) = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{A_i-\mu_A}{\sigma_A}\right)\left(\frac{B_i-\mu_B}{\sigma_B}\right) \tag{4.6}$$

which can also be written as

$$\rho(A,B) = \frac{cov(A,B)}{\sigma_A\sigma_B} \tag{4.7}$$

The function will return a 2x2 matrix

$$R = \begin{bmatrix}\rho(A,A) & \rho(A,B)\\ \rho(B,A) & \rho(B,B)\end{bmatrix} \tag{4.8}$$

Since a list of data points is perfectly correlated to itself $\rightarrow \rho(A,A) = \rho(B,B) = 1$
In addition, since $corrcoef(A,B) = corrcoef(B,A) \rightarrow \rho(A,B) = \rho(B,A)$
The value of this correlation will always be in the range of $[-1,1]$, where 1 is strongly positive correlated and -1 is strongly negative correlated. Closer values to 0 indicate less correlation in the dataset. Empirically, a value of $\pm 0,3$ is considered to show slight correlation[29].

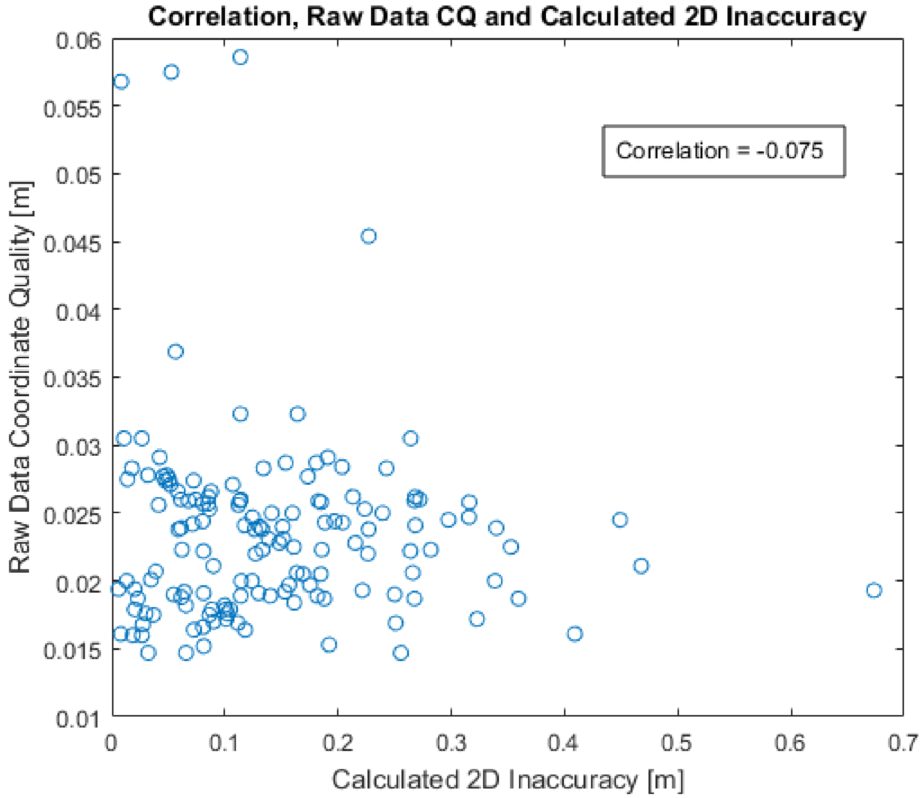

Figure 4.7: Correlation between given and calculated accuracy values

The data shown in Figure 4.7 show a minimal negative correlation value, far below the empirical threshold for being considered correlated. Subsequently, it has to be concluded that the data is not correlated. A correlation would have shown that 'bad' points are found to be inaccurate trough interpolation as well, but this is not the case.

The figure also shows the disparity in numerical values between the calculated and the raw data accuracy information.

