Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2023

Visual-inertial SLAM using a monocular camera and detailed map data

Ludvig Berglund and Viktor Ekström

LINKÖPING
UNIVERSITY

Master of Science Thesis in Electrical Engineering

Visual-inertial SLAM using a monocular camera and detailed map data

Ludvig Berglund and Viktor Ekström

LiTH-ISY-EX--23/5549--SE

Supervisor: **Anton Kullberg**
ISY, Linköpings universitet
Jonas Nygårds
Swedish Defence Research Agency, FOI

Examiner: **Gustaf Hendeby**
ISY, Linköpings universitet

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Abstract

The most commonly used localisation methods, such as GPS, rely on external signals to generate an estimate of the location. There is a need of systems which are independent of external signals in order to increase the robustness of the localisation capabilities. In this thesis a visual-inertial SLAM-based localisation system which utilises detailed map, image, IMU, and odometry data, is presented and evaluated. The system utilises factor graphs through Georgia Tech Smoothing and Mapping (GTSAM) library, developed at the Georgia Institute of Technology. The thesis contributes with performance evaluations for different camera and landmark settings in a localisation system based on GTSAM. Within the visual SLAM field, the thesis also contributes with a sparse landmark selection and a low image frequency approach to the localisation problem. A variety of camera-related settings, such as image frequency and amount of visible landmarks per image, are used to evaluate the system. The findings show that the estimate improve with a higher image frequency, and does also improve if the image frequency was held constant along the tracks. Having more than one landmark per image result in a significantly better estimate. The estimate is not accurate when only using one distant landmark throughout the track, but it is significantly better if two complementary landmarks are identified briefly along the tracks. The estimate can also handle time periods where no landmarks can be identified while maintaining a good estimate.

Acknowledgments

Thank you Jonas Nygårds for contributing with both knowledge and great character which set the basis for a successful project, and FOI in general for the hospitality and effort put into this project. We would also like to thank Fredrik Hemström for assisting with the data collection and construction of the data collection system. The thesis relies heavily upon previous works made public by Frank Dellaert and the Georgia Institute of Technology in the GTSAM toolbox and we would like to thank you for the contribution made within this general field.

Linköping, March 2023
Ludvig Berglund and Viktor Ekström

Contents

Notation

Variable list

Notation	Definition
x	State
X	Set of states
f	Motion function
h	Measurement function
y	Measurement
Y	Set of measurements
u	Control input
t	Time
g	Gravitational constant
p	Position
v	Velocity
V	Velocity from wheel encoders
R	Rotation matrix
ϕ	Euler angles
$(\cdot)^n$	Navigation frame
$(\cdot)^b$	Body frame
a	Specific force
ω	Rotation rate
C_i^j	Rotation matrix from frame i to j
E	Rotation rate matrix
F, G, H	Jacobians
θ	Odometry yaw
D	Distance
$\phi, \gamma.\delta$	Angles
s	Scene view
(u, w)	Coordinates of the projection point in pixels
$f_{x,y}$	Focal length in pixel units

Variable list. cont

Notation	Definition
η	Process noise
e	Measurement noise
$\doteq$	Definition
$\omega^{\wedge}$	Skew-symmetric matrix
ψ_i	Factor node
Ψ	Set of factor nodes
ζ	Variable node
X	Set of variable nodes
Σ	Measurement covariance matrix
A	Factor graph measurements Jacobian
Q	Square root information matrix
Δ	State update vector
S_j, S_k	Separators for Bayes net and tree
$\tilde{F}_k$	Remaining variables
$\tilde{C}_k$	Clique
Π_k	Parent clique
α, β	Threshold values
$\mathcal{T}$	Bayes tree
$c_{x,y}$	Principle point (usually at the image center)
$[R\|T]$	Joint rotation-translation matrix
M	Car pose
C	Camera pose
L	Landmark

Abbreviations

Abbreviation	Definition
EKF	Extended Kalman filter.
GNSS	Global navigation satellite system.
GTSAM	Georgia Tech Smoothing and Mapping. A toolbox with a complete SfM model structure implemented.
IMU	Inertial measurement unit. It measures acceleration and angular velocity.
LBA	Local bundle adjustment estimates the geometry of image sequences taken by a camera.
RTK	Real-time kinematics.
SfM	Structure from motion.
SLAM	Simultaneous localisation and mapping.

Definitions

Expression	Definition
Epipolar geometry	A data set which describes a geometry from two viewpoints.
Odometry	A motion sensor that can give the total distance travelled.
Frame	Equivalent to camera image and both will be used interchangeably throughout the thesis.
Pose	An objects position and rotation described in the global coordinate space.
FOI	Swedish Defence Research Agency.

