NTNU
Norwegian University of
Science and Technology

Increasing the Accuracy of Positioning in Mobile Mapping Systems

A method supported by Simultaneous
Localization And Mapping

Marianne Løvås

Master of Science in Engineering and ICT

Submission date:	July 2017
Supervisor:	Hossein Nahavandchi, IBM
Co-supervisor:	Helén Rost, TerraTec Sweden AB
	Narve Schipper Kjørsvik, TerraTec AS

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

Master thesis

(TBA4925 - Geomatics, Master's Thesis)

Spring 2017
for

Marianne Løvås

Increasing the Accuracy of Positioning in Mobile Mapping Systems
A method supported by Simultaneous Localization And Mapping

BACKGROUND

Development of Mobile Mapping System (MMS) began in the late 1980s and is constantly growing. Thanks to continuous developments in both scanning and positioning technologies, Mobile Mapping Systems are gaining more and more importance in many applications. Different solutions are available on the market with different technical specifications. Terrestrial MMS technology goes back to the 90's when the first experiments showed the potential of mobile mapping for GIS applications. Ultimately, this gave birth to commercial operating systems used for example for 3D mapping of road, railways, city and coastal areas. There are, nowadays, different commercial MMS technologies, showing the best example of sensor integration for optimal acquisition of 3D georeferenced spatial data. One example is the Optech Lynx Mobile Mapping System operated by TerraTec.

The georeferencing of data from the remote sensing system, e.g. a laser point cloud, is based on the position and orientation of the platform of the Mobile Mapping System. The navigation sensors collect data about the platform's position and orientation. A Kalman filter can be used to utilize this information and estimate a trajectory of the platform. The trajectory is the path described by the platform's movements in space.

The estimation of the trajectory based on the sensor data works well and can give results with an accuracy of a few centimetres in areas with GNSS signals from multiple satellites. In areas with high buildings, trees or other obstacles, the loss of GNSS signals can, however, lead to an accuracy of the trajectory that is significantly worse.

TASK DESCRIPTION

In large mobile mapping projects, there are high demands on both accuracy of the data and efficiency in data collection and processing. Estimation of trajectory is performed using a well-known problem in robotic mapping called “Simultaneous Localization And Mapping (SLAM)”. SLAM is the problem of creating a map of an unknown environment at the same time as positioning the trajectory in this environment. The purpose of this thesis is to investigate the efficiency of SLAM in the post-processing of mobile mapping data.

The thesis should contain a description of the theory of mobile mapping systems, including the hardware and software packages used. As a result of the thesis, based on the collected data in the field, the student should comment on the achievable accuracy improvement by introducing SLAM in the post-processing of mobile mapping data.

Potential methods for re-calculating the trajectory of the platform (Optech Lynx Mobile Mapping System in this study) in the case of GNSS satellite signal loss in mobile mapping system should be investigated.

ADMINISTRATIVE/GUIDANCE

The work on the Master Thesis starts on February 13, 2017

The thesis report as described above shall be submitted digitally in DAIM at the latest at July 10, 2017

External supervisors:
Helén Rost, TerraTec Sweden AB
Narve Schipper Kjørsvik, TerraTec AS

Supervisor at NTNU and professor in charge:
Hossein Nahavandchi

Trondheim, February 13, 2017

Abstract

In areas where GNSS satellite signals become weak or unavailable a period of time, accuracy of the positioning of mobile mapping systems based on aided inertial navigation can be drastically reduced. This can be problematic in, for instance, city areas with high buildings, forests and tunnels. An increased number of terrestrial land surveyed points can be necessary to get a sufficiently good accuracy of the estimated trajectories in these areas, which makes the process more time and work consuming.

In mobile mapping, laser scanners are typically used to gather information about the surroundings, but they can also be used to find information about relative movement of the mobile mapping system. Taking advantage of Simultaneous Localization And Mapping (SLAM), the observations from the laser scanner can be used to aid inertial navigation when calculating the trajectory. Including SLAM in re-calculation of the trajectory can be done to improve the positioning accuracy of it. The improved trajectory can, in turn, be used to improve accuracy of the point cloud.

Potential methods for including SLAM have been developed and tested in this thesis. The first method presented is used to improve global consistency of the trajectory by the use of back-end SLAM. This is done by using loop closure events. The second method uses incremental observations to improve local consistency of the trajectory by the use of front-end SLAM.

A software package has been developed for transforming point cloud observations from TerraMatch to information that can be used in the SLAM algorithms as part of the trajectory calculation in TerraPos. A relative point cloud observation in TerraMatch is linked to the other point cloud observations of the same feature. The developed methods use this information to perform data association in SLAM.

The methods have been tested in GNSS denied areas in the point cloud. Both the presented methods show an improved accuracy and precision of the trajectory by the use of SLAM to re-calculate it. Tests of the method using back-end SLAM show that the method has the potential to improve the trajectory when a part of the trajectory with low precision overlaps a part with high precision. Identification and removal of systematic errors was found to be important in the incremental observations used for the front-end SLAM method.

Sammendrag

Støttet treghetsnavigasjon kan brukes til posisjonering av bilbåren laserskanning (mobile mapping). I områder der GNSS-signaler blir utilgjengelige, kan nøyaktigheten på posisjonering av trajektorien til bilen bli betraktelig redusert. Dette kan være problematisk i for eksempel byområder med høye bygninger, skogområder og tunneler. I slike områder kan det være nødvendig å øke antall landmålte punkter for å få tilstrekkelig god nøyaktighet på posisjoneringen, noe som gjør prosessen mer tid- og arbeidskrevende.

I bilbåren laserskanning brukes laserskannerne typisk til å samle inn informasjon om omgivelsene, men de kan også brukes til å finne informasjon om den relative bevegelsen til bilen. Ved å benytte SLAM (Simultaneous Localization And Mapping), kan observasjoner fra laserskanningen brukes til å støtte treghetsnavigasjonen. Bruk av SLAM i beregning av trajektorien til bilen kan forbedre nøyaktigheten til trajektorien. Den forbedrede trajektorien kan deretter brukes til å forbedre nøyaktigheten av punktskyen fra laserskanningen.

I denne masteroppgaven er to potensielle metoder for å utnytte SLAM i etterprosessering av data fra bilbåren laserskanning utviklet og testet. Den første metoden benytter seg av back-end SLAM og såkalte «loop closure events», der trajektorien krysser seg selv, for å forbedre den globale nøyaktigheten til trajektorien. Den andre metoden utnytter inkrementelle observasjoner i front-end SLAM for å forbedre den lokale nøyaktigheten til trajektorien.

Et program er utviklet for å transformere punktskyobservasjoner fra TerraMatch til informasjon som kan brukes i SLAM-algoritmene brukt for å beregne trajektorien til bilen i TerraPos. En relativ punktskyobservasjon i TerraMatch er knyttet til de andre punktskyobservasjonene av samme objekt. Denne informasjonen brukes til å gjøre dataassosiering i SLAM.

Metodene er testet i deler av punktskyen der GNSS observasjoner ikke er tilgjengelig. Begge de presenterte metodene viser en forbedret nøyaktighet og presisjon på trajektorien til bilen ved bruk av SLAM for å beregne trajektorien. Testing viser at når en del av trajektorien som har lav presisjon overlapper en del som har høy presisjon, har metoden som bruker back-end SLAM potensiale til å forbedre nøyaktigheten til trajektorien. Identifisering og fjerning av systematiske feil fra de inkrementelle observasjonene seg å være viktig for den metoden som bruker front-end SLAM.

Preface

This thesis is a completion of a Master of Science in Engineering and ICT, with specialization in geomatics, at the Norwegian University of Science and Technology (NTNU). The work for this thesis is done in co-operation with TerraTec.

I would like to express my sincerest gratitude towards my dedicated supervisors at TerraTec, Narve Schipper Kjørsvik and Helén Rost. Their guidance, support and great efforts have been invaluable throughout the work for this thesis, and their enthusiasm has been contagious.

A great thanks is also handed to my supervisor at NTNU, Hossein Nahavandchi.

Appreciations are extended to the people at TerraTec in general, for providing me with the topic of research, data collections, software and an office spot.

Finally, I would like to thank my dear family and friends, for all their love, support and attempts to understand my priorities towards my thesis.
And to my parents especially, I owe you both a great share of my success.

Marianne Løvås
Trondheim, July 10th 2017

Table of Contents

List of Figures

List of Tables

Abbreviations

DCM Directional Cosine Matrix. 7, 8

DGNSS Differencial GNSS. 10, 31

DMI Distance Measurement Indicator. 12

ECEF Earth Centered Earth Fixed. 6

ECI Earth Centered Inertial. 6

EKF Extended Kalman Filter. 18, 24, 25, 35, 62

GNSS Global Navigation Satellite System. 1, 2, 6, 8–10, 12, 16, 18, 26, 27, 29, 31, 41–45, 47, 48, 50, 51, 53, 54, 57, 61–63

GPS Global Positioning System. 8, 10

IMU Inertial Measurement Unit. 6–8, 12

INS Inertial Navigation System. 1, 6–8, 12, 13, 18, 21, 25–27, 31, 41, 47, 51, 60, 61, 64

LIDAR Light Detection and Ranging. 13, 27

MMS Mobile Mapping System. 1, 27–29, 41

NED North East Down. 7

PDOP Position Dilution Of Precision. 9, 29

SLAM Simultaneous Localization And Mapping. 2, 3, 5, 21–25, 27, 30, 31, 33, 35, 37, 41, 42, 48, 51, 53, 54, 60–64

TOW Time of Week. 10, 36, 43, 45, 46, 51, 54

Chapter 1

Introduction

1.1 Mobile Mapping

Mobile Mapping is the process of collecting geospatial data from a vehicle in motion. A Mobile Mapping System (MMS) is composed of a navigation system, giving a position for each period of time, and a remote sensing system, collecting information about the system's environment. Usually, the navigation system includes an Inertial Navigation System (INS), Global Navigation Satellite System (GNSS) and an odometer. The remote sensing system can consist of advanced cameras and laser scanners.

Development of Mobile Mapping System (MMS) began in the late 1980s and is constantly growing. Thanks to continuous developments in both scanning and positioning technologies, Mobile Mapping Systems are gaining more and more importance in many application fields. Puente et al. [2013] review different solutions available on the market with a comparison of their technical specifications. Terrestrial MMS technology goes back to the 1990s when the first experiments showed the potential of mobile mapping for GIS applications [El-Sheimy, 1996]. Ultimately, this gave birth to commercial systems used for example for 3D mapping of road, railways, city and coastal areas. Nowadays, there are different commercial MMS technologies, showing the best example of sensor integration for optimal acquisition of 3D georeferenced spatial data. One example is the Optech Lynx Mobile Mapping System operated by TerraTec.

Georeferencing of data from the remote sensing system, e.g. a laser point cloud, is based on position and orientation of the platform (vehicle) of the Mobile Mapping System. Navigation sensors collect data about the platform's position and orientation. A Kalman filter can be used to utilize this information to estimate a trajectory of the platform. The trajectory is the path described by the platform's movements in space. TerraPos can be used to calculate the trajectory by the use of a Kalman filter. TerraPos is a software system developed by the company TerraTec AS.

Estimation of trajectory based on sensor data works well and can give results with an accuracy of a few centimeters in areas with GNSS signals from multiple satellites. However, in areas with high buildings, trees or other obstacles, the loss of GNSS signals can lead to an accuracy of the trajectory that is significantly worse.

The point cloud is generated based on the calculated trajectory and raw laser data. This means that errors and drift in the trajectory in many cases can be seen by shift and rotations in overlapping parts of the point cloud. Tie points are point features than can be identified in multiple overlapping parts of the point cloud. Point cloud observations can be used to measure the shift and rotation of overlapping parts of the point cloud. Point cloud observations can be found by the software packages TerraScan and TerraMatch developed by the company Terrasolid OY. Adjustments based on the errors estimated from the point cloud observations can also be made in these software packages.

Tie point observations are defined by a vector from the trajectory to the tie points at time of scanning. Tie point observations can be introduced in the Kalman filter when calculating the trajectory in TerraPos. Including tie point observations can reduce drift and increase accuracy of the trajectory. When doing this, the estimation of trajectory corresponds with the Simultaneous Localization And Mapping (SLAM) problem from robotic mapping. SLAM is the problem of creating a map of an unknown environment while finding the trajectory in this environment.

Many SLAM algorithms focus on estimating discrete map features, so-called landmarks. In landmark-based SLAM it is essential for landmark observations to be linked to the correct landmark. Observations of the same landmark in the point cloud need to be of the same position in the scanned terrain. This is called the *data association problem* in SLAM. Point cloud observations in TerraMatch are linked to a feature in the terrain. Thus, the data association can be done by the information from point cloud observations in TerraMatch.

Optimal landmark-based SLAM algorithms are computationally demanding, particularly for a high dimensional system state (i.e., with many landmarks). Efficient use depends on a careful selection of landmarks.

The SLAM functionality in TerraPos can utilize tie point observations and incremental point cloud observations in a re-calculation of the trajectory. Thus, information collected by the laser scanner can be used to aid localization of the platform and increase accuracy of the trajectory.

1.2 Research Objective

In large mobile mapping projects, there is high demand for data accuracy and for efficient data collection and data processing. A previous study [Løvås, 2016] indicated that re-calculating the trajectory using SLAM algorithms in TerraPos can increase accuracy of the trajectory. This, in turn, can improve accuracy of the point cloud.

An efficient method for using SLAM in post-processing of mobile mapping data is needed for it to be usable on a large scale. For it to be efficient, it needs to limit both computational cost and the need for man hours in processing. The objective of this thesis is to find potential methods for re-calculating the trajectory in TerraPos using information from the point cloud found in TerraMatch.

Identification of bottlenecks and possibilities for integration of TerraPos and TerraMatch are investigated. Point cloud observations will be found in TerraMatch. Information from TerraMatch will be transformed to information in a format that can be utilized in TerraPos to re-calculate the trajectory. Lastly, a method for adjusting the point cloud by the re-calculated trajectory is investigated.

The proposed methods will be tested with real data sets, to see whether they have potential to increase accuracy in areas where the accuracy of the original trajectory is not sufficiently high. A typical accuracy demand for mobile mapping projects is $0.05m$.

Chapter 2

Fundamentals of Mobile Mapping

This chapter will present the theoretical background of mobile mapping. Navigation theory, scanning methods and Kalman filtering used in the processing the data will be described in detail. Camera technology has not been used in this study and is not discussed.

2.1 Common Concepts

A *tie point* is a point that does not have known coordinates, but that can be identified by two or more overlapping parts of the laser point cloud. Tie points are called landmarks in the SLAM paradigm.

Overlapping parts of the point cloud are scanned at different times. Points in the point cloud are connected to a point in the trajectory by a time stamp.

A *tie point observation* is an observation of a tie point. It is defined by a vector from the trajectory to the tie point.

Point cloud observations are offsets and rotations found between overlapping parts of the point cloud.

2.2 Aided Inertial Navigation

An Inertial Navigation System (INS) is used to track position and orientation of the mobile mapping system, i.e. the system's *pose*. The INS filters data from the sensors in the Inertial Measurement Unit (IMU) resulting in position, velocity and orientation of the system relative to a starting pose.

The IMU consist of high rate sensors that typically have an update frequency of tens to a few hundred hertz. Low rate sensors like GNSS, with a typical update rate of one to ten hertz, can be utilized to aid the INS.

A problem with the INS is that errors accumulate over time. The INS continuously adds the measured change to the last calculated position. Accuracy in measurements will be reduced not only by errors in the changes from the last measurement, but also by the errors in measuring changes between positions before that. External sensors are needed to get a good starting state of the system and are helpful to limit errors and drift. GNSS measurements are often used to give position updates. An odometer mounted in a wheel can provide updates based on measured along-track distance traveled. [Dudek and Jenkin, 2008]

2.2.1 Coordinate Frames

In aided inertial navigation several different sensors are used to make a navigation solution. The sensors make observations in different coordinate frames. Hence, keeping track of the various coordinate frames is necessary for sensor fusion. To transform coordinates from one coordinate frame to another, rotation and translation between frames is needed. This section defines some relevant coordinate systems [Farrel, 2008]:

An *Inertial Frame (i-frame)* is a reference frame that is non-accelerating relative to inertial space. The origin of an inertial frame is arbitrary and the axis may point in any three mutually perpendicular directions. The Earth Centered Inertial (ECI) frame is an example of a (nearly) non-accelerating frame, with origin of the frame in the mass center of Earth.

Earth Frame (e-frame) is an Earth Centered Earth Fixed (ECEF) frame and moves and rotates with the Earth. The origin of the frame is in the Earth's center of mass. The x-axis points towards the intersection between the equator and the prime meridian, the z-axis points in the direction of the Earth's rotation axis defined by the Conventional Terrestrial Pole and the y-axis completes the right-handed orthogonal coordinate system.

Body Frame (b-frame) is a frame that is rigidly attached to the vehicle of interest. The x-axis points in the forward direction of the platform (vehicle), the z-axis points to the bottom of the platform and the y-frame completes the right-handed orthogonal coordinate system. The rotations around the x-, y- and z-axis are named *roll* (ϕ), *pitch* (θ) and *heading* (ψ) respectively. This frame can be used for relating the navigation system to other sensors on the platform, e.g. a laser scanner.

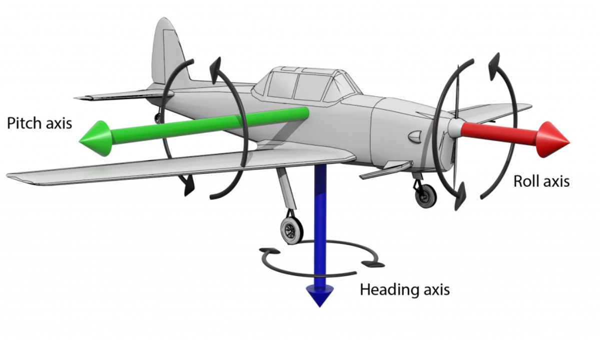

Figure 2.1: Roll, pitch and heading on aircraft [Oxford Technical Solutions, 2014]

Sensor Frames (s-frame) is an orthogonal right-hand system, realized by the triad of accelerometers and gyroscopes in the INS.

Geographic Frame (g-frame) is a North East Down (NED) frame and is defined locally on Earth. The x-axis points towards North and y-axis towards East. The z-axis points towards the center of Earth. The end result of the trajectory and point cloud from mobile mapping is often given in this coordinate system.

2.2.2 IMU

The IMU consists of three accelerometers and three gyroscopes. Together these sensors make relative observations of position and orientation of the IMU.