4.5 Discussion

Since the empirical 2D coordinate quality values for total stations ($\approx 0.005 meters$), and for GNSS receivers ($\approx 0.02 meters$), are much lower than the achieved accuracies, there is reason to discuss possible inaccuracy sources.

There are a number of inaccuracy sources that might cause the values to be larger than their empirical values. An existing limitation in this method of verifying positional accuracies, which is due to the total station hardware. A number of factors could influence the values gathered in the interpolation, and they are quite difficult to distinguish.

There could be issues with internal hardware timing. Usually, exactly **when** a data point is recorded is not of much importance in typical TS surveying jobs (at least not to the closest 1/100th of a second). The fact that the raw data time values in both datasets are only given to the closest 1/100th of a second might also be problematic itself since this implies an internal time inaccuracy of $\pm 0.005 seconds$ right from the beginning. That might not seem like a lot, but already at the average vehicle speed in the experiment ($\approx 4m/s$), this equates to $2cm$, which is about 14% of the standard deviation found. In addition, a $2cm$ accuracy is not unusual raw data coordinate quality for phase measurements to begin with, so this is an obvious problem. Since inaccuracy values are much bigger than this even over a large number of data points, there is clearly something else going on as well. Another possibility is simply the timing of the writing of a data point into memory inside the internal system itself. How accurate that internal time really is, and if it is possible to further independently investigate this issue, might be an important extension if the experiment were to be repeated. GNSS receivers have more focus on time-taking, and should therefore empirically be more reliable in this regard.

These timing issues should only affect the along trajectory value, but the perpendicular value (*'distance to line'*) is also significantly higher than empirically expected.

This could be caused by the frequency of TS measurements. During turns, the distance to line value would naturally be higher than normal. The higher the turn-rate, and lower the frequency, the worse this effect would be. If the vehicle was driving in a circle, values should be easily recognizable because all *'distance to line'* values would skew towards one side. Since the path of the vehicle in the experiment follows more of a lemniscate (figure-eight, see Figure 3.9), this is not that obvious in the data. Generally, the data point frequency in the TS dataset is quite good (5-6Hz, while GNSS has 1Hz), and values are still very high even for the turn-rates of a slowly driving vehicle.

The transformation of the TS datapoints (section 3.2) from their local system to the *EUREF89 UTM32N* coordinate system could possibly add some uncertainties to the accuracy of the TS data, resulting in worse accuracy in the comparison as well. Leica Infinity returns residuals as measurements of how well fitted the transformation appears to be. The transformation is based on common averaged control points, and these points are stationary, and by extension are expected to have a better accuracy. This means that given residuals might not validly reflect the certainty of the transformation. Residuals of the transformation are $\approx 2mm$, and if the transformation had caused any big offsets, the data should not have been normal distributed around an expected point like in Figure 4.6.

It is possible that some of the larger values appear as a result of an undetected gross error (as discussed in section 3.2.4). This might happen when an unaddressed special case (like a geometry case) causes an erroneous 'random' value, but that value happens to be just good enough to not be detected by gross error detection. In the case of individual points, this might be the case, but overall, the values are far too evenly distributed for this to be the only reason for inaccuracies.

In general, the accuracy values are likely caused as cumulative errors by a multitude of sources like total stations, GNSS receiver, transformation, interpolation, time-taking, hardware and vehicle path.

There are several possible improvements to be made. A lot more data could have been collected in a larger scale experiment and might have more insight into the inaccuracy sources. Although time reference synchronized systems might have given more accurate results, the tests showed the importance of accurate time-keeping when comparing systems in real-time. In a potential C-ITS situation, similar timing issues will appear and a solution like the interpolation might be necessary in the case of independent systems.

The interpolation script itself is relatively versatile in terms of data input and further generalization for future similar applications involving unsynchronized datasets.

Chapter 5

Summary

This chapter will summarize what has been done and which results were achieved. Each original objective is addressed, and recommendations for further work is suggested.