1 Introduction

There are multiple localisation and position estimation methods that can be used to track a vehicle's movement and location, such as global navigation satellite system (GNSS) or the further developed GNSS based method real-time kinematics (RTK). In this thesis, a different method is evaluated, which utilises a simultaneous localisation and mapping (SLAM) algorithm using an inertial measurement unit (IMU), odometry, image data, and detailed map data of the area. By collecting the IMU and odometry data for a vehicle whilst driving, it is possible to recreate the trajectory of the vehicle. Due to uncertainty in sensors and data interference, only relying on an IMU and odometry model will cause integration drift, which is when small errors and uncertainties are integrated over multiple time steps, making them significantly impact the result [1].

In this thesis, images from a monocular pinhole camera are used to calibrate the model to prevent integration drift. A simulated camera was used since the localisation system process the measurements offline. The landmark global coordinates can be found from the detailed map data. The chosen landmarks were distinguishable trees, houses, or intersections with visible edges. The system then continuously updates the estimate of the landmarks' positions based on the measurements. The camera projection error, which is the difference between observed projection in the image and estimated projection based on vehicle location and a camera model, is minimised with local bundle adjustments (LBA). The LBA method aims to reconstruct the 3D object with optimal parameter settings such as camera pose [2]. This can be used in the SLAM algorithm similar to structure from motion (SfM) problems [3]. When data from all sensors are collected, the visual-inertial SLAM problem can be solved. This is done with a method called iSAM2 (incremental smoothing and mapping) which uses inference in graphical models to estimate the trajectory. The main focus of this thesis is to evaluate the localisation system in terms of accuracy, how different landmarks affect the localisation capability, and what impact image frequency imposes on the estimated trajectory. The trajectory modelling, calibration, and localisation are performed post-data collection and not in real-time. This work is conducted in affiliation with the Swedish Defense Research Agency (FOI) who has requested an evaluation of a GPS-denied localisation method using iSAM2 and the open source C++ library Georgia Tech Smoothing and Mapping (GTSAM) developed at Georgia Tech.

1.1 Purpose

The purpose is to create a system that can be used to localise a vehicle, and evaluate certain key features of the developed system. The system will act as an alternative to already established localisation methods such as GNSS, with high localisation robustness as no external signals are needed. The system uses a camera to identify landmarks with known global coordinates from detailed map data and interpret IMU and odometry data to generate the vehicle position estimate. The beneficial aspect of this system compared to GNSS is that it can be performed without any active signal communication outside of the vehicle if the system has access to the map data on a local data storage unit.

1.2 Problem statement

With a camera based localisation system, it is of interest to evaluate how different frequencies at which images are captured affect the localisation capability. Each image captured with the camera that contains a landmark will pose a constraint on the vehicle position estimate. It is therefore of interest to evaluate different landmark settings and see how the relative distance and translation between vehicle and landmark, amount of landmarks visible in each image, and how the amount of landmarks visible throughout the vehicle path affects the localisation estimate. The problem of data association is assumed to be solved in this thesis, meaning that it is known which landmarks are seen in each image. The projected landmarks are assumed to be visible in all images regardless of possible vegetation that would exist in a real-world setting. Thus, this thesis aims to evaluate the following aspects.

- How accurate is the localisation estimate of the system compared to the ground truth track?
- How do different image frequencies affect the localisation estimate of the system?
- How does the selection of landmarks in the images affect the localisation estimate of the system?

1.3 Division of labor

Ludvig was responsible for collecting map data. He also derived the odometry model and the odometry Jacobian and implemented this into the localisation model. Ludvig also had the main responsibility for evaluating the frequency and amount of landmark setting impact on the estimate. Viktor was responsible for the theoretical explanations to the probabilistic inference in the graphical models. Viktor also retrieved the data regarding the poses used as ground truth based on RTK, INS, and odometry and evaluated the sparse landmark selection.

1.4 Outline

The thesis introduction clarifies which aspects that will be studied, and the problem statements that will be studied in the thesis. In Chapter 2 related work to the visual SLAM field is presented to put the thesis in a wider context. All relevant theory for the used localisation method is presented in Chapter 3. It contains information about motion and measurement models, and probabilistic inference using graphical models. The method used to gather data and answer the problem statement is presented in Chapter 4, where the data collection, possible biases, and data processing are discussed. The results can be found in Chapter 5 where relevant data is presented and is further expanded upon in Chapter 6, where the problem statement is elaborated with regards to the result. The method is also a subject in the discussion chapter, where the method decisions are evaluated and the result is discussed. In Chapter 7, conclusions based on Chapter 6 and possible future work is presented for the evaluated method. The appendix consists of two parts. The first is related to the theory chapter and the second to the results.