Observation equations for gyroscopes and accelerometer are given in the following paragraphs. Subscript denotes coordinate frame of measurement. Superscripts describe the coordinate frame the observations is given in. ω^s_{es} describes angular velocity measured in s-frame relative to e-frame, parametrized in s-frame (superscript).

Gyroscopes are used to measure angular velocity, i.e. change in orientation. A gyroscope measures angular velocity around a single axis, so three gyroscopes mounted orthogonal to each other are needed to keep track of 3-dimensional motion.

Observation equation of gyroscopes [Farrel, 2008]:

$$\omega^s_{es} = \omega^s_{is} - C^s_e \omega^e_{ei} \tag{2.1}$$

where ω^s_{es} is rotation of body relative to the Earth, ω^s_{is} is the gyroscope observation, C^s_e is a Directional Cosine Matrix (DCM) transforming from e-frame to s-frame and ω^e_{ei} is the Earth rotation.

Three accelerometers mounted orthogonal to each other are used to find kinematic force acting on the platform. Specific force, acceleration relative to free-fall, is what is measured by the accelerometers. Specific force is the combined effect of kinematic acceleration and other forces acting on it, e.g. the gravity. Thus knowledge of local conditions, which can be found by gravitation models, is required to find kinematic force. The output from accelerometers is usually velocity increments in s-frame.

Observation equation of accelerometers [Farrel, 2008]:

$$\ddot{r}^e_{es} = C^e_s f^s_{is} - 2\Omega^e_{ie}\dot{r}^e_{es} + C^e_g g^g_{es} \tag{2.2}$$

where $\ddot{r}^e_{es}$ is acceleration of the sensor in e-frame, C^e_s is a DCM transforming from s-frame to e-frame, f^s_{is} is observed specific force, $2\Omega^e_{ie}\dot{r}^e_{es}$ is the Coriolis effect and $C^e_g g^g_{es}$ is gravity transformed to e-frame.

Information typically wanted from the INS is orientation and position. Orientation is obtained by integrating the angular velocity. The INS usually integrates acceleration once, giving velocity as output. To calculate position, the velocity is integrated once more. Hence, any residual vector of acceleration will lead to a quadratic error in position. The specific force is measured in s-frame, and transformed to e-frame before integration. Thus, orientation errors will result in a rapid growth in position errors. Biases in angular velocity and acceleration leads to drift in INS estimations, which leads to estimations done by the INS alone only giving good results for a period of seconds or minutes.

External information can be used to aid the INS and reduce drift of the system. One example of this is *zero velocity updates*, where the system is standing still for a period of time. This information can be used to reduce drift in velocity of the system. When it is standing still, the only force acting on the system is gravity. Thus, a zero velocity update can be used to estimate *roll* and *pitch* and improve alignment of the IMU.

2.2.3 GNSS

Position can be found by measurements from Global Navigation Satellite System (GNSS) and can be used to aid the INS. GNSS is a term for satellite navigation systems with global coverage. There are multiple systems operating today. The original system is the American Global Positioning System (GPS). GLONASS is a Russian version of a GNSS. The Chinese BeiDou and European Galileo are both under construction and have some working satellites. They are planned to be globally operative in 2020 [BeiDou Navigation Satellite System, 2017] [ESA, 2017]. Many GNSS receivers today use signals from several systems.

A number of monitoring stations worldwide track position of the satellites, and this information is used to estimate their orbits. Precise ephemerides is post-processed information about orbit parameters for satellites. An orbit accuracy of $0.025m$ can be achieved for GPS by precise ephemerides [IGS, 2017].

GNSS satellites transmit a signal, and code and carrier phase of the signal can be used to find distance between the satellite and the receiver. Code measurements use the time from the signal was transmitted till it reaches the receiver to calculate distance between them. Measurements of carrier phase are more precise and are used for applications with higher demands on accuracy. Distance from multiple satellites to a receiver can be used to estimate position of the receiver, see figure 2.2.

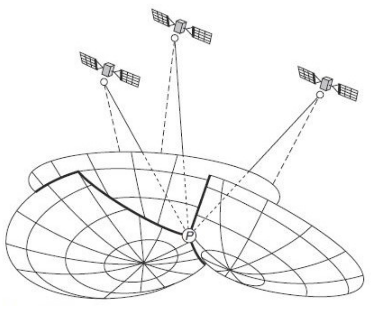

Figure 2.2: Resection of 3 satellites [Skogset and Norberg, 2014]

Clear sight from the GNSS receiver to satellites is needed. The signal cannot go through thick trees, mountains or buildings. In these environments, it can be problematic to get signal from enough satellites to be able to estimate a position of the receiver. Another problem in environments like these is *multipath*, happening when the signal does not arrive at the receiver by a direct path. The cause of this can be reflecting surfaces near the receiver, like a building or a car. See figure 2.3.

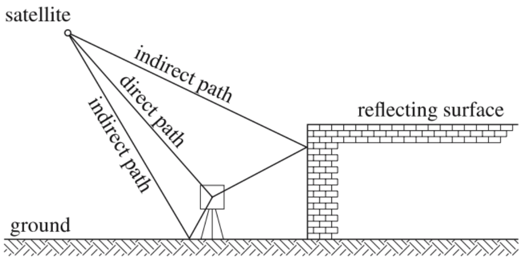

Figure 2.3: Illustration of multipath [Hofmann-Wellenhof, Lichtenegger and Wasle, 2008]

Poor geometry of satellites can decrease the accuracy of the estimated position of the receiver. When satellites used for measurements are well spread out seen from the receiver, there is good geometry of satellites. Bad geometry happens when the satellites are close together. This can, for instance, be a problem when high buildings limit the view of the sky. Position Dilution Of Precision (PDOP) is a measure of 3-dimensional satellite geometry. The higher the PDOP-value, the lower the accuracy of position estimation.

Time in GNSS is given by a GPS week number and Time of Week (TOW). TOW is given as seconds since the GPS week started.

Carrier Phase Measurement

In carrier phase measurements, change of phase in the carrier phase of the signal received from the satellite and the receiver-generated replica of the signal is measured. This gives a fraction of a cycle and defines a fractional part of the distance between satellite and receiver (given by the wavelength of the signal). This fractional part of the distance can be measured with high precision. In addition to this, an initial number of whole cycles is needed to get the entire distance. The initial number of whole cycles is unknown, and this number is called the ambiguity in this thesis. Once the ambiguity is solved, distance between satellite and receiver is known.

Observation equation for GNSS carrier phase is [Misra and Enge, 2012]:

$$\phi = \rho + c(\delta t_R - \delta t^S) + T - I + \lambda N + \epsilon_\phi \tag{2.3}$$

where ϕ is phase measurement expressed in range, ρ is geometrical distance between receiver and satellite, c is speed of light, δt_R and δt^S is clock bias of receiver and satellite, T and I are atmospherical delays of the signal caused by the troposphere and the ionosphere respectively, λ is carrier wavelength, N is ambiguity and ϵ_ϕ is other errors.

As there are errors in both the clock in the satellite and in the receiver, the clock offsets need to be estimated along with the ambiguity.

Differential GNSS

If the position of a GNSS receiver is known, the combined effect of all errors affecting the calculated distance between each satellite and this receiver can be estimated. The basic idea of Differencial GNSS (DGNSS) is to make these error estimates available for other GNSS receivers and use them to improve their position estimates. DGNSS has a potential accuracy of $5mm + 1ppm$ when used in post-processing analysis [Statens Kartverk, 2009].

Single difference is typically done by two receivers observing the same satellite at the same time, given by equation 2.4 and visualized by the orange dotted lines in figure 2.4.

$$\begin{aligned}\Delta\phi^i_{ab} &= (\rho^i_b - \rho^i_a) \\ &\quad + c((\delta t_{R}{}^i_b - \delta t_{R}{}^i_a) - (\delta t^{S^i}{}_b - \delta t^{S^i}{}_a)) \\ &\quad + ((T^i_b - T^i_a) - (I^i_b - I^i_a)) \\ &\quad + (\lambda N^i_b - \lambda N^i_a) \\ &\quad + (\epsilon_{\phi}{}^i_b - \epsilon_{\phi}{}^i_a) \\ &\approx (\rho^i_b - \rho^i_a) + c * (\delta t_{R}{}^i_b - \delta t_{R}{}^i_a) + (\lambda N^i_b - \lambda N^i_a) + \epsilon_{\phi}{}^i_{ab}\end{aligned} \tag{2.4}$$

If the baseline between receiver a and b is short, the clock and orbit errors of the satellite are the same for both receiver observations, giving $\delta t^{S^i}{}_b - \delta t^{S^i}{}_a = 0$. Short baselines also reduce tropospheric and ionospheric errors, as the signals have gone nearly the same way through the atmosphere. This gives $T^i_a \approx T^i_b$ and $I^i_a \approx I^i_b$.

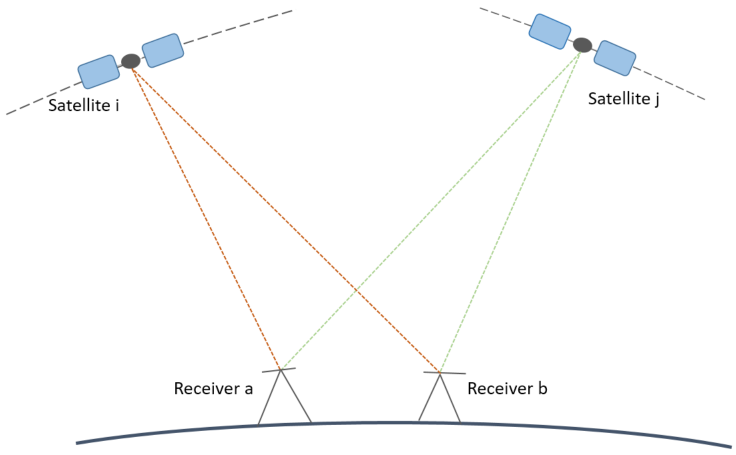

Figure 2.4: Illustration of DGNSS

Double difference is the difference between to single differences observed at the same time, visualized by the orange and the green dotted lines in figure 2.4, given by equation 2.5. Two receivers and two satellites are used. Receiver clock errors are observed twice and can be differenced out. This is often used as a base for estimating the ambiguity.

$$\begin{aligned}\nabla\Delta\phi^{ij}_{ab} &= \Delta\phi^j_{ab} - \Delta\phi^i_{ab} \\ &\approx \rho^{ij}_{ab} + \lambda N^{ij}_{ab} + \epsilon_{\phi}{}^{ij}_{ab}\end{aligned} \tag{2.5}$$

Integer Ambiguity Resolution

Ambiguity resolution is the process of resolving the unknown number of N to an integer value in carrier phase measurements. The ambiguity is then said to be resolved, or fixed (the *fix solution*). Getting the wrong fix will cause an error in estimated distance between satellite and receiver. Fixing N with an error of one cycle will give an error of typically $0.20m$.

The LAMBDA (Least-Squares AMBiguity Decorrelation Adjustment) method is one of several methods for resolving integer ambiguities, and the method used in TerraPos. The LAMBDA method is shortly described in the following paragraphs [Misra and Enge, 2012]:

The first step is a weighted least-squares estimation, using a set of double difference carrier phase measurements for different satellites. The weighted least-squares criterion takes correlation among the double differences into account. The result of this estimation is real-valued numbers of ambiguities. The real-valued number are called *float solutions*.

The next step is to define the search space for integer ambiguities. The boundary for the search space is an ellipsoid defined by the variance matrix, centering the float solution. Due to high correlation between ambiguities in GNSS, the ellipsoid is usually extremely elongated. A transformation that decorrelates the ambiguities as much as possible is done to make the search space more spherical. The search is then carried out to find the integer ambiguities.

The last step is validation of the ambiguity solution. This is important, since fixed solutions should only be used if there is enough confidence in this solution. Validation of the ambiguity is still an open problem. Further discussion of it can be found in [Verhagen, 2005].

2.2.4 Odometer

Another sensor used to aid positioning by the INS is an odometer, also called a Distance Measurement Indicator (DMI). The odometer is a sensor that performs distance measurements. This can be done by counting number of revolutions for a wheel on the vehicle. The odometer can be used to limit along-track position errors. Errors and drift in odometer measurements can for instance come from incorrect estimation of wheel size, wheel slippage and compaction of the wheel or ground.

The odometer can give *zero velocity updates*, mentioned in section 2.2.2, informing the system of when it is standing still, which is helpful to reduce drift in the IMU.

2.3 Laser Scanning

Laser scanning is a surveying technique that uses laser to measure distance to a point on a surface. Laser scanners can collect a large amount of points on every second, making it an efficient surveying method. A set of laser points is called a *point cloud*. Data in a point cloud can be used to extract information about the scanned surface. It can also be used for visualization purposes. If images are collected along with the point cloud, the points can be colored by the images, giving a colored 3-dimensional model of the scanned area.

The coordinates of the scanned points is given in a local coordinate frame defined relative to the laser scanner. To get coordinates in another frame, the relation between the two frames has to be established. In most cases a 3D translation and rotation is sufficient. In terrestrial laser scanning, the scanner stands still on the ground while scanning, thus all points in the same scan are scanned from the same position. In mobile (mapping) and airborne laser scanning, the scanner is constantly moving, either on ground or in the air, making it more difficult to position the scanner at each point of time.

As mentioned in chapter 2.2, aided INS can be used to reconstruct the trajectory of the platform in mobile mapping. An estimated pose for the laser scanner is found for each point in time by the trajectory and translation and rotation between the local laser scanner coordinate frame and b-frame. The rotation (the *boresight*) and the translation between the two frames need to be determined by calibration. Errors in calibration cause systematic errors in the point cloud

There are different methods of laser scanning used in mobile mapping. The laser scanners used in this project uses *time-of-flight*, or Light Detection and Ranging (LIDAR), measurements, where a laser pulse is sent out and and the time it takes for it to return from the scanned surface is used to calculate the distance.

Another method is *phase measurement*, where a constant constant beam of laser light that is emitted from the scanner is utilized. The beam consists of laser light of alternating frequencies. The scanner then measures the phase shift in the laser light when it returns to the scanner after its reflection on an object's surface. This is used to calculate distance. To get an unambiguous range, the range is limited to the phase delay of one complete sine wave. This typically gives a maximum operating range of 100 m. The operating range is shorter than for time-of-flight scanning, but the precision is higher.

2.3.1 Time-of-Flight Measurement

In time-of-flight measurements, the laser scanner sends out a laser pulse, the pulse hits a surface and reflects back to the laser scanner. The range (distance), ρ, to the reflecting surface can be calculated based on the time, t, it takes from the laser pulse leaves the scanner till it comes back and the speed of light in vacuum,

c, corrected by a refractive index that depends on air temperature, pressure and humidity, n, see equation 2.6, [Vosselman and Maas, 2010].

$$\rho = \frac{c}{n} \frac{t}{2} \tag{2.6}$$

The range, ρ, and orientation, θ, to the point on the surface were the laser pulse was reflected, gives a point in the local laser scanner coordinate system. Different scanner mechanisms are used to move the laser pulse over the area to be scanned. An oscillating or rotating mirror directing the pulse across the area is a commonly used mechanism.

The area the laser pulse covers when reflecting on an object, its *footprint*, is not infinitely small, see formula 2.7, [Vosselman and Maas, 2010]. This results in an angular uncertainty of the reflected point. w is the radius of the laser beam at the contour of 13.5% of peak intensity. z in the distance from the beam waist location at its minimum radius ω_0 to the reflection point and z_0 is distance from the focusing lens to the beam waist location at its minimum radius. λ is wavelength of the laser source and n is the refractive index.

$$w(z) = \omega_0(1 + (\frac{\lambda(z - z_0)}{n\pi\omega_0^2})^2)^{1/2} \tag{2.7}$$

Part of the pulse can be reflected and part of it continue on and be reflected later, giving multiple returns. This can be helpful, giving more information about the terrain, for instance when scanning a tree. When scanning from the air, the first return of the laser beam may come from the canopy of a tree, the second from a branch and a last return form the ground. Figure 2.5 shows the difference between the first and the last return in a city area.

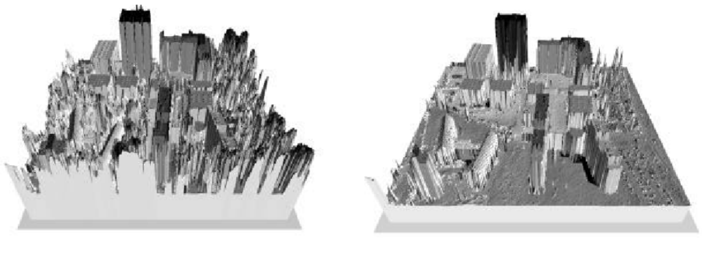

Figure 2.5: Image visualizing first return (left) and last return (right) in a city area scanned from air [Horemuz, 2015]

Larger footprint of the laser pulse reduces the accuracy of the reflected point. Increased distance between the scanner and the object to be scanned, gives a larger footprint. If the laser pulse reflects on a sloping surfaces the footprint will also be larger.

A measure of level of detail in point clouds in density. With higher point density, it is easier to define features in the point cloud. Number of laser pulses sent out per second, speed of the spread mechanism of pulses and distance from scanner to the surface to be scanned are parameters affecting density. In mobile mapping, velocity of the platform also affects the density.

As the points in a point cloud are given in 3-dimensional space, objects can be identified by their geometry or shape. This makes it possible to measure the size and volume of objects in the point cloud. The intensity of points can also be used to identify features in the point cloud. Intensity of a point is a measure of the strength of the return of the laser pulse that generated the point. It is a relative measure, partly dependent on the reflectiveness of the material of the reflecting surface and the incidence angle of the laser pulse. Intensity of the points is useful for visualization, see figure 2.6, and can be used for identifying features in the point cloud.

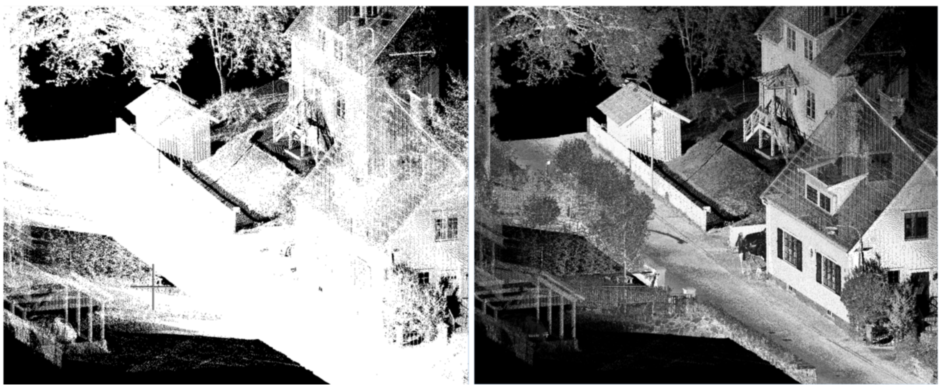
Figure 2.6: Point cloud from mobile mapping with all points in white color (left image) and points colored by their intensity (right image)

Multipath can cause problems in laser scanning. This happens when the laser pulse hits more than one surface before returning to the scanner. Because the pulse travels a longer distance than straight back to the scanner, the resulting laser point will appear further away from the scanner than the surface it represents. Other errors can be systematic and caused by imperfections in instrument manufacture and assembly.