5.1 Summary and Conclusions

For this thesis an experiment was conducted (section 3.1) to collect data for further analysis of positional accuracies. The findings of that analysis suggest that accuracies in reality are quite a bit worse than empirically given by these positioning methods. How accurate that assessment is is discussed in detail in section 4.5.

In reference to the Master's thesis' objectives as defined in section 1.2:

The correlation between time and positional accuracy in unsynchronized datasets has been explored in great detail in section 4.2.2. The linear connection, as well as importance of accurate time measurements has been demonstrated.

GNSS accuracies have been analyzed, and it seems as if the total station data had a hard time verifying the results. When comparing the values to the interpolated values to the TS dataset, the positional offset are quite a bit bigger than the raw data (Figure 4.7). It has to be concluded that either the raw data is not reliable, or that verifying raw data with an unsynchronized dataset is not viable.

Accuracy values that were found were captured in motion, and indicate that inaccuracy values are significantly larger than their expected stationary counterparts (at least found with this method). The data also shows that being in motion gives a slightly worse accuracy in the direction of motion (Figure 4.6).

An interpolation script was successfully developed and used to analyze the data. The development is discussed in detail in section 3.2. Outputs of the script are time offset values as well as positional accuracy values. The script is robust enough to handle most special cases, making the preparation of raw data relatively easy. The interpolation part of the developed code is given in Appendix A.

Recommendations for Further Work

- **Short-term**

  There are some more potential raw data values that could be investigated for correlations, such as heights, curvature and measurement frequency. Height could be introduced into the interpolation and create a 3D Accuracy model for all recorded points. Curvature is indirectly given in the raw data because the recorded points for a line that could be fitted to a curvature value. Accuracy values changing significantly because the vehicle is turning seem implausible, but could be investigated further with the dataset. The frequency of the TS datapoints changes slightly trough-out the dataset, and a correlation to the time offset could be investigated.

- **Medium-term**

  A new, larger scale data collection in more controlled environments could be conducted. This should include synchronized time reference systems. Measurements for even larger top speeds, and investigations of possible correlations with speed beyond the obvious ($TimeInaccuracy * Speed = PositionalInaccuracy$) could be introduced. The equipment that was used in this experiment was good, but obviously not meant specifically for this kind of operation. Total Stations that focus on super-precise time taking could be used if the time interpolation is used as a method for verification.

  Another experiment could also investigate the same kind of accuracies for code measurements.

- **Long-term**

  Adapting the knowledge gained in this thesis and follow-up experiments into autonomous vehicles as a guideline/buffer for calculation of C-ITS maneuvers.