2

Related work

There have been many interesting contributions to monocular vision-based localisation algorithms over the last decade, e.g., [4], [5] and [6]. The relevant contributions within SLAM for this thesis can mainly be divided into visual SLAM (vSLAM), which solely uses camera inputs and visual-inertial SLAM (viSLAM) which is complemented with an IMU. Visual-inertial SLAM has gained increased popularity because of the possibility to integrate IMU measurements between images.

For vSLAM the methods can be divided into filter-based and keyframe-based approaches [7]. The filter-based methods are derived from the original solutions to the SLAM problem. These methods include EKF-based algorithms, for instance, MonoSLAM [8]. It also includes FastSLAM and its monocular SLAM correspondence which are derived from particle filters [9]. Another example of an algorithm based on the EKF filter is the multistate constraint Kalman filter (MSCKF) which contains a measurement model that can utilise geometric constraints from observing static features consecutively [10]. In general, these filter-based methods estimate the landmarks' positions as well as the camera's pose, using states expressed for both the landmarks and the camera. This could lead to problems regarding computational scalability [7].

The keyframe-based approaches, on the other hand, can also be referred to as optimisation-based approaches. These methods often descend from parallel tracking and mapping (PTAM) [11]. The difference from the filter-based methods is that, instead of using a filter that keeps track of both the poses and map, they utilise global optimisation. In a viSLAM setting, the process is often performed by LBA which can be explained as minimising the re-projection error between the predicted and observed image point of a landmark. Global optimisation can correct the drift effects, hence giving high accuracy. The higher accuracy also comes with a decrease in computation speed. Therefore, before PTAM many of these methods were mainly implemented offline since the required computational power was too high [7].

An interesting approach, within the subject of GNSS-denied vehicle localisation using visual-inertial SLAM, was proposed by Chiu et al. [12]. They achieved sub-meter navigation accuracy for a vehicle in a large-scale urban environment, by using a navigation system that couples the data from an IMU with observations of pre-mapped visual landmarks.

Furthermore, a localisation method that integrates the constraints of 3D traffic signs with LBA was proposed by Qu et al. in [13]. Their model was extended to automatically match traffic signs in images to those in a 3D landmark database. One can imagine the problem of outdoor landmarks having different appearances depending on the time of year. This was addressed by Beall and Dellaert [14] who integrated stereo imagery from different times of the year and used that as the basis for localisation. Many of the modern monocular visual SLAM algorithms are derived from incremental SfM methods. A novel monocular SLAM method that integrates recent advancements in SfM was made in [15]. They suggested a technique relating to graphical solutions that is more robust against errors in map initialization and adopted a global SfM method for the pose-graph optimisation.

A visual SLAM method using an omnidirectional camera was developed at Shanghai University, where an EKF was used to localise a vehicle in an unknown environment. One noted benefit of using an omnidirectional camera was the panoramic field of view which enabled each frame to capture more landmarks than a regular directional camera would be able to. In their solution, corner features were found in the images and later used for the EKF, which they concluded gave a good result [16]. Another contribution within the monocular visual SLAM field was the work of Andréz Díaz and Eduardo Caicedo who recreated the trajectory of a camera using corner detection in an EKF and extracted the six degrees of freedom pose of the camera. Their solution did not require any IMU or GNSS measurements. In their implementation, as many as 40 landmarks were used in each image, and images were captured at 10 Hz [17].

These methods generally rely heavily on identifying an abundance of landmarks with unknown global coordinates in each image, and with a high frequency. This thesis instead aims to utilise detailed map data of an area, lower the frequency at which images needs to be captured, and few visible landmarks per image to generate the estimate. The map data will associate landmarks with an estimated location, which will be continuously updated by the SLAM algorithm. The path estimate will be supported by IMU measurements and an odometry model, which reduces the dependence on the camera measurements. To produce a path estimate, an algorithm called iSAM2, implemented in the package GTSAM, will be used [18],[19]. Some advantages of iSAM2 compared to implementing an EKF are higher quality for nonlinear models, and that smoothing can be implemented to perform updates in batches or incrementally [20]. It was however shown by [21] that for some real-world implementations that difference in performance between incremental smoothing and an EKF is small. The article also states that there are possible benefits of using a smoothing approach such as easy incorporation of delayed measurements.

3 Theory

To present a solution to the problem statement, a graph-based localisation system has been implemented. In the following sections, the theory behind each significant component of the localisation system is explained. The theory behind the used graphical models in the GTSAM C++ library will be presented, as well as how iSAM2 is used to solve the estimation problem.