2.3.2 Scan Matching

To get better coverage and accuracy of an area, parts of it are often scanned more than one time, giving overlap between the different scans. When scanning is performed while the scanner is standing still, at least three well-distributed points, that can be identified in both scans, are needed to match the two scans. This process can be called *scan matching.* The task is to find the link between the different scans and then transforming all points by that link.

Strip Adjustment

In airborne and mobile laser scanning, the scanner is constantly moving, and this movement is estimated by the trajectory. The laser points are linked to the trajectory by a time stamp, the vector from the laser scanner to the point and the vector from the trajectory to the laser scanner. The accuracy of the point cloud cannot be better than the accuracy of the trajectory. In areas with sufficient GNSS signals, the point cloud generated by the trajectory, can be quite good. Offsets and orientation errors between *strips*, the overlapping parts of the points cloud, can however often be found. *Strip adjustment* can be done to reduce errors and increase the accuracy of the point cloud. Offsets between strips can be caused by a number of reasons, e.g. trajectory drift (errors in position and orientation), boresight error and errors caused by laser scanner.

There are multiple algorithms made for matching overlapping strips from mobile mapping and airborne laser scanning. The general procedure is usually:

1. Find connection between the scans/strips to match; trajectory position offsets (dx, dy, dz) and orientation errors d_{roll}, d_{pitch}, $d_{heading}$)
2. Adjust point cloud according to offsets and errors found

2.4 Kalman Filter

Kalman filtering is an important computational tool in mobile mapping. The Kalman filter is used for state estimation. It combines the information in a continous noisy model of the dynamics of the system, see equation 2.8, with a series of noisy measurements and a measurement model, see equation 2.9 [Brown and Hwang, 2012]. By this, estimates of the state tend to be more accurate than estimates by a single measurement alone.

$$\dot{x} = Fx + Gu \tag{2.8}$$

$$z_t = H_t x_t + v_t \tag{2.9}$$

x is the state vector, u is a vector forcing function whose components are white noise, z is the noisy measurement vector, v is the measurement noise vector and F, G, H are known matrices.

The Kalman filter is an implementation of the Bayes filter under the assumtion of linear and gaussian models. If this is the case, the Kalman filter is the optimal state estimation algorithm [Farrel, 2008]. The Kalman filter has a number of applications, for instance in navigation, robot motion planning and trajectory optimization. It is useful for sensor fusion.

The discrete Kalman filter is divided into two steps; the time update state and the measurement update step, executed at each time step t. At each time step, the gaussian estimation of the state is represented by mean, x_t and covariance, P_t. Equations 2.10 to 2.14 describe the discrete Kalman filter algorithm and is based on [Brown and Hwang, 2012].

In the time update step, also called the prediction step, the state of the current time step is estimated, $\tilde{x}_t$, see equation 2.10. This estimation is done based on the best estimate of the previous state $\hat{x}_{t-1}$ and the state transition model, Φ. The state transition model describes how the system state evolves by itself.

Equation 2.11 gives an estimate of the covariance matrix of the state before any measurements are done, $\tilde{P}_t$. It describes the uncertainty of the estimate of the state vector. It is based on the covariance matrix of the previous best estimate of the system state, $\hat{P}_{t-1}$, the state transition model, Φ, and the process noise Q_{t-1}. Φ and Q are computed based on F and G.

$$\tilde{x}_t = \Phi_{t-1}\hat{x}_{t-1} \tag{2.10}$$

$$\tilde{P}_t = \Phi_{t-1}\hat{P}_{t-1}\Phi_{t-1}^T + Q_{t-1} \tag{2.11}$$

In the measurement update step, measurements done by the sensors at time t, z_t, are used to improve the estimation of the state vector found by the prediction step, giving the best estimate of the state vector at time t, $\hat{x}_t$. Equation 2.12 to

2.14 describes the measurement update. H_t is the design matrix, giving the ideal connection between the measurement and the state vector at time t, and R_t is the measurement noise.

$$K_t = \tilde{P}_t H_t^T (H_t \tilde{P}_t H_t^T + R_t)^{-1} \tag{2.12}$$

$$\hat{x}_t = \tilde{x}_t + K_t(z_t - H_t \tilde{x}_t) \tag{2.13}$$

$$\hat{P}_t = (I - K_t H_t)\tilde{P}_t \tag{2.14}$$

The Kalman gain, K_t, describes the relative uncertainty between the predicted estimated state found in the prediction step and the sensor observations. It is used to find a weighted sum of them. The lower the uncertainty in a sensor is, the more the system will trust the sensor observation and the estimate will be pushed toward this instead of the predicted estimate. With high gain, more weight is put on the measurements. With low gain, more weight is put on the system model prediction. Low gain smooths out more noise, but decreases responsiveness of the the system.

The latest prediction is based only on the last predicted state. All necessary information from previous observations is stored in the last prediction. This reduces the memory needed to run the filter.

When running the Kalman filter in post-processing, it is possible to run it both forwards and backward. By doing this, the state at a given time will be estimated by all other observations, both observations before and after the observation at the given time.

As discussed in section 2.2.2, the INS consists of a high rate sensors with high short-time stability, but drifts over time. The GNSS is a low rate sensor and has biases in a short period of time, but high long-time stability. Hence, the INS and GNSS are complementary sensors. The INS is used to predict the system state and GNSS measurements are used to correct the measurements.

Extended Kalman Filter

If the models are not linear, the Extended Kalman Filter (EKF) can be used. In this implementation, the models describing the dynamics and measurements of the system are linearised at each point in time to estimate the linear case, and the Kalman filter is then used.

Complementary Kalman Filter

In a *Complementary Kalman Filter*, the dynamics model does not estimate the system state, but system errors. What enters the Kalman filter is the difference between the INS observations and observations from aiding sensors, e.g. the GNSS. The actual state dynamics do not enter the system. The complementary Kalman filter is an effective means to obtain a linear system and to avoid complex filter tuning [Kjørsvik, 2010]. See the flowchart describing the process in figure 2.7.

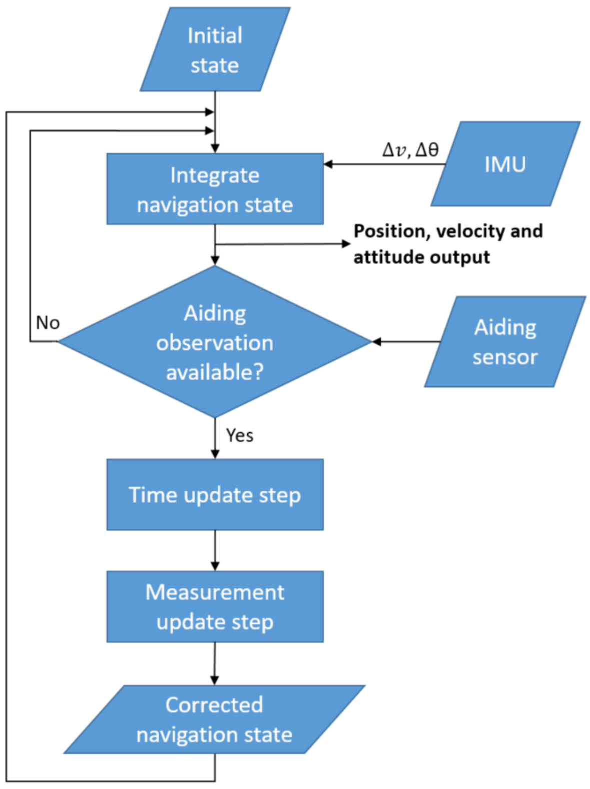
Initial state
Integrate navigation state
Δv, Δθ
IMU
Position, velocity and attitude output
No
Aiding observation available?
Aiding sensor
Yes
Time update step
Measurement update step
Corrected navigation state

Figure 2.7: Flowchart of the Complementary Kalman filter process. Adapted from figure 1 in [Kjørsvik, 2010]

Delayed-State Filter

The delayed-state filter takes advantage of the relationship between current measurements and the past state of the system, see the measurement equation in 2.15. Delayed measurements can come from a vision system, e.g. incremental observations from laser scanning. A detailed explanation of the delayed-state filter can be found in [Brown and Hwang, 2012].

$$z_t = H_t x_t + J_t x_{t-1} + v_t \tag{2.15}$$

Because of the x_{t-1} term in the measurement equation, it does not fit the usual Kalman filter. The measurement update step is thus changed to the following equations:

$$K_t = (\tilde{P}_t H_t^T + \Phi_{t-1} P_{t-1} J_k^T) L_k^{-1} \tag{2.16}$$

$$\tilde{z}_t = H_t \tilde{x}_t + J_t x_{t-1}) \tag{2.17}$$

$$\hat{x}_t = \tilde{x}_t + K_t (z_t - \tilde{z}_t) \tag{2.18}$$

$$\hat{P}_t = \tilde{P}_t - K_t L_t K_t^T \tag{2.19}$$

where L is the residual covariance given by:

$$L_t = H_t \tilde{P}_t H_t^T + R_t + J_t P_{t-1} \Phi_{t-1}^T H_t^T + H_t \Phi_{t-1} P_{t-1} J_t^T + J_t P_{t-1} J_t^T \tag{2.20}$$

Chapter 3

Simultaneous Localization And Mapping

Simultaneous Localization And Mapping (SLAM) is a computational problem widely used in estimation in the robotics community. This chapter will present the principles of SLAM and discuss the theoretical background for the methods presented and tested in this thesis. *The robot* discussed in this chapter represents the platform in mobile mapping. *Landmarks* in this chapter correspond to the concept of tie points in this thesis.

3.1 Basic Concepts

The SLAM problem is the problem of estimating a map of an unknown environment while keeping track of the robots pose relative to the map [Thrun and Leonard, 2008]. The map can be used to limit the errors in state estimation and aid the INS measurements.

The SLAM problem can be explained as follows [Thrun and Leonard, 2008]. A robot starts at a known position, x_0. Relative measurements given by equation 3.1 are made to measure how the robot moves. The movement of the robot is uncertain. This makes the position of the robot at time t, x_t, gradually more difficult to determine, see figure 3.1. The trajectory of the robot is given by equation 3.2. u_t describes the motion between the time t and $t-1$.

$$u_t = u_1, u_2, u_3, ..., u_t \tag{3.1}$$

$$x_{1:t} = x_0, x_1, x_2, ..., x_t \tag{3.2}$$

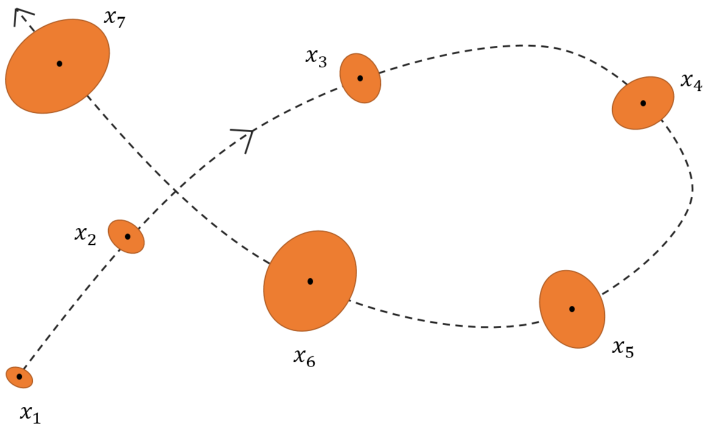

Figure 3.1: Figure showing growth of uncertainty as the robot moves. Estimated pose is a black dot with the uncertainty of the estimated pose as an orange ellipse. The larger the uncertainty ellipse is, the less certain the estimation of the robot's pose is.

As the robot moves, it senses its environment, m. Under the assumption that only one measurement, z, is done at each point in time, equation 3.3 gives the sequence of measurements. Examples of sensors are cameras and laser scanners.

$$z_t = z_1, z_2, z_3, ..., z_t \tag{3.3}$$

Measurements of the environment, $z_{1:t}$, and movement of the robot, $u_{1:t}$, are used to perform *mapping*, recovering of the map m, and *localization*, recovering of the position of the robots $x_{1:t}$. This gives the equation of the full SLAM problem, see equation 3.4. Two models are needed to solve the SLAM problem; a mathematical model that relates the sensor measurements, u_t, to position of the robot, x_t, and a model that relates measurements, z_t, to the environment, m, and the robot location, x_t.

$$p(x_{1:t}, m|z_{1:t}, u_{1:t}) \tag{3.4}$$

The alternative to the full SLAM problem is online SLAM, see equation 3.5. In online SLAM, only the current position of the robot is of interest, not the entire trajectory. Online SLAM can for instance be used by autonomous robots. They need to know where they are and how they can move next.

$$p(x_t, m|z_{1:t}, u_{1:t}) \tag{3.5}$$

3.1.1 Landmark-based SLAM

The assumption in landmark-based SLAM is that the environment contains a number of point landmarks that can be sensed by the robot. By the use of observations from a laser scanner, the landmark's positions can be found by range and direction from the robot.

Loop closure events are used to get global consistency in a map. A robot relying on relative measurements interprets the world as an ”infinite corridor”, see the left image in figure 3.2. A second observation of a landmark in the environment gives a loop closure event. By this, information of the corridor intersecting itself is obtained. Thus, the real topology of the environment is found. This can be seen in the right image in figure 3.2 where the second observation of landmark A and B is used for loop closure events. [Cadena *et al.*, 2016]

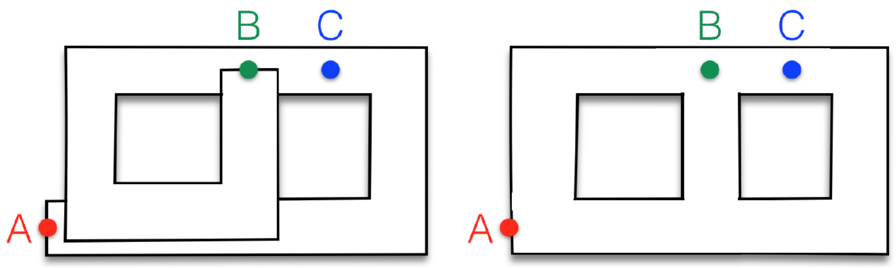

Figure 3.2: The left map is built from relative measurements from point A to point B. The right map is build using SLAM with loop closure events, estimating the real topology in the map [Cadena *et al.*, 2016]

Loop closure events can also reduce the uncertainty of the robot's state. If a landmark is observed at the beginning of a robot's trip, uncertainty of both the robot and the landmark will be quite low. After some time, uncertainty of the robot's pose has grown, and the same landmark is sensed again, see figure 3.3. The robot's pose can then be corrected by the observed range and direction to the landmark, which reduces uncertainty of the robot's position. This does not only improve uncertainty of the estimated robot's pose, but of all landmark estimations. Information that helps localize the robot propagates through the map [Thrun, Burgard and Fox, 2005].

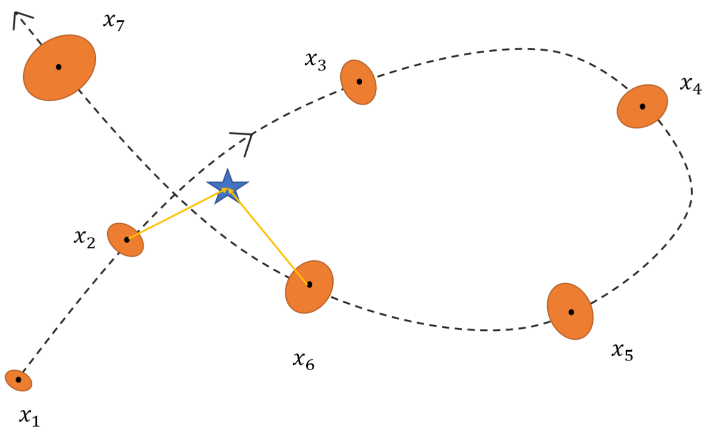

Figure 3.3: Figure showing how uncertainty of the robot pose is affected by loop closure. Compared to figure 3.1, where no landmark observations are included, it can be seen that the error growth is reduced by observing the landmark (blue star) twice. Estimated pose is a black dot with the uncertainty of the estimated pose as an orange ellipse. The larger the uncertainty ellipse is, the less certain the estimation of the robot's pose is.

One of the main sources of algorithm failures in SLAM is data association [Cadena *et al.*, 2016]. Data association matches measurements done by the robot to landmarks in the environment. Erroneous data association degrades the quality of state estimation and makes it harder to detect outliers later on. Landmark detection and validation is known as the *data association problem* in SLAM

3.1.2 SLAM paradigms

The Extended Kalman Filter (EKF) SLAM is the earliest formulation of SLAM and has been applied to a large number of navigation problems. The algorithm can only handle a limited number of landmarks before getting very computationally demanding, but it is suitable for integration with systems with multiple sensors.

Graph-based SLAM uses a graphical representation of the SLAM problem. Landmarks and robot locations are thought of as nodes in a graph, and an arc links consecutive pairs of nodes. Arcs also exist between locations and landmarks. The main limiting factor in EKF SLAM is the covariance matrix, which grows with a larger state vector and takes space and update time. The update time in graphical models is constant. Performing optimization of the graph can, however, be expensive. [Thrun and Leonard, 2008]

The EKF SLAM is what is implemented in TerraPos and used for the work in this thesis. Thus, this method will be discussed further in the next section.

3.2 Extended Kalman Filter SLAM

In addition to estimating the robot's pose, x_t, the EKF SLAM algorithm estimates position of all landmarks. By this, the coordinates of all landmarks are added to the state vector in the EKF. Hence, the state vector grows, see equation 3.6 for estimation in 3-dimensional space. $x, y, z, \phi, \theta, \psi$ is pose of the robot, $m_{1,x}, m_{1,y}, m_{1,z}$ is position of the first landmark, $m_{n,x}, m_{n,y}, m_{n1,z}$ is position of the n^{th} landmark and n is number of landmarks.

$$x_t = (x, y, z, \phi, \theta, \psi, m_{1,x}, m_{1,y}, m_{1,z}, ..., m_{n,x}, m_{n,y}, m_{n1,z}) \tag{3.6}$$

The more landmarks that are included in the estimation, the more complex the estimation will be. This means that finding as few landmarks as possible while maintaining an improved uncertainty in the estimation is ideal. It is also important not to add landmarks that will not be observed multiple times to the state vector. If they are not seen again, they will not contribute to the estimation.