Appendix A

Interpolation Script

The interpolation script that is run for every recorded GNSS point:

```
function [newtime,disttoline,newpoint,pos]=interpolation(tslist,gnsspoint,startpos)
%Returning 'pos' value to have a new 'startpos' value for next iteration
    pointlist = str2double(tslist(:,1:2));
    if startpos == 1 %Avoiding special case that would crash code
        startpos = 2;
    end
    initd = pdist([gnsspoint;pointlist(startpos-1,:)],'euclidean');
    for i = startpos:length(pointlist)
        X = [gnsspoint;pointlist(i,:)];
        d = pdist(X,'euclidean');
        if d >= initd
            pos = i - 1;
            break
        else
            initd = d;
        end
        pos = i;
    end
    if pos ~= 1 %Avoiding special case that would crash code
        A = pointlist(pos-1,:);
        B = pointlist(pos,:);
    else
        A = pointlist(pos,:);
        B = pointlist(pos+1,:);
    end
    EigenAB=(B-A)./norm(B-A);
    EigengnssB=(B-gnsspoint)./norm(B-gnsspoint);
    diff1 = (sum(abs(EigenAB-EigengnssB)));
    diff2 = Inf;
    if pos ~= length(pointlist) %Avoiding special case that would crash code
        C = pointlist(pos+1,:);
```

```
        EigenCB=(B-C)./norm(B-C);
        diff2 = (sum(abs(EigenCB-EigengnssB)));
    end
    if diff1 <= diff2
        side = 1; %The GNSS point is behind its closest TS point
        dist = point_to_line([gnsspoint(1),gnsspoint(2),0], [A(1),A(2),0], [B(1),B(2),0]);
        v = EigenAB;
        R = [cosd(90) -sind(90); sind(90) cosd(90)];
        vR = v*R;
        movev = dist*vR;
        if norm(gnsspoint+movev-B) <= norm(gnsspoint-B)
            newpoint = gnsspoint+movev;
        else
            newpoint = gnsspoint-movev;
        end
        disttoline = pdist([0,0;movev],'euclidean');
    else
        side = 2; %The GNSS point is in front og its closest TS point
        dist = point_to_line([gnsspoint(1),gnsspoint(2),0], [C(1),C(2),0], [B(1),B(2),0]);
        v = EigenCB;
        R = [cosd(90) -sind(90); sind(90) cosd(90)];
        vR = v*R;
        movev = dist*vR;
        if norm(gnsspoint+movev-B) <= norm(gnsspoint-B)
            newpoint = gnsspoint+movev;
        else
            newpoint = gnsspoint-movev;
        end
        disttoline = pdist([0,0;movev],'euclidean');
    end
    d1 = pdist([newpoint;pointlist(pos,:)],'euclidean');
    if side == 1 %Looking backwards
        d2 = pdist([pointlist(pos,:);pointlist(pos-1,:)],'euclidean');
        perc = d1/d2;
        currenttime1 = second(datetime(list(pos,3)));
        currenttime2 = second(datetime(list(pos-1,3)));
        if pos ~= i
            newtime = currenttime1 - abs(currenttime1-currenttime2)*perc;
        else %possibility the GNSS point is after last TS measurement
            newtime = currenttime1 + abs(currenttime1-currenttime2)*perc;
        end
    else %Looking forwards
        d2 = pdist([pointlist(pos,:);pointlist(pos+1,:)],'euclidean');
        perc = d1/d2;
        currenttime1 = second(datetime(list(pos,3)));
        currenttime2 = second(datetime(list(pos+1,3)));
        newtime = currenttime1 + abs(currenttime1-currenttime2)*perc;
    end
end

function d = point_to_line(pt, v1, v2)
      a = v1 - v2;
      b = pt - v2;
      d = norm(cross(a,b)) / norm(a);
end
```

Bibliography

[1] D. Lange, "Intelligente transportsystemer og analyse av posisjonsdata basert på statens vegvesens gnss-datasett", *TBA4560, Specialization Project*, 2017.

[2] L. Geosystems. (2018). Leica smartstation - a total station with integrated gnss, [Online]. Available: https://leica-geosystems.com/products/gnss-systems/systems/leica-smartstation (visited on 07/06/2018).

[3] SogeLink. (2018). Land2map, [Online]. Available: https://www.sogelink.com/land2map (visited on 07/06/2018).

[4] G. D. Jenssen. (2017). Automatisering av transportsystemet, [Online]. Available: https://www.vegvesen.no/_attachment/2044010/binary/1213406?fast_title=Automatisering+av+transportsystemet+i+et+historisk+perspektiv.pdf (visited on 05/28/2018).

[5] D. Etherington. (2018). Elon musk's self-driving strategy still doesn't include lidar, [Online]. Available: http://tcrn.ch/2E9jI9t (visited on 06/06/2018).

[6] SAFESPOT. (2010). Safespot integrated project, [Online]. Available: http://www.safespot-eu.org/ (visited on 06/07/2018).

[7] (2018). Its-g5 is ready to roll, [Online]. Available: http://www.itsg5-ready-to-roll.eu/ (visited on 05/27/2018).

[8] F. Tronchetti, Ed., *Beidou's global navigation satellite services: regulatory challenges and way forward*, 68th, Abstract: IAC-17,E7,4,5, IAC (International Astronautical Congress), Adelaide, Australia, Sep. 2017.