3.1 Model overview

The purpose of the model is to estimate the trajectory of a vehicle. This is achieved by defining states for the landmarks, vehicle and camera, and then express motion- and measurement models which describes these states' transition between time steps. The motion and measurement models are then included in a factor graph where an incremental smoothing and mapping (iSAM2) algorithm is used to form probability distributions of the states. The model uses states x, a motion model f^{INS}, measurement models h, control inputs u, state observations y, process noise η and measurement noise e in the factor graph to find the maximum a posteriori (MAP) estimate of the states. The estimate is driven by an inertial navigation system (INS) based motion model f^{INS}, using IMU measurements. In addition to the motion model, there are three measurement models used in the estimation problem, h^{odo}, h^{cc}, and h^{proj}. Measurement model h^{odo} measures the vehicle's position and orientation using an odometry model, h^{proj} measures landmark projections in images, and h^{cc} measures the rotational difference between the vehicle and the vehicle-mounted camera.

3.2 States used in the estimation

To construct a GTSAM graphical model, a set of states and state correlation needs to be set in each new time step. In GTSAM graphical models previous states are stored in the graph making it possible to access states in time step $k - n$, if $n \leq k$. The state vector x consists of vehicle position p, vehicle velocity v, rotation matrices R^{car} and R^{cam} relative the global coordinate system and the landmarks' positions L as

$$x = \begin{bmatrix} p^T & v^T & R^{carT} & R^{camT} & L^T \end{bmatrix}^T, \tag{3.1}$$

where the rotations R^{car} and R^{cam} are notated as vectors with all the matrix components. All states are defined in the global coordinate system. Each state is a part of a variable in the factor graph representation where $M = [p^T, v^T, R^{carT}]^T$, $C = R^{cam}$ and L is the landmark states. The motion model f^{INS} sets state progression in p, v and R^{car} between time step k and $k+1$ according to

$$\begin{bmatrix} p_{k+1} \\ v_{k+1} \\ R^{car}_{k+1} \end{bmatrix} = f^{INS}(p_k, v_k, R^{car}_k, u^{IMU}_k, \eta_k), \tag{3.2}$$

where the input u^{IMU}_k consists of acceleration a_k and angular velocity ω_k measurements from the IMU. The function f^{INS} can be found in Section 3.7. State correlations in the graph are also set using three different measurement models that were implemented according to

$$y_k = \begin{bmatrix} y^{odo}_k \\ y^{cc}_k \\ y^{proj}_k \end{bmatrix} = \begin{bmatrix} h^{odo}(x_k, x_{k-1}, e^{odo}_k) \\ h^{cc}(x_k, e^{cc}_k) \\ h^{proj}(x_k) + e^{proj}_k \end{bmatrix}. \tag{3.3}$$

The noise is not modelled as additive in h^{odo} and h^{cc}, as the measurement noise is included in the measurement model functions where it develops non-linearly. The measurement y^{odo}_k correspond to an odometry motion model as described in Section 3.8, and y^{cc} is the relation between the orientation of the vehicle and camera according to

$$y^{cc}_k = u^{cam}_k. \tag{3.4}$$

The input signal u^{cam} is the relative pose between the vehicle and the camera. Both the process noise η_k and measurement noise e_k are approximated as Gaussian noise for simplicity reasons and will be further described for the INS and the measurement models. The y^{proj}_k measurement is a pixel coordinate of a projected landmark in an image. The measurement functions h are defined as

$$h^{odo}(p_k, p_{k-1} R^{car}_k, R^{car}_{k-1}, e^{odo}_k) = \begin{bmatrix} p_k - p_{k-1} \\ R^{carT}_{k-1} \; R^{car}_k \end{bmatrix} \tag{3.5a}$$

$$h^{cc}(R_k^{car}, R_k^{cam}, e_k^{cc}) = \begin{bmatrix} R_k^{camT} \; R_k^{car} \end{bmatrix} \tag{3.5b}$$

$$h^{proj}(r_k, \phi_k^{cam}, L_k) = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{R}_{11} & \mathcal{R}_{12} & \mathcal{R}_{13} & T_x \\ \mathcal{R}_{21} & \mathcal{R}_{22} & \mathcal{R}_{23} & T_y \\ \mathcal{R}_{31} & \mathcal{R}_{32} & \mathcal{R}_{33} & T_z \end{bmatrix} \begin{bmatrix} L^x \\ L^y \\ L^z \\ 1 \end{bmatrix}, \tag{3.5c}$$

where all parameters in $h^{proj}(r_k, \phi_k^{cam}, L_k)$ are further explained and expanded upon in Section 3.9.