A key limitation of the EKF SLAM is the necessity of careful landmark selection. This often makes it necessary to either select only perceptual distinctive landmarks or to keep landmarks far away from each other, reducing the chance of confusing them with each other. [Thrun, Burgard and Fox, 2005]

Both observations of landmarks, back-end SLAM, and incremental tie point observations, front-end SLAM, can be utilized in EKF SLAM. The observation types are complementary and are well suited to aid the INS.

3.2.1 Back-end SLAM

Back-end SLAM takes advantage of the loop closure effect in landmark-based SLAM, see section 3.1.1. The objective is to aid global localization of the trajectory.

Landmark observations give position and orientation updates for the robot pose directly. Thus, integration of measurements is not needed and this limits drift in the observations.

3.2.2 Front-end SLAM

Front-end SLAM takes advantage of incremental observations from for instance a visual sensor, to aid the pose estimation in the Kalman filter. The front-end module extracts relevant features from sensor data and measures relative change. When this is done with a visual sensor, e.g. laser scanner or camera, it is called visual odometry. It corresponds to the estimated "corridor" in the left image in figure 3.2. The measurements from the visual odometer can be used to aid local localization of the trajectory, but even small errors will cause drift.

The visual odometer gives updates as position increments. Position increments need to be integrated (summed) once to give position, which is what we are interested in. This leads to some drift in the observations.

Observations from the visual odometer are not as prone to drift as accelerometers, where double integration is needed, but the precision of a visual odometer observations is lower. The frequency of visual odometer is typically somewhere between the frequency of INS measurements and GNSS measurements. Because of this, the effect of the visual odometer measurements is complementary to the effect of INS measurements and GNSS measurements.

Chapter 4

Data Collection and Post-Processing

The platform (vehicle) used for data collection and software used to post-process collected data is presented in this chapter. Two methods using SLAM is post-processing of mobile mapping data will be presented.

4.1 Optech Lynx Mobile Mapping System

The Optech Lynx MMS operated by TerraTec consist of several instruments and sensors used to produce accurate data. The INS and its sensors is mounted inside the vehicle. The odometer is mounted in the left rear wheel. Two laser scanners using LIDAR technology and the GNSS antenna can be seen mounted on the roof of the vehicle in figure 4.1. The scanners spin diagonally of the car's driving direction. Having two scanners increases coverage. Objects can be scanned both before the car passes the object from one scanner and from the other scanner when it has passed. Both laser scanning and imaging can be done by this mobile mapping system. For the 360 degree image, a ladybug camera system is mounted on top of the car. It consists of 6 cameras that take images in different directions. Only laser data is used for mapping in this study.

The Optech Lynx MMS is a multi-purpose system and can be applied to a number of different projects. Examples are mapping of road, rail road and coastal areas.

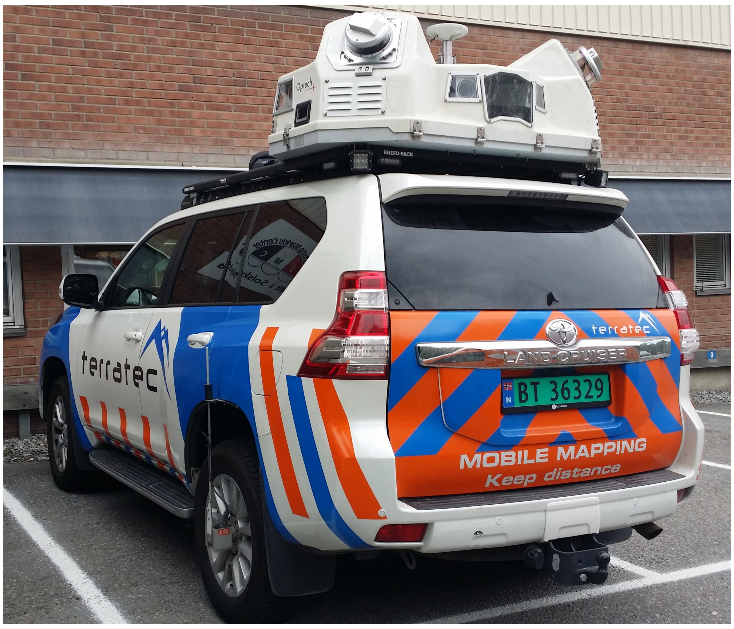

Figure 4.1: The Optech Lynx MMS seen from behind

4.1.1 Technical Specifications

System specification for Optech Lynx MMS is found in table 4.1 below.

Table 4.1: Technical Specification of Optech Lynx MMS

Frequency (points/sec/scanner)	600.000 Hz
Laser points per second	1.200.000
Measuring method	Time-of-flight
Number of returns	4
Range (with 10% reflection)	250 m
Scanner frequency	250 Hz per scanner
Field of view	360° x 2
GNSS/IMU	Applanix AP60 (POS/LV 610)
Update frequency raw IMU data	200 Hz
Update frequency raw GNSS data	5 Hz
360 camera, Ladybug 5	30 Megapixels

4.1.2 Error Budget of Measurements

As mentioned in section 2.2.3, DGNSS has a potential accuracy of $0.005m + 1ppm$. In this project, only baselines shorter than $10km$ are used, which gives $0.015m$ as a potential accuracy for the longest baselines.

The system specification of Applanix AP60, which is the GNSS-inertial system for continuous mobile positioning used in the Optech Lynx MMS, gives an RMS of a post-processed trajectory of $0.02 - 0.05m$ in position, $0.005deg$ is roll and pitch and $0.015deg$ in heading [Trimble, 2017]. This is given both in the case without GNSS outages and with a $60s$ GNSS outage.

Problems arise when GNSS signals are vacant for a longer period of time. As mentioned in 2.2, this due to GNSS observations being used as external aid to support inertial navigation and reduce drift. Examples of areas where this can be an issue are tunnels, city areas with high buildings and forest roads.

The laser observations of the Optech Lynx MMS has a precision of the range of $0.005m$. Absolute accuracy of observations done by the system is $0.05m$ [Teledyne Optech, 2017]. This accuracy is given under the assumption of a post-processed trajectory from good GNSS data (PDOP< 4), aid from odometer and $10m$ range. In the case of poor or lost GNSS, the performance is expected to degrade.

4.2 Software

The flowchart in figure 4.2 describes the process of post-processing mobile mapping data without SLAM. The steps of trajectory processing, point cloud generation and point cloud observations is explained in the following sections.

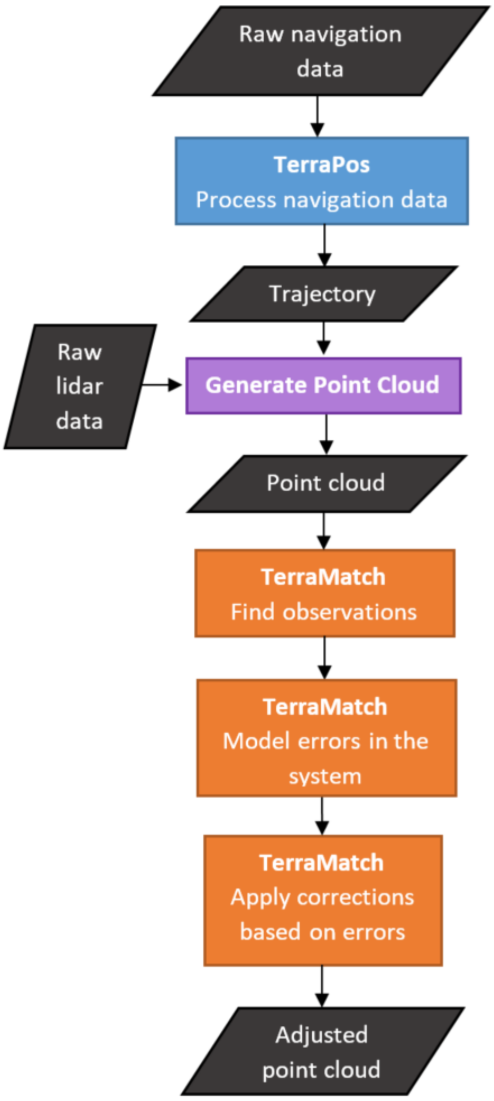

Figure 4.2: Flowchart describing method of post-processing without SLAM.

4.2.1 TerraPos - Trajectory Processing

TerraPos is a TerraTec software made for GNSS/INS post-processing and was used in this thesis. Navigation data from the mobile mapping system can be used to calculate a trajectory of the platform's position and orientation. In mobile mapping, raw data from the INS, GNSS and odometer is typically used. In this project, data from GNSS reference stations with short baselines to the trajectory were used to perform DGNSS. Precise ephemerides were used to get the most accurate data about satellites, and this was downloaded in TerraPos.

An Extended Complementary Kalman filter was used to combine raw data from different sensors to one trajectory. Output from TerraPos is a smoothed trajectory and additional information about it. This additional information includes quality of the trajectory, accuracy of positions, number of satellites, baselines and so on.

TerraPos supports Simultaneous Localization And Mapping (SLAM) in the estimation. Tie point observations can be used to get a loop closure effect by back-end SLAM. Incremental tie point observations can be used to aid positioning by front-end SLAM.

4.2.2 Optech LMS - Point Cloud Generating

The point cloud is generated based on the trajectory computed in TerraPos and raw laser data from the mobile mapping system. Optech Lidar Mapping Suite (LMS) is used to do this in this study. The output from LMS is a point cloud in .las-files in a user specified g-frame. The generation of .las-files is a time-consuming process.

4.2.3 TerraMatch - Point Cloud Observations

Terrasolid software packages TerraScan and TerraMatch can be used for point cloud processing and matching. Overlapping parts of a point cloud can be used to find misalignment angles and location errors.

Strip adjustment is explained in section 2.3. The process of strip adjustment in TerraMatch can be divided into three steps:

1. Find point cloud observation by the difference between strips. This can be done by tie line observations. If points with a known position are available, the difference between the strips and the known points can also be found.
2. Model the errors in the system ($dE, dN, dD,$ $d_{roll}, d_{pitch}, d_{heading}$) by the observations found.
3. Adjust the point cloud by the error models found.

Tie line observations is a collective name for different types of observations such as surface, lines and points that can be observed in the point cloud in TerraMatch. The information of tie line observations can be saved in a binary tie line file (.til-file).

A *surface line observation* is a representation of a surface, for instance of flat ground or a vertical wall. Surface line observations are 1-dimensional relative observations and can be used to find the shift and rotation between two (or more) representations of the same surface from different points in time.

Surface line observations can be searched for automatically in TerraMatch. These observations can be found efficiently with a limited amount of work required from the user. Classification of the point clouds can be done by e.g. range, distance from trajectory, intensity, return number and geometry of points. It is possible to perform search for surface lines in only a specific class, limiting the search area to, for instance, the road surface and reducing the computational time of the search.

Figure 4.3 and 4.4 shows examples of surface line observations. Several options are available for adjusting the surface line search, e.g. distance between each surface line observation (both in distance and time), the size of the surface the surface line represents, distance from scanner and search area.

Another type of tie line observation that can be found in TerraMatch is *XYZ observations*. XYZ observations are observations of point features that can be identified in the point cloud and need to be found manually by the user. Examples of point features that can be used are centers of manholes and corners of road markings. These observations are 3-dimensional. XYZ observations can also be saved in a .til-file.

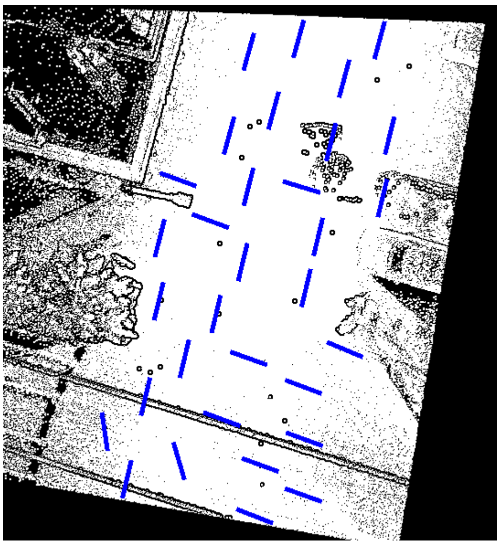
Figure 4.3: Surface lines (blue) on flat ground, shown from above. (TerraMatch)

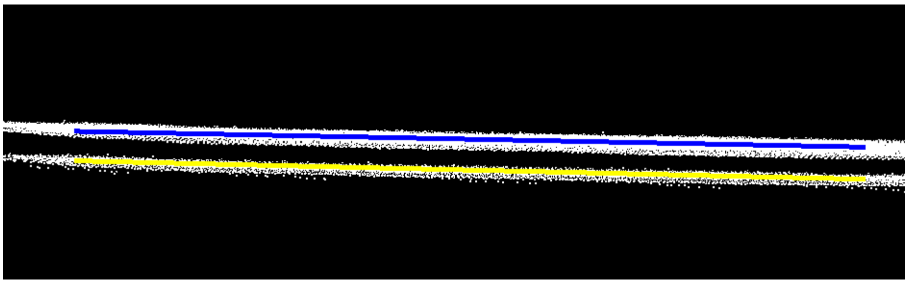

Figure 4.4: Observations of a surface line on flat ground, shown from the side. An offset between the first observations (blue) and the second observation (yellow) can be seen. (TerraMatch)

4.3 Introducing SLAM in Post-Processing

Two methods for re-calculating the trajectory based on point cloud observations have been developed. One method using back-end SLAM and one method using front-end SLAM. The general procedure of post-processing is similar for both of them and is described in the flowchart in figure 4.5. This general procedure is based on the procedure without SLAM presented in figure 4.2.

The first step was to process the trajectory in TerraPos. Based on the trajectory, the point cloud was generated in LMS. The point cloud and trajectory were then imported into TerraMatch.

The next step was to make point cloud observations in TerraMatch. The 1-dimensional surface line observations can be searched for automatically in TerraMatch, making it a suitable observations type. Tie line observations were used for both methods. These observations are given in a .til-file.

A software package was developed for this thesis for transforming information in the .til-file from TerraMatch, to an observation type and a format that TerraPos can read and utilize.

All data and choices made for the original calculation of the trajectory were kept for the re-calculation of the trajectory. This was done by cloning the subproject for the original trajectory in TerraPos. The file containing information extracted from the point cloud observations was added to the subproject. The trajectory was then re-calculated.

The re-calculated trajectory could have been used in LMS to generate a new point cloud, but this would make the process too time-consuming for it to be a valid option to the currently used processing procedure. This was solved by finding the difference between the original and re-calculated trajectory in TerraPos. The difference, defined by dE, dN, dD, $d_{roll}, d_{pitch}, d_{heading}$ for each second of the trajectory, was exported to a .tms-file. The .tms-file was then applied to the point cloud in

TerraMatch by the tool *Apply corrections*. It adjusts the point cloud to fit the re-calculated trajectory.

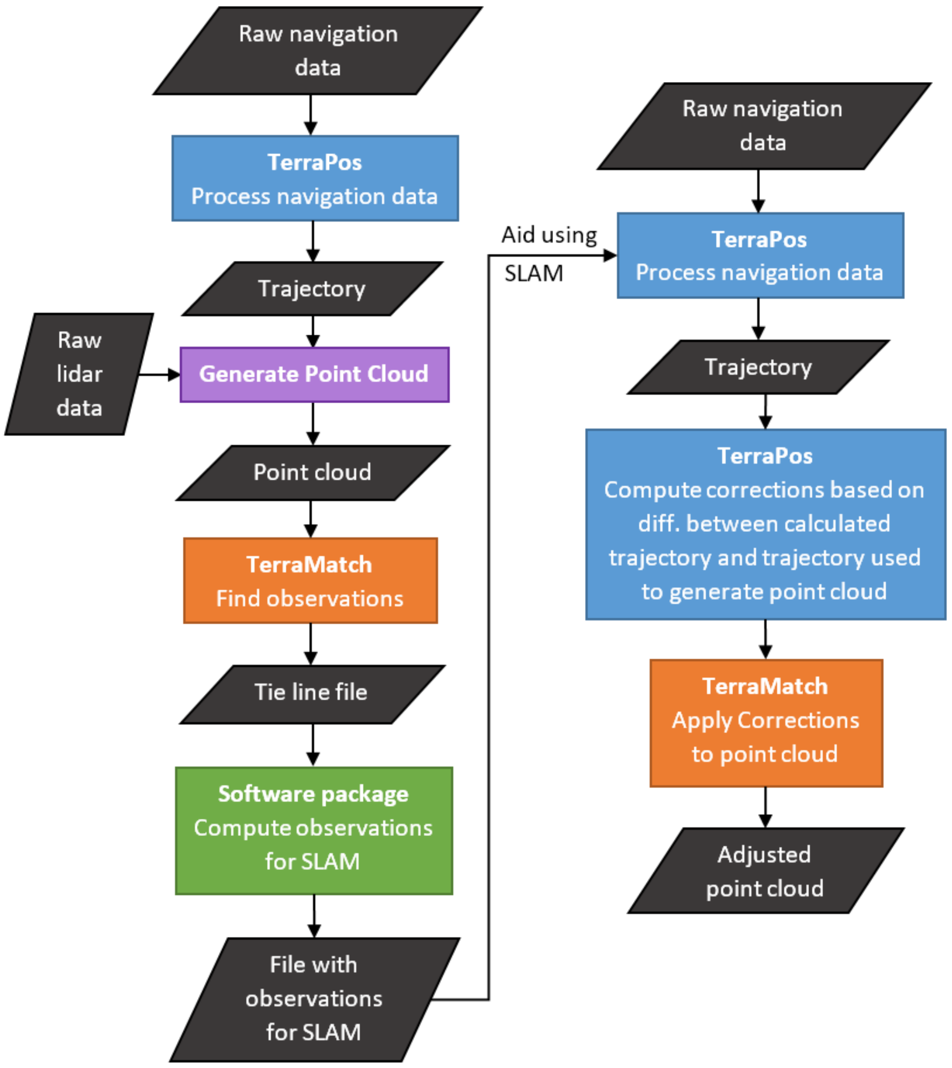

Figure 4.5: Flowchart describing method of post-processing using SLAM.

4.3.1 Back-End SLAM

The software package developed by the author of this thesis reads the .til-file and creates a .POS-file with tie point observations. The .POS-file can be added to TerraPos and used when re-calculating the trajectory. This was done to take advantage of the loop closure effect in back-end SLAM. Back-end SLAM is described in section 3.2.1.