[9] ESA. (2017). What is galileo?, [Online]. Available: `http://www.esa.int/Our_Activities/Navigation/Galileo/What_is_Galileo` (visited on 05/26/2018).

[10] W. E. Hofmann-Wellenhof B. Lichtenegger H., *GNSS - Global Navigation Satellite Systems*, 1st ed. SpringerWienNewYork, 2008.

[11] T. Skogseth and D. Norberg, *Grunnleggende landmåling*, 3rd. Gyldendal Norsk Forlag, 2014, ISBN: 978-82-05-44934-3.

[12] T. Skogseth, *Lecture notes, teoretisk geomatikk*, NTNU Trondheim, 2016.

[13] Kartverket. (2017). Cpos, [Online]. Available: `https://www.kartverket.no/posisjonstjenester/cpos/` (visited on 05/10/2018).

[14] ——, (2017). Referanserammer for norge, [Online]. Available: `https://www.kartverket.no/Kunnskap/Kart-og-kartlegging/Referanseramme/Referanserammer-for-Noreg/` (visited on 05/28/2018).

[15] ——, (2016). To nye basestasjoner, [Online]. Available: `https://www.kartverket.no/om-kartverket/Nyheter/to-nye-operative-basestasjoner/` (visited on 07/06/2018).

[16] B. Kavanagh and T. Mastin, *Surveying: Principles and Applications*, 9th ed. Pearson, 2013, ISBN: 978-01-37-00940-4.

[17] Trimble. (2008). Trimble survey controller software: Purpose, advantage, and objective of resection setup, [Online]. Available: `http://trl.trimble.com/docushare/dsweb/Get/Document-406432/SC_Purpose` (visited on 05/26/2018).

[18] (2018). Open street maps, [Online]. Available: `https://www.openstreetmap.org/#map=18/63.41422/10.40704` (visited on 06/01/2018).

[19] Google. (2017). Google streetview, [Online]. Available: https://www.google.com/maps/@63.4140393,10.4092539,3a,75y,314.1h,89.78t/data=!3m6!1e1!3m4!1sOkpp2azsh5DW2OVucNK3Cw!2e0!7i13312!8i6656 (visited on 06/10/2018).

[20] Leica. (2018). Leica geosystems, [Online]. Available: https://leica-geosystems.com/ (visited on 05/31/2018).

[21] LeicaGeosystems. (2018). Tps1200 user manual, [Online]. Available: http://www.surveyequipment.com/PDFs/TPS1200_User_Manual.pdf (visited on 05/26/2018).

[22] Leica. (2018). Leica infinity, [Online]. Available: https://leica-geosystems.com/products/total-stations/software/leica-infinity (visited on 05/25/2018).

[23] MathWorks. (2018). Matlab, [Online]. Available: https://se.mathworks.com/products/matlab.html (visited on 05/24/2018).

[24] PostGIS. (2018). St_distance, [Online]. Available: https://postgis.net/docs/ST_Distance.html (visited on 05/26/2018).

[25] F. C. Leone, L. S. Nelson, and R. B. Nottingham, "The folded normal distribution", *Technometrics*, vol. 3, no. 4, pp. 543–550, 1961, ISSN: 00401706. [Online]. Available: http://www.jstor.org/stable/1266560.

[26] MathWorks. (2018). Matlab boxplot documentation, [Online]. Available: https://se.mathworks.com/help/stats/boxplot.html (visited on 05/31/2018).

[27] R. Johnson and D. Wichern, *Applied Multivariate Statistical Analysis*, 6th ed. Pearson, 2007, ISBN: 978-0-13-187715-3.

[28] MathWorks. (2018). Matlab correlation documenation, [Online]. Available: https://se.mathworks.com/help/matlab/ref/corrcoef.html (visited on 06/01/2018).

[29] D. Hinkle, W. Wiersma, and S.G.Jurs, *Appliead Statistics for Behavioral Sciences*, 5th ed. Houghton Mifflin, 2003, ISBN: 9780618124060.