3.3 An introduction to factor graphs in GTSAM

GTSAM is a C++ library created by a team at the American university Georgia Tech [22]. The library implements sensor fusion for robotics and computer vision applications. The areas of implementation are mainly SLAM, visual odometry, and SfM. GTSAM is also compatible with both python and MATLAB, which will be used in this project. The toolbox is built around factor graphs which are graphical models that can be used when modelling estimation problems.

A factor graph is a bipartite graph, i.e., a graph where the nodes can be divided into two independent and disjoint sets called factors and variables. It can be described as a graphical model of the probability distribution $P(X|Y)$ where X are the states and Y the measurements. The factor graph structure implemented in this thesis can be seen in Figure 3.1. The variables contain states that are part of the motion and measurement models and are visualised in the figure as vehicle pose and velocity M, camera rotations C, and the landmarks L to intuitively represent how the hardware and map components correlate. The connecting dots represent the factors that contain probabilistic information about the variables. The factors that connect the camera poses to the landmarks are called projection factors in this thesis and correspond to local bundle adjustment, which is when a landmark projection pixel coordinate from the camera model is compared to the observation. There are a priori factors connected to the variable M_1 and all landmarks L in the factor graph, which corresponds to an initialisation of the states. A brief description of the factors used in the implemented factor graph can be seen in Table 3.1.

Table 3.1: *Description of the different factors in Figure 3.1.*

Factor	Description
ψ_{car}	Prior on the first vehicle pose
$\psi_{landmark}$	Prior on a landmark's position
ψ_{INS}	Factor between two car poses using the INS
ψ_{odo}	Factor between two car poses using odometry
ψ_{cc}	Factor that constrains the car and camera poses
ψ_{proj}	Factor that uses the projection of the landmark

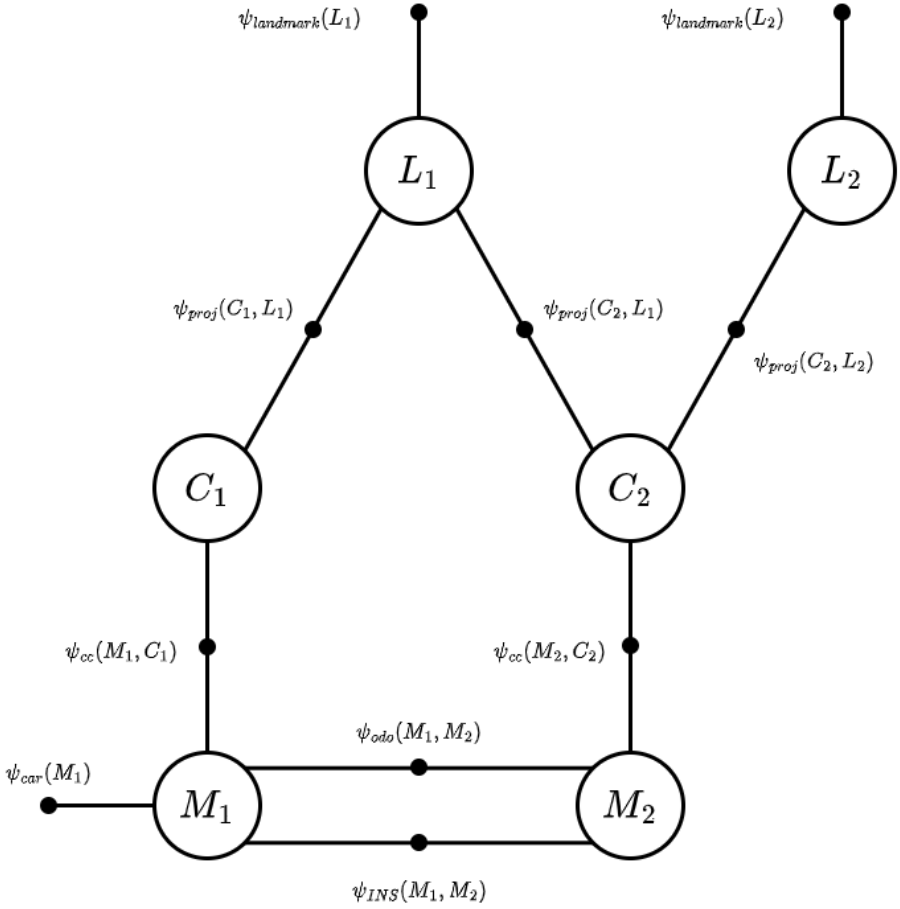

Figure 3.1: *A factor graph consisting of three variable node types; vehicle pose and velocity, M_k, camera pose, C_k at time step $k = 1, 2$. As well as two landmarks, L_1 and L_2.*

3.4 Bayesian inference

The states are found through statistical estimation, where observations are the basis for acquiring knowledge about said states. Updating the inferred state information as new measurements are obtained is called recursive estimation. Within the framework of Bayesian estimation, both observations and the states used in the estimation are stochastic variables. In this section, the general solution will be presented in terms of Bayesian recursions, and differences between general filtering and smoothing methods will be defined within a Bayesian framework. The information in this section is mainly obtained from [23].