The software package can make tie point observations from both surface line observations and XYZ observations from TerraMatch. The process of transforming the two point cloud observations types to tie point observations is presented in the following sections.

The EKF SLAM algorithm in TerraPos does not have to take the data association problem into account, as data association is done in TerraMatch. Surface line observations are associated with a tie line, and XYZ observations are associated with a 3-dimensional point feature.

Normal Vector Observations

Surface line observations in TerraMatch are 1-dimensional observations of offset between strips. A normal vector from the first surface line observation to the next observation of the same surface represents the direction of this offset, regardless of the angle of the scanned surface. The direction of the normal vector gives the direction of adjustment for strips and trajectory. The procedure for extracting tie point observations from the surface line observations is described in the following paragraphs.

Surface line observations are defined by start and end point of the line in the .til-file. This information was used to find the center of the first observation of each surface line. The normal vector was found from the other observations of the same surface line to the center of the first observation, see figure 4.6.

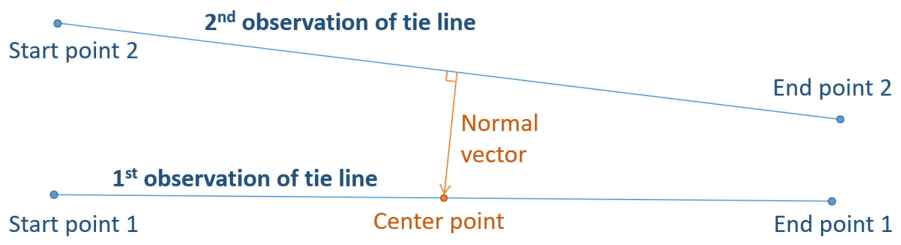

Figure 4.6: Normal vector observation found by two observations of the same surface line. Information written/drawn in orange is found by the software package. Information in blue is given from TerraMatch.

In addition to the normal vector, it was necessary to find the vector from the trajectory to the observation, see figure 4.7. Without this information, it would be hard to know whether the offset comes from drift in position or orientation of the platform.

To find the vector from the trajectory to the observation, the position of the trajectory at the time of observation was found. The time of the tie line observations was given in the .til-file from TerraMatch. TerraPos can output trajectory information at given points in time. Hence, a .txt-file with TOW of all surface line observations was created. Event tool in TerraPos was used to read the .txt-file with observations times (events) and output trajectory information for the events. The vector from the trajectory to the observations was then calculated.

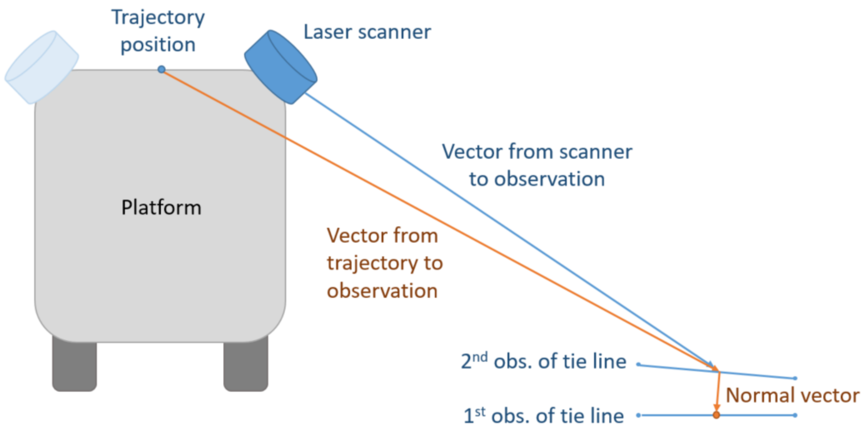

Figure 4.7: Tie point observation from surface line observation. Information of surface line observation given by TerraMatch in blue and the information calculated by the software package for the normal vector observations in orange.

The .POS-file with 1-dimensional tie point observations contained type of observation, time of observation, vector from trajectory to observation and unit vector of normal vector giving the direction of correction, as well as reference system coordinates and height were given in.

XYZ Observations

XYZ observations are 3-dimensional point cloud observations of overlapping strips. As these observations need to be found manually in TerraMatch, it is a more time-consuming process than the automatic tie line search. The benefit of relative XYZ observations is that by only one observation, corrections in all three directions is obtained. It is possible to develop automatic search algorithm for finding XYZ observations, but this was not the scope of this study.

The tie point observation, in this case, is the vector from the trajectory to the observation, visualized in orange in figure 4.8. The equivalent vector for the second observation is also needed. The information given from TerraMatch and TerraPos is shown in blue in figure 4.8 and includes trajectory position, the vector from the scanner to the observation and position of the observations.

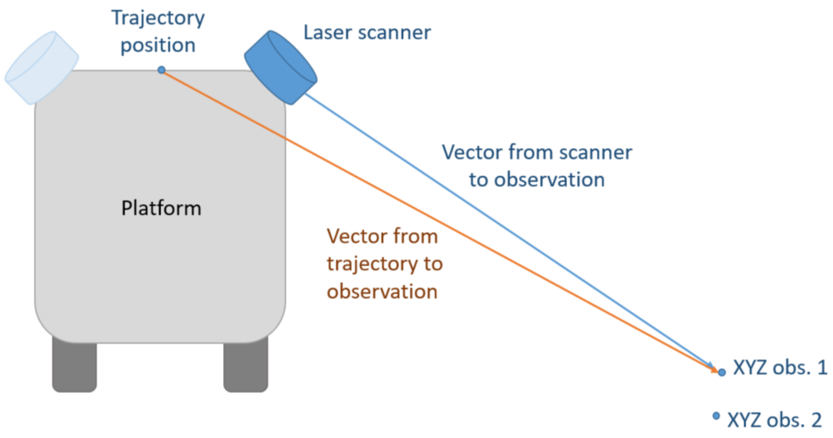
Trajectory position
Laser scanner
Vector from scanner to observation
Platform
Vector from trajectory to observation
XYZ obs. 1
XYZ obs. 2

Figure 4.8: Tie point observation from XYZ observation. Information of XYZ observation given by TerraMatch in blue and the information calculated by the software package for in orange.

4.3.2 Front-End SLAM

The second method developed for using SLAM in re-calculation of trajectory is based on front-end SLAM. This method is based on incremental height observations, providing the height of the trajectory based on incremental point cloud observations in TerraMatch. Incremental height observations are used as a visual odometer. Front-end SLAM and visual odometer is described in section 3.2.2.

The same method can be applied to incremental observations in north and east direction, provided it is possible to get continuous observations from TerraMatch. Such observations can be found in areas with many large buildings close to the road or in tunnels but were not available for this study.

Incremental Height Observations

Incremental height observations used for this thesis were generated from difference in height observed by scanner 1 and scanner 2, dz. The scanners were mounted so that they scanned in different directions at given times, see figure 4.9. Because of the angle between them, the position that was observed by scanner 1 at time t_1 is scanned by scanner 2 at time $t_2 = t_1 + dt$. If the observation of height is not the same by scanner 1 at t_1 and scanner 2 at time t_2, there must be some error in the system, e.g. systematic error or drift in the estimated trajectory.

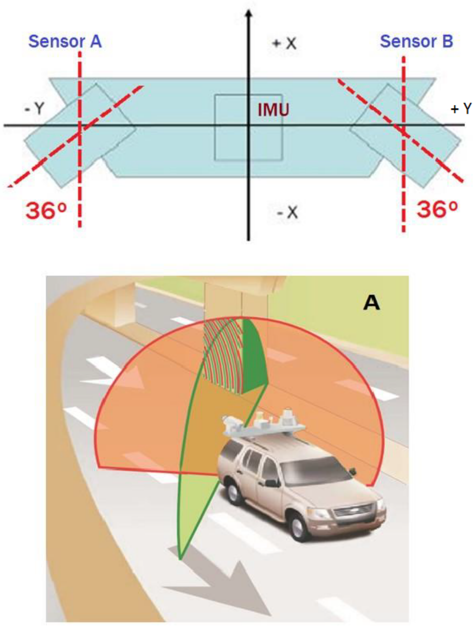

Figure 4.9: The angle between scanner 1 (sensor A) and scanner 2 (sensor B) cause a time gap between the observations of the same position by different scanners [TerraTec AS, 2016]

Pairs of surface line observations by scanner 1 and scanner 2 with a short time between were found by an automatic search for surface line observations in Terra-Match. This gave a large number of incremental surface line observations. For the testing done, surface line observations were found on flat horizontal surfaces, which in this case was the road surface. By using surface line observations of the road, continuous observations were possible. An example of this can be seen in figure 4.10.

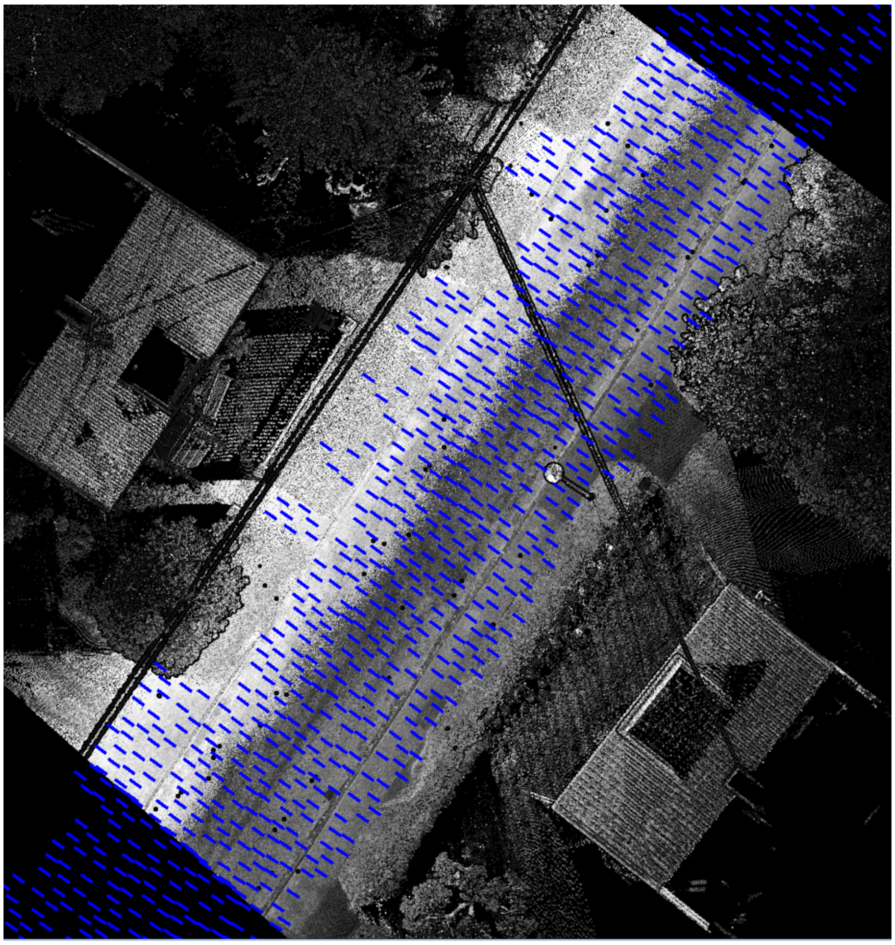

Figure 4.10: A part of the test area used with surface line observations in blue (TerraMatch)

The .til file with surface line observations from TerraMatch was used to make an .obs-file listing all incremental surface line observations found. For each observation, time t_1 of scanning the point from scanner 1 and time t_2 of scanning the same point from scanner 2 was given. The incremental observation $dz = z_{t1} - z_{t2}$ was also given. Additionally, height of the trajectory at t_1 and t_2 was provided.

In a world with no other errors than drift, the sum of observations in the time increments would equal the total drift. This is usually not the case in practice. There is a number of systematic errors in the system. A systematic error of a few millimeters for all increments will lead to a large error in observed drift in the system. As an example, a systematic error of $0.001m$ in all observations and observations every $0.25sec$ will cause a drift of $0.24m$ in only a minute. Thus, systematic errors needed to be identified and removed.

Time between observations from scanner 1 and 2, $dt = |t_1 - t_2|$, was not constant. It is dependent both on speed, orientation of the vehicle and distance from the trajectory to the observations in y-direction in b-frame (left/right of the car).

Least-square estimation was done to get a constant dz for each time increment. The estimation was done based on the incremental surface line observations in the .POS-file corrected for systematic errors. Time increments of $0.25s$ were used for the estimation. Smaller time increments would have given a finer resolution, but also made it harder to find connections of height observations between all time increments.

Estimated trajectory height was used to make a .txt-file with incremental height observations and standard deviation of these heights for each time increment. As time increments are $0.25s$ long, there are 4 estimates per second. The .txt-file with incremental height observations was added to TerraPos as a visual odometer to aid positioning, by the use of the delayed-state filter described in section 2.4. Interpolation of height estimates had to be done to fit the time of the discrete update step in the Kalman filter in TerraPos. The trajectory was then re-calculated.

Chapter 5

Numerical Investigations

The new methodology using SLAM that is developed in this study was tested in an area in Stockholm. Data from the suburb Bromma was used as a test area, see figure 5.1. Scanning was done on May 17th, 2016 with the Optech Lynx MMS. Results are presented in this chapter.

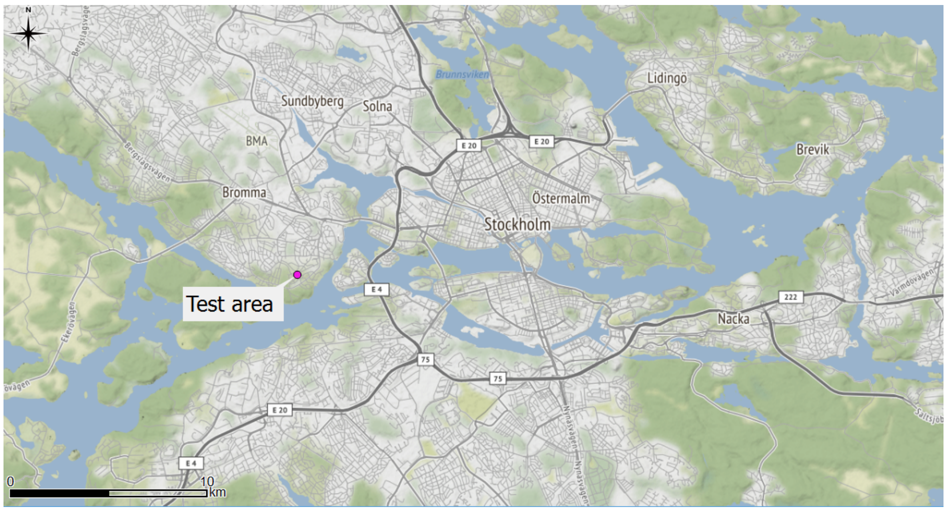

Figure 5.1: Map showing the location of the test area (© OpenStreetMap)

Problematic areas in mobile mapping are areas where GNSS fix cannot be achieved for a longer period of time. This can be because GNSS signals are blocked by high buildings, trees or a tunnel. As mentioned in 2.2, this leads to drift in INS measurements as they are not aided by GNSS observations. Because of the drift, it can difficult to achieve an accuracy under $0.05m$.

To simulate an area without good GNSS coverage, *test trajectories* were made by removing GNSS observations for a period of time in the calculation of the trajectory. By doing this, a *reference trajectory* could be obtained by including GNSS observations throughout the trajectory period.

The reference trajectory was used as a reference to evaluate the effect of re-calculating the trajectory using SLAM. This was done by comparing the original trajectory and the re-calculated trajectory to the reference trajectory.

Other than the removal of GNSS observations in a part of the test trajectories, all trajectories were calculated using the same raw data and settings. Before and after the period without GNSS observations, GNSS observations were available in the calculation of the trajectory. Hence, the drift in position and orientation is small at the beginning and end of the period without GNSS observations.

5.1 Back-End SLAM

5.1.1 Test Area

Back-end SLAM is used to get a loop closure effect. Thus, a part of the trajectory with a loop was chosen for testing, see figure 5.2.

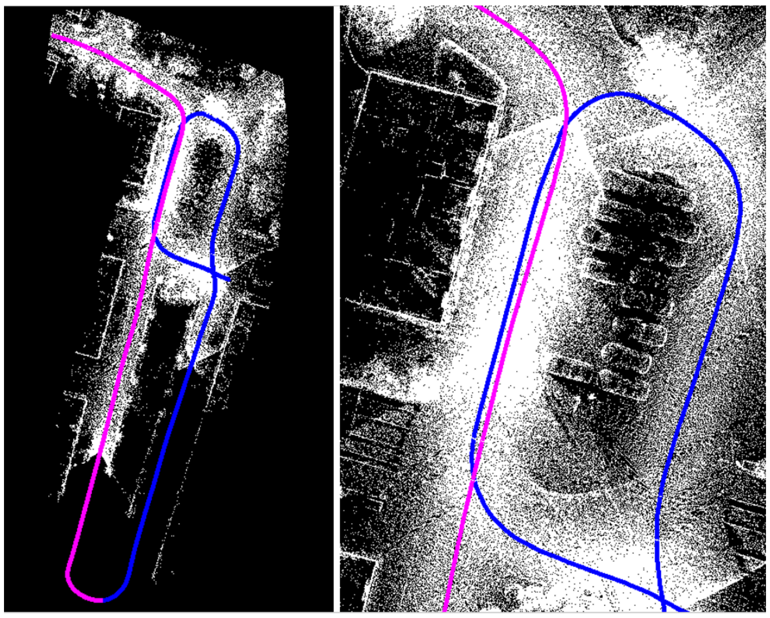

Figure 5.2: The left image shows the test area. The right image shows the part where there is overlap close up. The blue part of the trajectory does not have GNSS observations. (TerraMatch)

GNSS observations were included for the pink strip, but removed for the blue strip in figure 5.2. The blue strip starts at TOW 206472 and ends at TOW 206584, which makes the part without GNSS observations 113 seconds long.

To see the effect of loop closure, tie point observations were found in the part where the pink and blue strip overlaps. This area was scanned at two different periods of time, once when the platform was in the pink part of the trajectory and later when it was on the blue part. The hypothesis was that the pink strip with higher accuracy could be used to improve the positioning in the blue strip, were GNSS observations were removed.

Difference between the original test trajectory and the reference trajectory can be seen in figure 5.3. Difference in north direction was $0.143m$ at its largest. In east direction, it was $0.157m$. The largest difference in down direction was $0.074m$. It can be seen that the difference approaches zero at the beginning and end of the period without GNSS observations. The zero velocity update in the middle of the test period reduces drift.