3.4.1 Optimal filtering and smoothing

Both optimal filtering and smoothing methods aim to estimate unknown states $X = \{x_0, \dots, x_n\}$, given noisy measurements, $Y = \{y_1, \dots, y_n\}$. The measurements yield information about the states through indirect observations. From a Bayesian point of view, the problem is to find the posterior distribution $P(X|Y)$, through means of probabilistic inference. According to [24], two assumptions are needed to find the desired information from the probability distribution. The first is that the prior distribution of the state vector is known, i.e., $P(X)$ is known. The second assumption is that there exists a likelihood $P(Y|X)$ which couples the observations to the states. From these assumptions, it is possible to define the posterior distribution through Bayes' law as

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)}. \tag{3.6}$$

The prior $P(X)$ is assumed to be a Markov sequence, i.e., future states only depend on the present state. Thus, the prior can be described by $P(x_0)$ and $P(x_{k+1}|x_k)$. The term $P(Y|X)$ corresponds to the measurement's likelihood function and describes the dependence between a measurement y_k and state x_k. The term $P(Y)$ is a normalising constant and can through marginalisation be written as

$$P(Y) = \int P(Y|X)P(X)dX. \tag{3.7}$$

Note that the formulation in (3.6) corresponds to the full posterior distribution, i.e., it contains all states and all measurements. Solving this would yield the optimal solution but would also have a negative impact on the required computational power. This is because an added measurement would require the entire distribution to be recalculated. Many different approaches exist to solve this problem, some through recursions with marginal distributions. The marginal distributions used for Bayesian filtering methods consist of the current state x_k, given the current and previous measurements $y_{1:k}$, i.e. $P(x_k|y_{1:k})$. This differs from smoothing methods which keep previous states, making it possible to re-estimate them when new measurements are obtained. This method requires more computational power and is described in Section 3.6.2.

Probability densities with many dimensions can often be factorised to contain several densities where each covers a smaller dimension. The product of these yields the entire probability density. Assuming a Markov sequence

$$P(X) = P(x_0)\prod_{k=0}^{n} P(x_{k+1}|x_k), \tag{3.8}$$

$$P(Y|X) = \prod_{k=1}^{m} P(y_k|x_k), \tag{3.9}$$

where the distributions are assumed to be multivariate Gaussian distributions and are, in this thesis, modelled with a factor graph. The distribution for the state transitions $P(x_{k+1}|x_k)$ is obtained from the INS motion model f^{INS} as

$$P(x_{k+1}|x_k) = \frac{1}{\sqrt{|2\pi\Sigma^{INS}|}} \exp\Bigg(- \frac{1}{2}\Big\|f^{INS}(x_k, u_k) - x_{k+1}\Big\|^2_{\Sigma^{INS}}\Bigg), \tag{3.10}$$

where the variable Σ_{INS} denotes the covariance for the INS. The mathematical notation $\|e\|^2_\Sigma$ is defined as

$$\Big\|e\Big\|^2_\Sigma \doteq e^T\Sigma^{-1}e = \Big\|\Sigma^{-1/2}e\Big\|^2_2, \tag{3.11}$$

where $\|\cdot\|^2_2$ is the squared Euclidean norm. The distribution for the measurement models is formulated with

$$P\Big(y^i_k|x_x\Big) = \frac{1}{\sqrt{|2\pi\Sigma^i|}} \exp\Bigg(- \frac{1}{2}\Big\|h^i(x_k) - y^i\Big\|^2_{\Sigma^i}\Bigg), \tag{3.12}$$

where y^i and h^i correspond to the measurements and their respective measurement model.

3.4.2 Maximum a posteriori inference

There are different methods to obtain point estimates the state X. A common method, which is used in this thesis, is the maximum a posteriori estimate. It finds the states that maximise the posterior density $P(X|Y)$ as

$$X^{MAP} = \arg\max_X P(X|Y) = \arg\max_X \frac{P(Y|X)P(X)}{P(Y)}. \tag{3.13}$$

3.5 Inference in factor graphs

Factor graphs, as described in Section 3.3, contain factor nodes $\psi_i \in \Psi$ corresponding to control inputs and measurements, and variable nodes $\{M_i, C_i, L_i\} \in X_i$ that are assigned to a factor ψ_i. The states presented in Section 3.2 can be seen as sub-states to the different variable nodes, hence the variable notations in the factor graph will be used from this point. MAP inference in factor graphs simply maximizes the posterior probability

$$X^{MAP} = \arg\max_X \prod_i \psi_i(X_i). \tag{3.14}$$

The different factors are defined as Gaussian distributions in the factorisations presented in (3.10) and (3.12), and are expressed as

$$\psi_{INS}(M_{k+1}, M_k) = P(x_{k+1}|x_k), \tag{3.15a}$$

$$\psi_{odo}(M_k, M_{k-1}) = P(y^{odo}_k|x_k), \tag{3.15b}$$

$$\psi_{cc}(M_k, C_k) = P(y_k^{cc}|x_k), \tag{3.15c}$$

$$\psi_{proj}(C_k, L_k) = P(y_k^{proj}|x_k). \tag{3.15d}$$

The MAP estimate can then be found by taking the negative log of (3.14), using the factor definition in (3.15d), which gives a nonlinear least squares problem

$$\begin{aligned} X^{MAP} = \arg\min_X \Bigg\{ & \sum_{i=0}^{I} \left\| f^{INS}(M_i, u_i^{IMU}) - M_{i+1} \right\|^2_{\Sigma_i^{INS}} + \\ & \sum_{j=1}^{J} \left\| h^{odo}(M_{i_j}, M_{i_{j-1}}) - y_j^{odo} \right\|^2_{\Sigma_j^{odo}} + \\ & \sum_{k=1}^{K} \left\| h^{cc}(M_{i_k}, C_{i_k}) - y_k^{cc} \right\|^2_{\Sigma_k^{cc}} + \\ & \sum_{n=1}^{N} \left\| h^{proj}(C_{i_n}, L_{i_n}) - y_n^{proj} \right\|^2_{\Sigma_n^{proj}} + \\ & \sum_{m=1}^{O} \left\| L_{i_m} - \hat{L}_m \right\|^2_{\Sigma_m^L} \Bigg\}, \end{aligned} \tag{3.16}$$

where the last summation corresponds to priors for the landmarks with an initial position estimate from the map data $\hat{L}_m$. From here (3.16) is linearised. Therefore it is assumed that the system has an adequate linearisation point. By first defining the state update vector $\Delta \triangleq X_i - X_i^0$ with a linearisation point X_i^0 we can obtain the following linear least squares problem

$$\begin{aligned} \Delta^* = \arg\min_\Delta \Bigg\{ & \sum_{i=0}^{I} \left\| F_i^{INS,i-1} \tilde{M}_i + G_i^{INS,i} \tilde{M}_{i+1} - a_i \right\|^2_{\Sigma_i^{INS}} + \\ & \sum_{j=1}^{J} \left\| H_j^{odo,i_j} \tilde{M}_{i_j} - b_j \right\|^2_{\Sigma_j^{odo}} + \\ & \sum_{k=1}^{K} \left\| H_k^{cc,i_k} [\tilde{M}_{i_k}, \tilde{C}_{i_k}]^T - c_k \right\|^2_{\Sigma_k^{cc}} + \\ & \sum_{n=1}^{N} \left\| H_n^{proj,i_n} [\tilde{C}_{i_n}, \tilde{L}_{i_n}] - d_n \right\|^2_{\Sigma_n^{proj}} + \\ & \sum_{m=1}^{O} \left\| \tilde{L}_{i_m} \right\|^2_{\Sigma_m^L} \Bigg\}, \end{aligned} \tag{3.17a}$$

$$a_i \doteq M_{i+1}^0 - f^{INS}(M_i^0, u_i), \tag{3.17b}$$

$$b_j \doteq y_j^{odo} - h^{odo}(M_{i_j}^0, M_{i_{j-1}}^0), \tag{3.17c}$$

$$c_k \doteq y_k^{cc} - h^{cc}(M_{i_k}^0, C_{i_k}^0), \tag{3.17d}$$

$$d_n \doteq y_n^{proj} - h^{proj}(C_{i_n}^0, L_{i_n}^0), \tag{3.17e}$$

where the notation $(\tilde{\cdot})$ is the deviation from the expected value for the respective state. F^{INS} is the Jacobian for the INS while G^{INS} corresponds to a unit matrix in the case of this thesis. Each matrix H is the observation Jacobian for its respective measurement model. The component a corresponds to the prediction error of the motion model while b, c, and d are observation residuals. The Jacobians and residuals can be combined into one matrix A, called the measurement Jacobian, and one vector q respectively. This is done by dropping the covariances by using definition (3.11) as