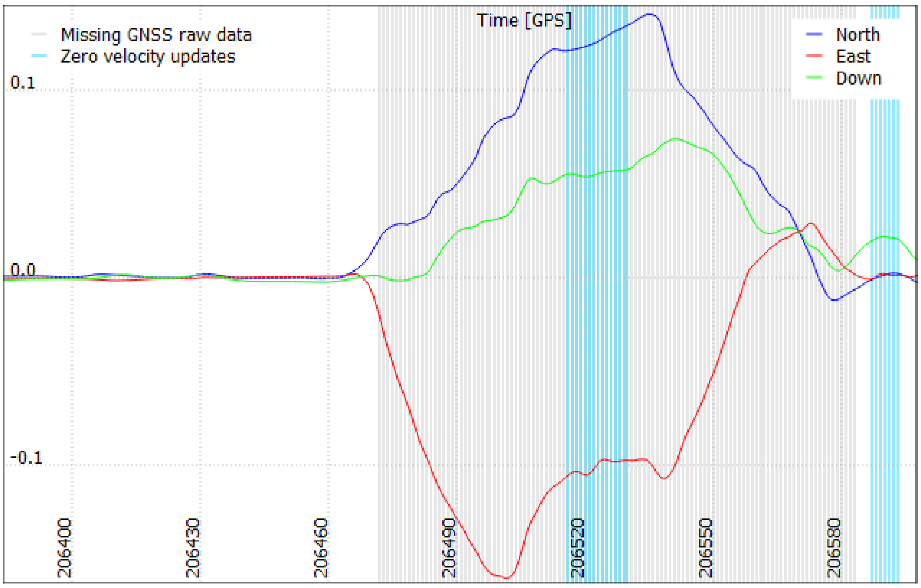

Figure 5.3: Difference in meters between original test trajectory and reference trajectory. The part without GNSS observations is gray, zero velocity updates are shown in blue (TerraPos)

Standard deviation of the reference trajectory in the test period is given in figure 5.4. The standard deviation was approximately $0.03m$ in down direction and $0.01-0.02m$ in north and east direction.

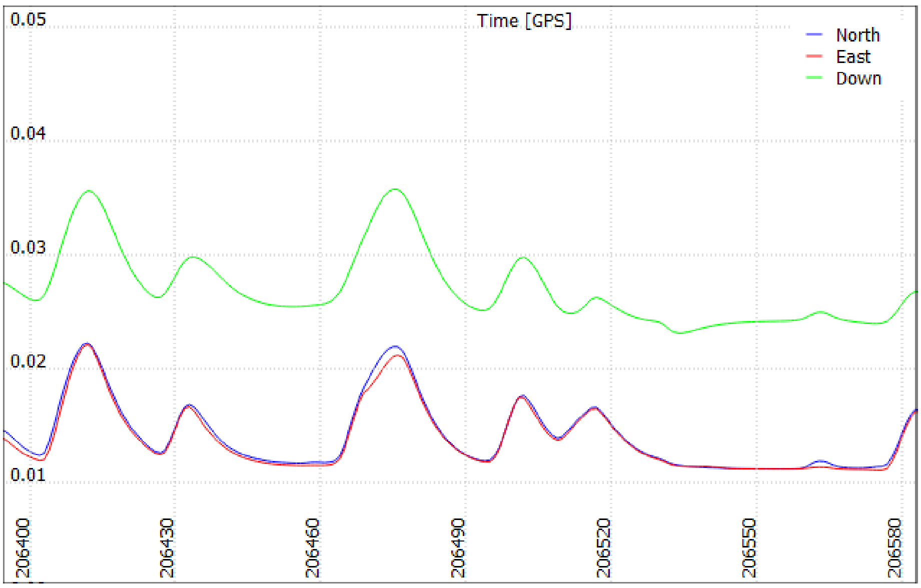

Figure 5.4: Standard deviation in meters of the position of the reference trajectory in the test area (TerraPos)

5.1.2 Results

As described in chapter 4.3, tie point observations can be made from different point cloud observations. The tie point observations are different implementations of similar observations, and they are expected to give similar results.

If a distinct object in the overlapping area exists, e.g. a manhole, a 3-dimensional tie point can be useful, as only one tie point is needed to get offset in all directions. This tie point type is based on XYZ observations in TerraMatch and will need to be found manually. If there are both horizontal and vertical surfaces (both in north and east direction), normal vector observations work well and can be found automatically. This was available by the road and buildings in the overlapping area in figure 5.2. Thus, 1-dimensional tie point observations made from normal vector observations are used in the tests presented.

Tests were done to investigate how many tie points should be included and whether or not to include tie point observations with a long range. Finally, a discussion on the benefit of this method when the overlapping strips both either have or do not have GNSS available is included.

Include tie point observations with a long range?

As discussed in section 2.3, errors grow as range (distance) from the scanner to the scanned surface grows. Tests were done to investigate if only tie point observations with a short range should be included. 2 tie points were included in down direction (height) and 2 in north and east direction for the test. The point cloud observations from TerraMatch used to compute the 4 tie point observations are given in table 5.1. The ones marked in yellow have a range larger than $10m$.

In the first test, only tie point observations with a range shorter than $10m$ were included. Tie point 1 and 2 are from a vertical wall, giving tie point observations in north and east direction. Tie point 3 and 4 are from the ground, giving tie point observations in down direction.

Table 5.1: Point cloud observations from TerraMatch. Observations with a range longer than $10m$ are marked in yellow.

Tie point	TOW	dNE	dN
1	205852.0	0.002	-
1	206419.0	0.009	-
1	206501.0	0.176	-
1	206563.8	0.018	-
2	205853.3	0.001	-
2	206409.8	0.023	-
2	206502.1	0.161	-
2	206508.3	0.150	-
2	206560.4	0.016	-
3	206408.9	-	-0.007
3	206510.1	-	+0.039
3	206559.5	-	+0.031
4	206413.1	-	-0.003
4	206561.6	-	+0.021

The result of the first test is visualized in figure 5.5 by the difference between the re-calculated test trajectory and the reference trajectory. Dark green vertical lines show the time of observations used and the gray area is the part without GNSS observations.

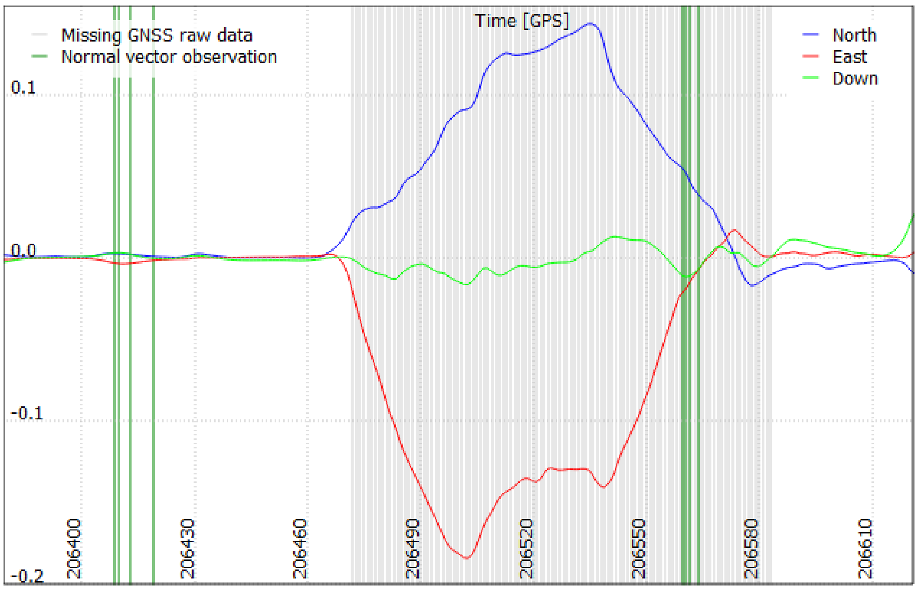

Figure 5.5: Difference in meters between re-calculated test trajectory and reference trajectory when including only point cloud observations that were closer to the scanner than $10m$. (TerraPos)

When comparing the difference in figure 5.5 with the original test trajectory in figure 5.3, it can be seen that tie point observations closer than $10m$ lead to an improvement in down direction (the green line) in this test. The difference from the reference trajectory was reduced from a maximum of $0.074m$ in down direction in the original test trajectory to $0.017m$ in the re-calculated test trajectory.

The difference from the reference trajectory in east and north direction did not show any improvement. Tie point observations in table 5.1 give the reason for this. The error in position in the overlapping area (the white rows) was only a couple of centimeters in north and east direction. These tie point observations did not provide the Kalman filter any information about the large drift between them. This drift can be seen in the tie point observations that have a larger range, with an offset of $0.150 - 0.180m$.

A second test was done with all observations in table 5.1. The yellow observations were scanned from approximately $30m$ away, introducing more errors in the observations. However, these observations give information about drift between the overlapping parts of the trajectory. The result of the test is visualized in figure 5.6. The dark green vertical lines represent the time of tie point observations. It can be seen that there are some additional tie point observations (between TOW 206490 and 206520) compared to the first test. These tie point observations were made between the periods of time where the platform was in the overlapping part of the trajectory.

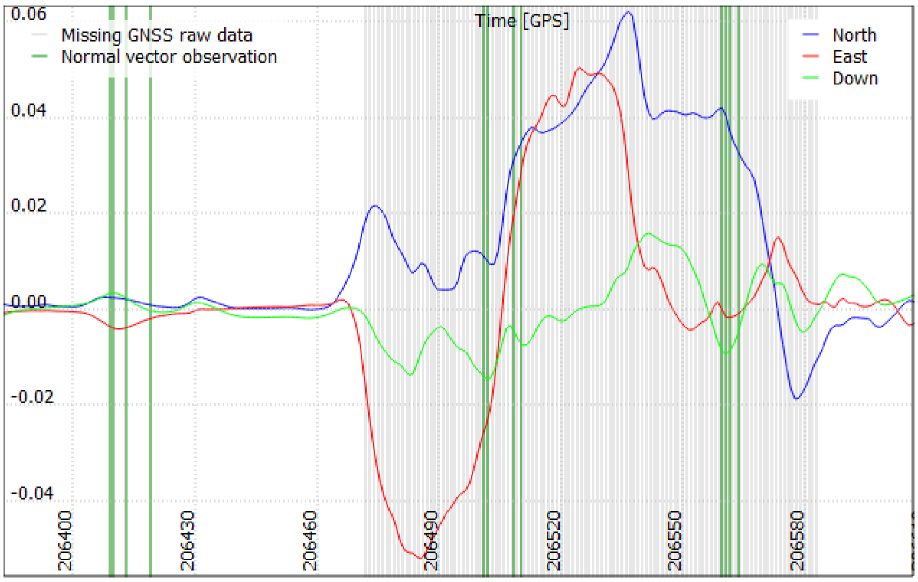

Figure 5.6: Difference between re-calculated test trajectory and reference trajectory when including observations further away than $10m$. (TerraPos)

Maximum differences were reduced to approximately $0.05-0.06m$ in east and north direction. The results of this test indicate that including observations that are done far away can be helpful, even when observations are likely to have larger errors.

Another effect worth noting is that the part of the trajectory with a low precision was the part that was mostly affected by adjustments. The part of the trajectory that was estimated with INS aided by GNSS observations did not change much by the re-calculation of the trajectory with the point cloud observations.

What is the effect of more tie points?

A large number of tie point observations can be computed from point cloud observations in TerraMatch. TerraPos can only benefit from one observation per second, limiting number of possible observations. The number of tie points that can be handled by the SLAM algorithm in TerraPos is also limited. As mentioned in 3.2, each tie point is added to the state vector in the Kalman filter, meaning that a large number will have a negative impact on processing time. Hence, it is of interest to investigate how the effect of adjustment changes with the number of tie points.

Four test trajectories were introduced for testing this, all with a different number of tie points, see table 5.2. Both tie point observations with short and long ranges were included.

Table 5.2: Test trajectories with different number of tie points

Test trajectory	Number of tie points north-east	Number of tie points height
1	0	0
2	1	1
3	2	2
4	4	3

The result of the re-calculated test trajectories can be seen in figures 5.7 to 5.9. Figure 5.7 shows difference between test trajectories and reference trajectory in north direction in blue. Figure 5.8 shows the same in east direction in red. Figure 5.9 visualizes difference in down direction in green.

Results from all test trajectories indicate that tie points used to aid positioning in areas without GNSS observations give a better accuracy of the trajectory, than not to include them. This can be seen by the fact that test trajectory 1 had the largest difference to the reference trajectory. To include 2-4 tie points seemed to be better than to include 1 tie point (test trajectory 2) in this test, but the improvement was not significant. The difference between including 2 (test trajectory 3) and 3-4 tie points (test trajectory 4) was very small in this test. Results indicate that 2-3 tie points are beneficial for about 2 minutes without GNSS observation.

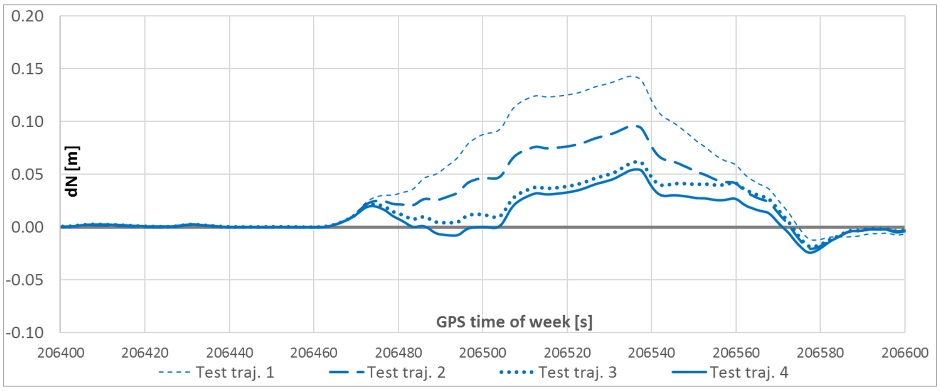

Figure 5.7: Difference in north bdirection between test traj. and ref. traj.

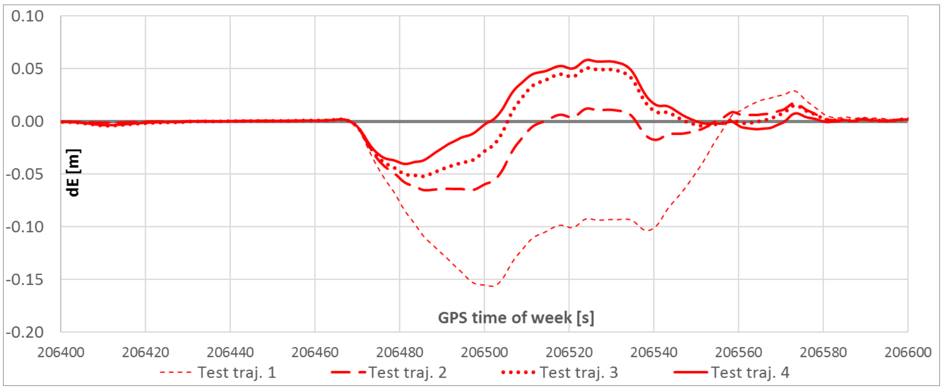

Figure 5.8: Difference in east direction between test traj. and ref. traj.

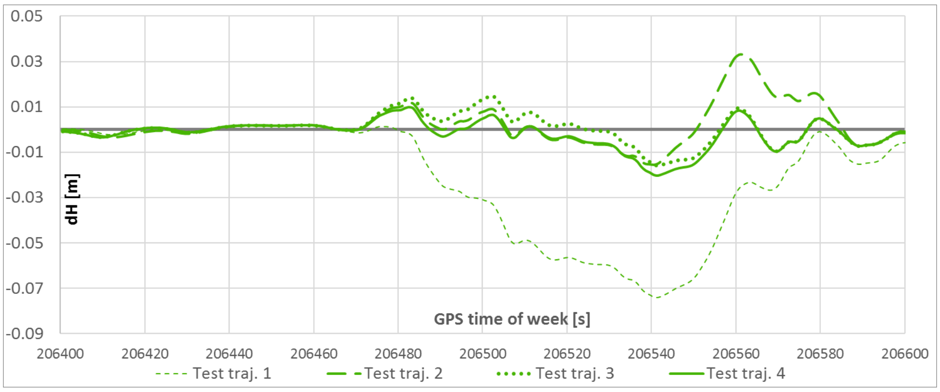

Figure 5.9: Difference in down direction (height) between test traj. and ref. traj.

Figure 5.10 shows standard deviation of test trajectory 4 in the part without GNSS observations. It was at its maximum 0.126m for north, 0.094m for east and 0.101m for down direction. For test trajectory 3 the same values were between 0.094m and 0.160m and for test trajectory 2, 0.130m to 0.225m. For test trajectory 1, the standard deviation was 0.240m for north, 0.340m for east and 0.420 for height at its largest. Thus, precision of the trajectory was improved also between the times of the tie point observations. It can be seen that the standard deviation was lower where there were tie point observations. This result indicates that more tie points are preferred for higher precision.

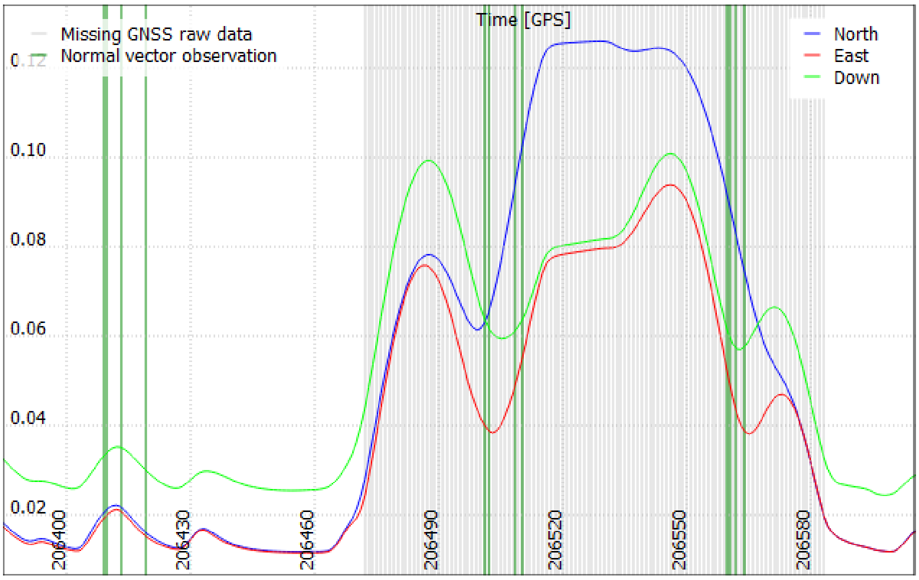

Figure 5.10: Standard deviation in meters of test trajectory 4. Vertical green lines indicate tie point observations. (TerraPos)

Beneficial for other cases?

What has been tested so far is a loop where standard deviation is low is one part of the trajectory and high when this area is passed again. The position of the tie point observation from the part of the trajectory with high precision is used to correct the part with low precision. The two alternatives to this case, are two overlapping parts of the trajectory that both have high or low precision.

If two parts of the trajectory with low precision is overlapping, the likelihood of the mean being significantly closer to the correct solution in not very high. With a larger number of tie point observations of each tie point, one could make the assumption that the tie point observations would be normally distributed and that the mean would be close to the true value, but in mobile mapping, the number of observations is normally not that high. Another issue is unmodeled systematic errors affecting all tie point observations the same way. This has been tested with tie points in a tunnel [Johnsbråten, 2015]. Unmodeled systematic errors from gravity in the tunnel led to errors in down direction dragging in the same direction for all tie point observations. Because of this, aid from tie points did not improve the solution.

If two parts of the trajectory with high precision is overlapping, offset between the two is likely to be small. A test was done to investigate how the effect of adjustment with tie point observations in back-end SLAM would affect the trajectory in this case. To test this, GNSS observations were included for both the pink and the blue part of the test trajectory in figure 5.2. Thus, the original test trajectory for this test was the same as the reference trajectory. The offsets found by tie point observations in TerraMatch for this trajectory were small, ranging from a few millimeters to a couple of centimeters.

The difference between the re-calculated test trajectory and the reference trajectory is visualized in figure 5.11. Without external reference data, there is no way do know which one of the re-calculated test trajectory and the reference trajectory is more similar to the true trajectory in this test. The difference between them was within $0.02m$. Offset in all directions was within the standard deviation of the reference trajectory, given in figure 5.4. This indicates that including tie point observations when the trajectory of both has high precision does not corrupt the estimation.

Standard deviation of the re-calculated test trajectory, see figure 5.12, was similar to the standard deviation of the reference trajectory, see figure 5.4. The standard deviation was however reduced in parts where there were tie point observations. In the part between TOW 206490 and 206520, the standard deviation is reduced from about $0.015m$ in north and east direction and $0.025 - 0.030m$ in down direction for the reference trajectory to $0.013m$ and 0.017 respectively for the re-calculated test trajectory. This indicates that tie points increase the precision of the estimation, even when there are GNSS observations aiding the INS in both observations of the tie point.

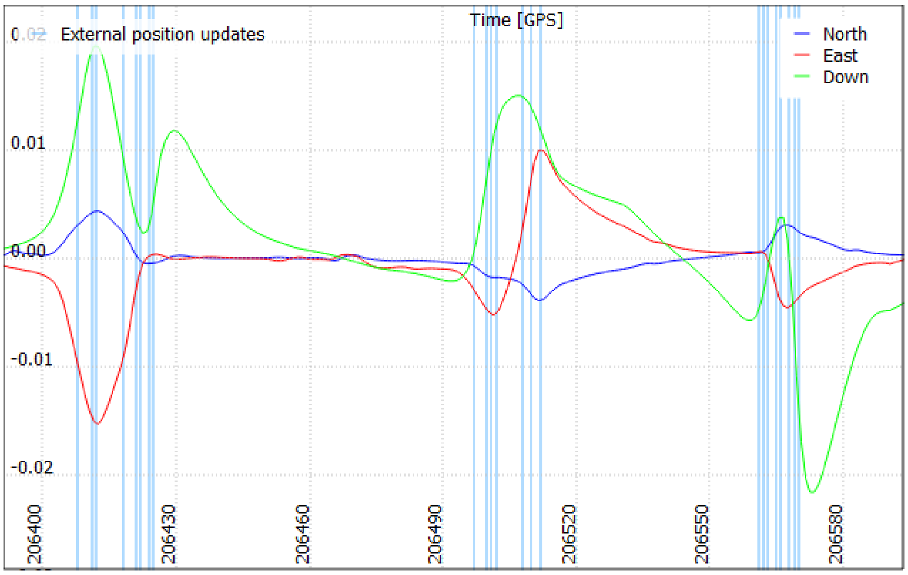

Figure 5.11: Difference in meters between the re-calculated test trajectory with GNSS observations throughout the trajectory and the reference trajectory. The light blue vertical lines indicate tie point observations. (TerraPos)

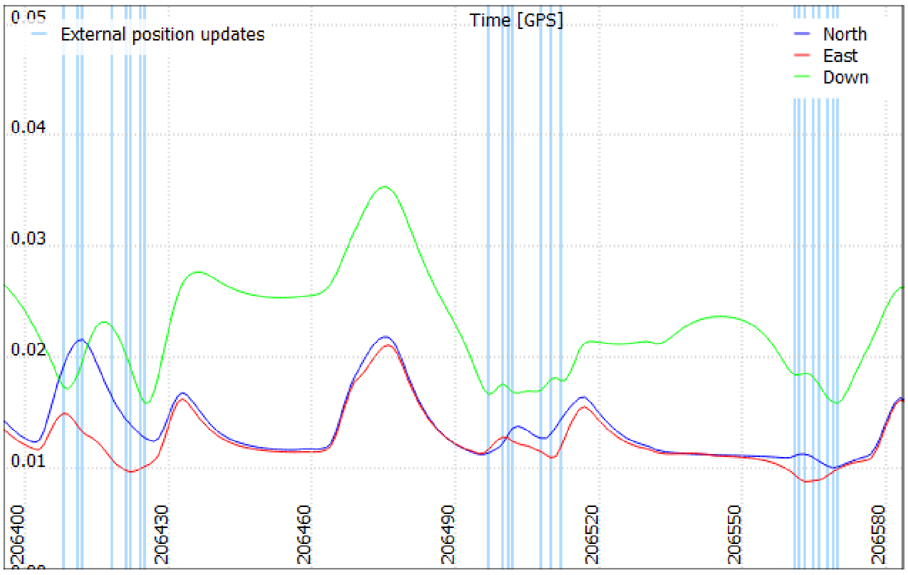

Figure 5.12: Standard deviation in meters of trajectory with GNSS observations throughout the trajectory and the reference trajectory. The light blue vertical lines indicate tie point observations. (TerraPos)

5.2 Front-End SLAM

5.2.1 Test Area

Front-end SLAM utilizes incremental height observations to aid the inertial navigation. Thus, a road with continuous point cloud observations was found, see figure 5.13.

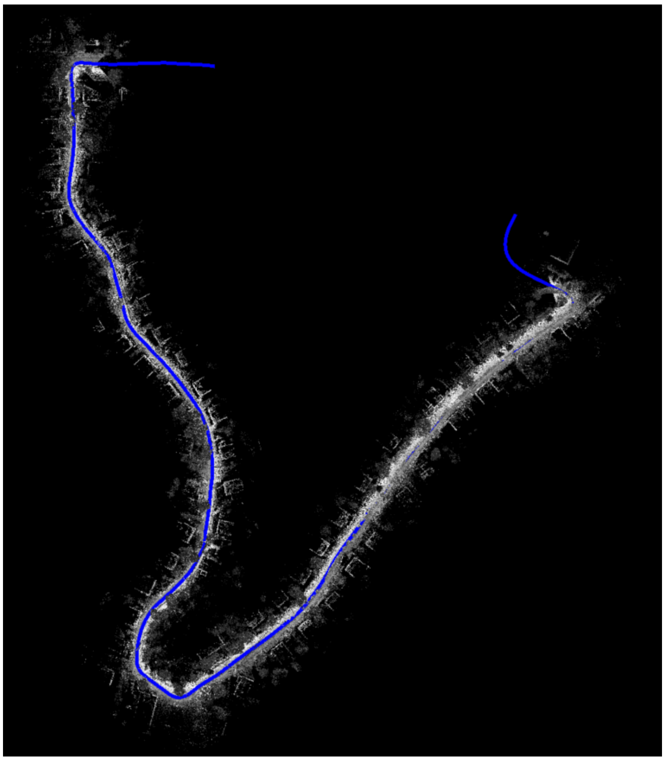

Figure 5.13: Test area with the trajectory shown as the blue line (TerraMatch)

GNSS observations in the time period 205775-205950 (GPS TOW) was removed for the test trajectory used for the front-end SLAM test, giving a period of 2 minutes and 55 seconds without GNSS observations. Difference between the original test trajectory and the reference trajectory was found and is visualized in figure 5.14. Difference in down direction is at the most $0.15m$ (the green line), while the original test trajectory in north and east direction (the blue and the red line) has maximum differences of $0.2 - 0.4m$. Zero velocity updates were done short time before and after the removal of GNSS observations, improving estimation of orientation before and after the test period. Standard deviation for the reference trajectory in the test area is given in figure 5.15.

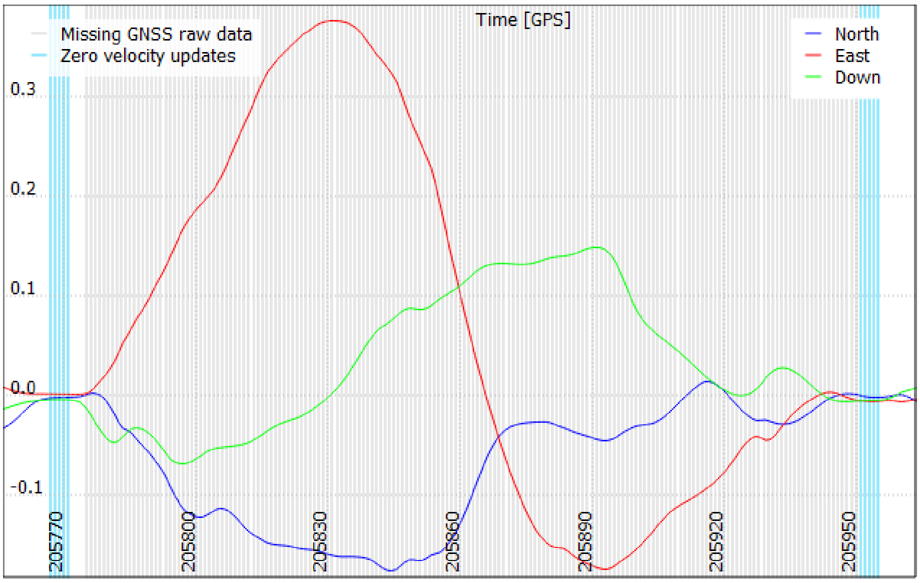

Figure 5.14: Difference in meters between original test trajectory (without incremental observations) and reference trajectory. The part without GNSS observations is gray, zero velocity updates are shown in blue (TerraPos)

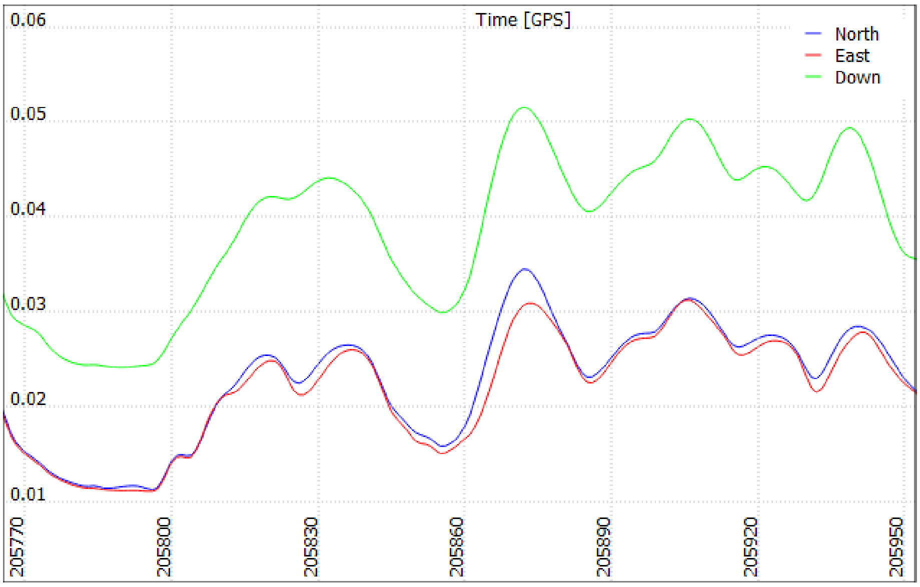

Figure 5.15: Standard deviation in meters of the position of the reference trajectory in test area (TerraPos)

Surface line observations were collected every 0.25 seconds, and each surface line represented an area of $0.5 * 0.25m$ on the road surface. Only observations within $10m$ of the trajectory in y-direction in b-frame (right/left of the platform) and $0.5m$ from the road surface in z-direction in b-frame (up/down from the platform) were included. This resulted in observations by scanner 1 and scanner 2 in 15515 surface lines in the test area.

5.2.2 Systematic Errors

The test trajectory was re-calculated with incremental height observations made from the surface line observations found in TerraMatch. The resulting trajectory still had a large difference to the reference trajectory, see the dashed line in figure 5.18. The improvement was expected to be larger. Hence, systematic errors in the system that could affect the improvement were investigated.

In an attempt to find systematic errors and sources of the errors, several different parameters were investigated. In figure 5.16, correlation between $dz = z_{ObsByScanner1} - z_{ObsByScanner2}$ and distance from scanner 1 in y-direction is visualized. These observations will sometimes be positive and sometimes negative, depending on which scanner scanned the point first. Observations more than $10m$ from the trajectory were not included, as observations further away are more sensitive to other errors, and in this test, only systematic errors were of interest. In a system without systematic errors, one would expect the median of the observations of differences to be approximately zero in all distances from the scanner, but this is clearly not the case in the plot.

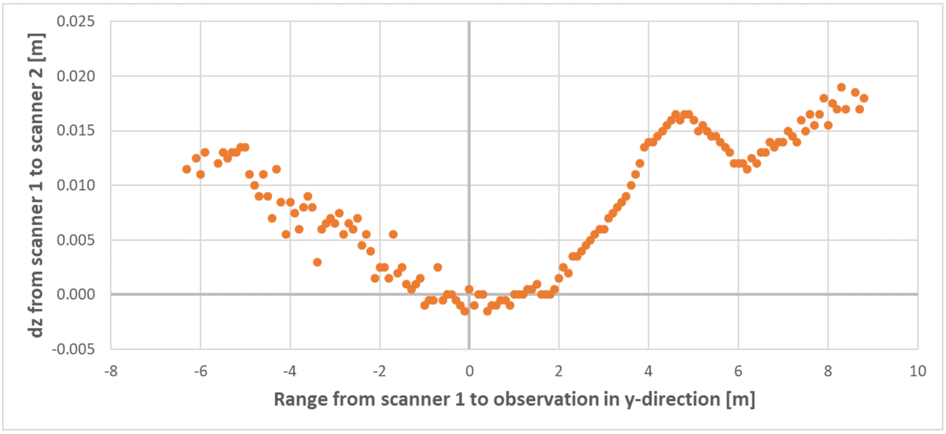

Figure 5.16: Correlation between the median of dz between observation of scanner 1 and scanner 2 and the range from scanner 1 to the observation in y-direction

Figure 5.17 shows correlation between $dz = z_{ObsByScanner1} - z_{ObsByScanner2}$ and distance from scanner 1 to the surface line observation. dz is expected to be approximately zero in a system without systematic errors also for this case. The plot shows that this was not the case and that the trend is not even linear.

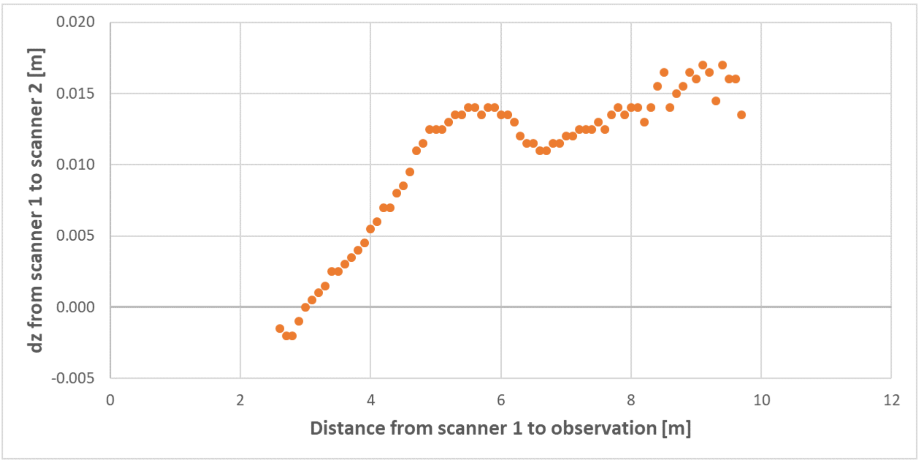

Figure 5.17: Correlation between the median of dz between observation of scanner 1 and scanner 2 and the distance from scanner 1 to the observation

These tests could indicate that there was more that one systematic error in the system, which makes it harder to identify the exact causes of them.

A model describing the systematic error by the trend of systematic errors found by the distance to the scanner was made and applied to the observations.

5.2.3 Results

Figure 5.18 shows difference between the test trajectories and the reference trajectory. The dashed line is showing the test trajectory without aid from GNSS or incremental height observations. This line is the same as the green line in figure 5.14. The dotted line shows the trajectory aided by incremental height observations that are not corrected for systematic errors.

The solid line in figure 5.18 is the result of applying correction for systematic errors to the incremental height observations. The trajectory with incremental height measurements corrected by systematic errors is no more than $0.03m$ from the reference trajectory. The result from this test indicates that including incremental observations that are corrected for systematic errors improves accuracy of the trajectory.

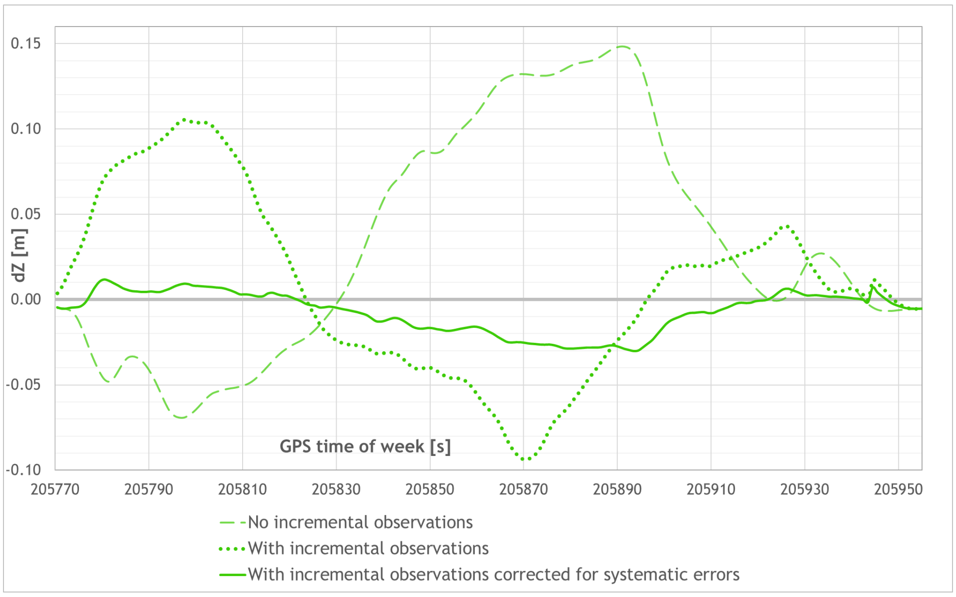

Figure 5.18: Difference in meters in down direction between test trajectories and reference trajectory

Standard deviation of the test trajectory was reduced from $0.995m$ (see figure 5.19) in down direction without incremental height observations to $0.058m$ (see figure 5.20) with incremental height observations corrected for systematic errors. The achieved standard deviation is not far from the one for the reference trajectory, given in figure 5.15. This indicates improvement in trajectory calculation by introducing incremental height observations.

Incremental height observations can be helpful in areas where the standard deviation is low. It would have been interesting to test both back-end and front-end SLAM together, but in the loop used for testing in back-end SLAM, laser scanning was not made for the entire loop (see figure 5.2), so this was not possible. To take advantage of front-end SLAM in mobile mapping it is important to scan for longer periods of time.

This observation type can be helpful where tie point observations are affected by unmodeled systematic errors, for instance in tunnels. The INS is affected by the gravity, and this can lead to drift, but incremental height observations based on point cloud observations will not be affected in the same way.

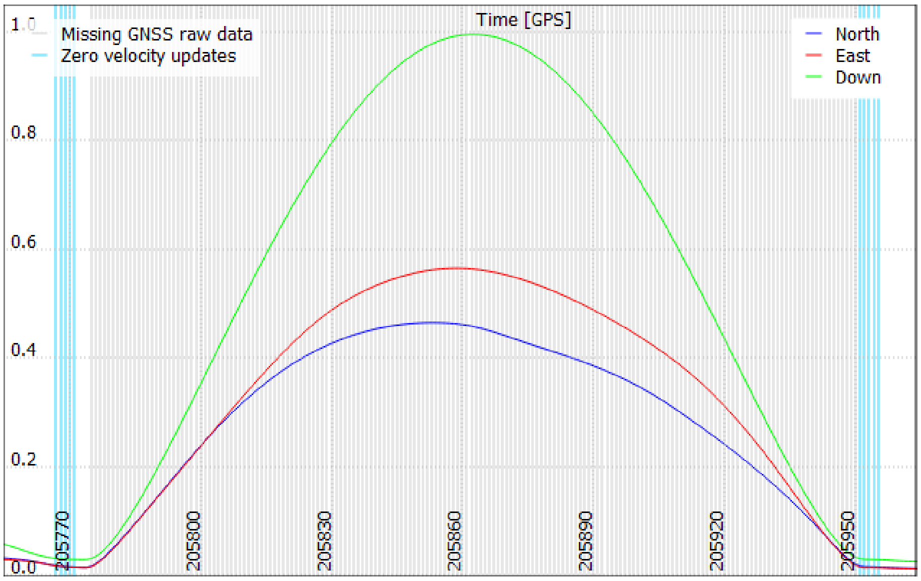

Figure 5.19: Standard deviation in meters of the position of the test trajectory before adjustment

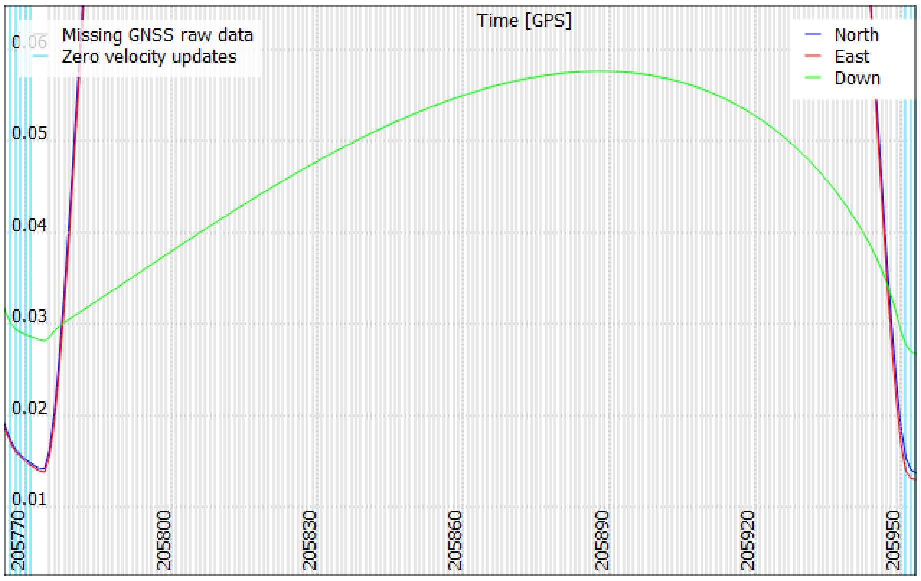

Figure 5.20: Standard deviation in meters of the position of the re-calculated test trajectory (incremental observations corrected for systematic errors)

Chapter 6

Summary and Conclusion

It was of interest to develop potential methods to introduce SLAM in post-processing of mobile mapping data. These methods should require a limited amount of extra man hours and computational cost compared to not using SLAM. To do this, point cloud observations were produced in TerraMatch and then transformed to observations that could be utilized in re-calculating the trajectory in TerraPos. Finally, a method for applying adjustments from the re-calculated trajectory to the point cloud was developed.

The positioning of the mobile mapping system used for this study is done by aided inertial navigation. Measurements made by the INS drifts over time. Thus, an accuracy of the trajectory of a few centimeters cannot be maintained for long by the INS itself. Measurements from GNSS and an odometer are used to aid the INS. The INS measurements are more accurate than measurements made by GNSS, but GNSS measurements do not drift over time and maintain long-time stability.

In areas where GNSS observations are not available, drift of the INS is still a problem. This can, for instance, happen in city areas with high buildings, under trees and in tunnels, which all are areas commonly mapped by mobile mapping systems. To limit the need for adjustment points from terrestrial land surveying, another type of measurement is needed to aid the INS.

SLAM is the problem of estimating a map of an unknown environment while estimating the system's position in this environment. Laser scanner measurements can be used to get information about relative movement of the platform of the mobile mapping system. This can help positioning of the platform using SLAM algorithms. When a tie point, a point features in the environment, is observed more than once, the tie point observations can be used aid the INS. Keeping track of what tie point a tie point observation belongs to, is known as data association. Detection and validation of the correct tie point for observations is crucial in many SLAM algorithms. Point cloud observations from TerraMatch are used to obtain tie point observations with the correct data association. Point cloud observations are found automatically in TerraMatch, and a software package was developed to transform them to tie point observations that can be used when re-calculating the trajectory by the EKF SLAM algorithm implemented in TerraPos.

Generation of point clouds is a time-consuming process. Thus, re-generating the point cloud to get an adjusted one was not an option. This was solved by finding the difference between the original and the re-calculated trajectory. This difference can be applied to the point cloud in TerraMatch, achieving an adjusted point cloud.

Back-end SLAM takes advantage of the loop closure effect. Loop closure events happen when tie point observations of the same tie point are made with some time between them. The effect of loop closure is improved global consistency in the map, as well as reducing uncertainty in the trajectory. Relative surface line observations can be automatically found in TerraMatch and transformed to tie point observations that can be utilized in TerraPos.

Front-end SLAM uses incremental height observations obtained from the time difference between scanning the same point by laser scanner 1 and 2. These observations work as a visual odometer and are used to improve local consistency in the map. Height observations were chosen as it is possible to get continuous point cloud observations of the road surface. Observations in north and east direction would also be beneficial, and could, for instance, be made in an area with continuous buildings near the road or in a tunnel.

For the testing done, poor GNSS coverage was simulated by removing the GNSS measurements for a period of time.

Results from back-end SLAM

An area with a loop in the test area was found. Half of the loop had GNSS measurements included in estimation of the trajectory, in the other half GNSS measurements were removed.

The first test done investigated how point cloud observations with a long range from the scanner to the scanned surface would contribute to the estimation. Tie point observations with long ranges are more sensitive to errors. It was found that regardless of this, tie point observations with long ranges improved the re-calculated trajectory in tests done.

The second parameter investigated was how many tie points should be used in an overlapping area. Tests were done with 0-3 tie points. The result of the test done indicated more tie points gave a better accuracy of the trajectory and reduced the standard deviation of it. However, the difference between 2 and 3 tie point was small, indicating that the effect of more than 2 tie points is not high in periods without GNSS observations for 113 seconds. Maximum difference from the reference trajectory for the part without GNSS observations was reduced from

$0.15m$ in north and east direction and $0.07m$ in down direction to $0.05m$ and $0.02m$ respectively after adjustment.

Results of the tests done show that the part of the trajectory without GNSS observations was mostly affected by tie point observations, while the part of the trajectory with GNSS observations remained similar to the original trajectory.

The effect of this method is not likely to be high if all tie point observations for a tie point have low precision. The effect is likely not to be that high for only tie point observations with high precision either. The test done for this case indicated that the difference between including the observations and not is small. It was seen that the standard deviation was reduced where there were tie point observations aiding the positioning.

Results from front-end SLAM

GNSS observations were removed for 2 minutes and 55 seconds. Incremental height observations found by the difference between point cloud observations by scanner 1 and scanner 2 were used to test this method.

It was found that removal of systematic errors in the system was important to get an improvement by incremental height observations. Even small systematic errors in incremental tie point observations will lead to drift.

Results from the test done showed improvement of accuracy of the trajectory. Without incremental observations, the difference between the test trajectory and the reference trajectory $0.15m$ at the largest. When incremental observations corrected for systematic errors were used in the trajectory re-calculation, the largest difference was reduced to $0.03m$.

Conclusion

Two methods of introducing SLAM in trajectory calculation have been introduced by this thesis. Tests done on both the method using front-end SLAM and the method using back-end SLAM show an improved accuracy and precision of the trajectory compared to not using SLAM in the post-processing. Tests are done with a period of 2-3 minutes without GNSS observations and show a potential accuracy of $0.02 - 0.03m$ in height for both methods. This is sufficiently good in many mobile mapping projects.

The methods lead to some extra work in post-processing but have the potential to reduce the need for points from terrestrial land surveying. Further investigation is needed to test the methods in different types of areas and projects to figure out when the use of back-end and front-end SLAM is reasonable.

6.1 Further work

In back-end SLAM, only one tie point observations each second can be included in the estimation in TerraPos. If the overlapping area used to collect tie point observations for loop closure is small, for instance crossing roads, it may be difficult to get a sufficient amount of 1-dimensional surface line observations to make multiple tie point observations for all directions. Algorithms to automatically make 3-dimensional point cloud observations could be beneficial in this case and should to be developed. Support for multiple tie point observations each second is another possible improvement.

To have automatic methods for collecting incremental point cloud observations for north and east direction would be beneficial. Potential surfaces are curbstones, buildings and painted lines on the road. There is functionality to get observations of painted lines in the road in TerraMatch, but it was not tested as there were no such lines in the test area used for this study. To get vertical surface lines from buildings is also possible in TerraMatch. It is however preferred to have continuous observations when using incremental observations to aid the INS, and this can be difficult with buildings. It could be done in dense city areas, where there are buildings close to the road. Tunnels are likely to benefit from front-end SLAM and observations of both the road and walls can be made.

A method for finding systematic errors in the mobile mapping system and laser scanners is needed to make the method automatic. This might be information that can be periodically found by calibration and used when processing data from the system.

Bibliography

[2013] Puente, I., González-Jorge, H., Martínez-Sánchez, J., Arias, P., (2013). *Review of mobile mapping and surveying technologies.* Measurement, 46(7), pp. 2127-2145.

[El-Sheimy, 1996] El-Sheimy, N., (1996). *A mobile multi-sensor system for GIS applications in urban centers.* In: The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vienna, Austria, Vol. XXXI, Part B2, pp. 95-100.

[Løvås, 2016] Løvås M. (2016) Simultaneous Localization And Mapping with GNSS-INS integration in TerraPos. Specialization Project. NTNU

[Farrel, 2008] Farrel, J.A. (2008) *Aided Navigation: GPS with high rate sensors.* 3rd edition, p.23-25. USA: McGraw-Hill

[Dudek and Jenkin, 2008] Dudek G. and Jenkin M. (2008). Inertial Sensors, GPS, and Odometry. Siciliano B. and Khatib O. (Eds.), *Handbook of Robotics.* 1st edition, p.477-484. Berlin: Springer-Verlag

[Oxford Technical Solutions, 2014] Oxford Technical Solutions (2014) Available at: http://www.oxts.com/technical-notes/why-use-a-survey-grade-inertial-navigation-system-on-an-unmanned-aerial-vehicle/ [Accessed 16 May 2017]

[Farrel, 2008] Farrel, J.A. (2008) *Aided Navigation: GPS with high rate sensors.* 3rd edition, p.386-388. USA: McGraw-Hill

[BeiDou Navigation Satellite System, 2017] BeiDou Navigation Satellite System. 2017. *Development Plan.* Available at: http://www.beidou.gov.cn/2012/12/14/2012121481ba700d7ca84dfc9ab2ab9ff33d2772.html. [Accessed 4 June 2017].

[ESA, 2017] ESA. 2017. *What is Galileo?* Available at: http://www.esa.int/Our_Activities/Navigation/Galileo/What_is_Galileo. [Accessed 4 June 2017].

[Skogset and Norberg, 2014] Skogseth T., Norberg D. (2014) *Grunleggende landmåling.* 3rd edition, figure 4.11. Norway: Gyldendal undervisning

[Misra and Enge, 2012] Misra P., Enge P. (2012) Global Positioning System: Signals, Measurements, and Performance. Revised 2nd edition, p.151-154. Massachuesetts: Ganga-Jamuna Press

[Misra and Enge, 2012] Misra P., Enge P. (2012) Global Positioning System: Signals, Measurements, and Performance. Revised 2nd edition, p.269-272. Massachuesetts: Ganga-Jamuna Press

[Verhagen, 2005] Verhagen, S. (2005).*The GNSS integer ambiguities: estimation and validation* PhD thesis, Delft University of Technology

[IGS, 2017] IGS. 2017. *Products.* Available at: http://www.igs.org/products/data. [Accessed 4 June 2017].

[Hofmann-Wellenhof, Lichtenegger and Wasle, 2008] Hofmann-Wellenhof B., Lichtenegger H., Wasle E. (2008) *GNSS - Global Navigation Satellite Systems : GPS, GLONASS, Galileo and More.* 1st edition, p.155. Wien: Springer

[Statens Kartverk, 2009] Statens Kartverk, Geodesidivisjonen (2009). *Satelittbasert posisjonsbestemmelse.*

[Vosselman and Maas, 2010] Vosselman G., Maas H.G. (2010) Airborne and Terrestrial Laser Scanning. p.3. Scotland: Whittles Publishing

[Vosselman and Maas, 2010] Vosselman G., Maas H.G. (2010) Airborne and Terrestrial Laser Scanning. p.12. Scotland: Whittles Publishing

[Horemuz, 2015] Horemuz, M. (2015) *Airborne Laser Scanning.* Royal Institute of Technology (KTH).

[Farrel, 2008] Farrel, J.A. (2008) *Aided Navigation: GPS with high rate sensors.* 3. edition, p.190. USA: McGraw-Hill

[Brown and Hwang, 2012] Brown, R.G., Hwang, P.Y.C. (2008) *Introduction to Random Signals and Applied Kalman Filtering.* 4. edition, p.143-151. USA: John Wiley and Sons

[Brown and Hwang, 2012] Brown, R.G., Hwang, P.Y.C. (2008) *Introduction to Random Signals and Applied Kalman Filtering.* 4. edition, p.121-122. USA: John Wiley and Sons

[Brown and Hwang, 2012] Brown, R.G., Hwang, P.Y.C. (2008) *Introduction to Random Signals and Applied Kalman Filtering.* 4. edition, p.228-231. USA: John Wiley and Sons

[Thrun and Leonard, 2008] Thrun S. and Leonard J. (2008). Simultaneous Location and Mapping. Siciliano B. and Khatib O. (Eds.) *Handbook of Robotics.* 1st edition, p.871-885. Berlin: Springer-Verlag

[Thrun, Burgard and Fox, 2005] Thrun, S., Burgard, W., Fox, D (2005) *Probabilistic Robotics.* p.310-332. Cambridge, Massachusetts: MIT Press

[Cadena *et al.*, 2016] Cadena C., Carlone L., Carrillo H., Latif Y., Scaramuzza D., Neira J., Reid I., Leonard J.J. (2016). Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age, *IEEE Transactions on Robotics*, 32(6), p.1309–1332. doi: 10.1109/TRO.2016.2624754

[Cadena *et al.*, 2016] Cadena C., Carlone L., Carrillo H., Latif Y., Scaramuzza D., Neira J., Reid I., Leonard J.J. (2016). Past, Present, and Future of Simultaneous Localization And Mapping: Towards the Robust-Perception Age, *IEEE Transactions on Robotics*, 32(6), p.1309–1332, fig. 1. doi: 10.1109/TRO.2016.2624754

[Kjørsvik, 2010] Kjørsvik, N. S. (2010).*Introduction to inertial navigation.* Technical report, Norwegian University of Life Sciences (NMBU).

[Trimble, 2017] Trimble. (2017). *Datasheet: Trimble AP60* Available at: https://www.applanix.com/downloads/products/specs/AP60_DS_NEW_0406_YW.pdf. [Accessed 22 June 2017].

[Teledyne Optech, 2017] Teledyne Optech. (2017). *Lynx SG Mobile Mapper: Summary Specification Sheet* Available at: http://www.teledyneoptech.com/wp-content/uploads/Lynx-Specsheet-SG-160404.pdf. [Accessed 22 June 2017].

[TerraTec AS, 2016] TerraTec AS. (2016) *Systemredovisning MMS.* figure 1, Lynx mobile mapper. Unpublished internal document.

[Johnsbråten, 2015] Johnsbråten, I. (2015).*Mobile Mapping System positioning in GNSS denied environments.* Master's thesis, Norwegian University of Life Sciences (NMBU).