$$A_i = \Sigma_i^{-1/2} H_i, \tag{3.18a}$$

$$q_i = \Sigma_i^{-1/2}(z_i - g_i(X_i^0)), \tag{3.18b}$$

for a measurement with a Jacobian H_i and covariance Σ_i. The term $(z_i - g_i(X_i^0))$ corresponds to the different residuals in (3.17). Then (3.17) can be written as

$$\Delta^* = \arg\min_{\Delta} \|A\Delta - q\|_2^2. \tag{3.19}$$

where the measurement Jacobian A is a large, sparse matrix and has a structure that is equivalent to the structure of the factor graph. This can be seen by first separating the Jacobians for the factor between the car and camera which gives

$$H^{ccM} = \frac{\partial h^{cc}}{\partial M}, \tag{3.20a}$$

$$H^{ccC} = \frac{\partial h^{cc}}{\partial C}, \tag{3.20b}$$

where the Jacobian is split into one part that contains the dependencies for the car and one for the camera. The same can be done for the projection factor with

$$H^{projC} = \frac{\partial h^{proj}}{\partial C}, \tag{3.21a}$$

$$H^{projL} = \frac{\partial h^{proj}}{\partial L}, \tag{3.21b}$$

where the Jacobian is split with one part for the camera and one for the landmark. To show an example of the measurement Jacobian A we can use the the factor graph presented in Figure 3.1 and for simplicity assume the covariances have already been multiplied to each Jacobian. In this case A becomes

$$A = \begin{array}{cc} \begin{array}{cccccc} M_1 & M_2 & C_1 & C_2 & L_1 & L_2 \end{array} & \\ \begin{bmatrix} G_1^{INS,1} & & & & & \\ F_2^{INS,1} & G_2^{INS,2} & & & & \\ H_2^{odo,1} & H_2^{odo,2} & & & & \\ H_1^{ccM,1} & & H_1^{ccC,1} & & & \\ & H_2^{ccM,2} & & H_2^{ccC,2} & & \\ & & H_1^{projC,1} & & H_1^{projL,1} & \\ & & & H_2^{projC,2} & H_2^{projL,1} & \\ & & & H_2^{projC,2} & & H_2^{projL,2} \\ & & & & I_3 & \\ & & & & & I_3 \end{bmatrix} & \begin{array}{c} \psi_{car} \\ \psi_{INS} \\ \psi_{odo} \\ \psi_{cc} \\ \psi_{cc} \\ \psi_{proj} \\ \psi_{proj} \\ \psi_{proj} \\ \psi_{landmark} \\ \psi_{landmark} \end{array} \end{array},$$

where each column correspond to a variable and each row to a factor in the factor graph defined by the labels above and to the right of the matrix. As the matrix and the graph have the same structure it possible to use a method called variable elimination. In terms of linear algebra variable elimination can be explained as factorising the measurement Jacobian into an upper-triangular matrix called the square root information matrix Q which fulfills the expression

$$A^T A = Q^T Q \tag{3.22}$$

In graphical terms, variable elimination means to factorise the factor graph into a Bayes net, which is a graphical representation of the probability distribution $P(X)$. This differs from the factor graph which contains the conditional probability $P(X|Y)$. What this means for the graphical structure can, according to [25], be described with the steps:

1. Choose a variable, here called ζ_j, for elimination.
2. Remove all factors $\psi_i(X_i)$, that are adjacent to ζ_j, from the factor graph.
3. Define the separator S_j as all variables involved in those factors, excluding ζ_j. A separator is a set of variables that separates two or more disjoint subsets of variables in a factor graph, such that the subsets are conditionally independent given the separator.
4. Create the product factor, which contains the parts of the matrices A and q

that are involved in the elimination of said variable, as

$$\begin{aligned}\Gamma(\zeta_j, S_j) &= \prod_i \psi_i(X_i),\\ &= \exp\Big\{-\frac{1}{2}\sum_i \|A_i X_i - q_i\|_2^2\Big\},\\ &= \exp\Big\{-\frac{1}{2}\|\bar{A}_j \begin{bmatrix}\zeta_j & S_j\end{bmatrix}^T - \bar{q}_j\|_2^2\Big\}.\end{aligned}$$

For instance, if the variable L_1 is to be eliminated, the matrix $\bar{A}_j$ would contain the Jacobians of the three involved factors as

$$\bar{A}_j = \begin{bmatrix} I_3 & & \\ H_1^{projL,1} & H_1^{projC,1} & \\ H_2^{projL,1} & & H_2^{projC,2}\end{bmatrix}$$

5. The next step is to factorise the product $\Gamma(\zeta_j, S_j)$ which, in this thesis is done with QR factorisation. GTSAM does however also supports other factorisation alternatives, such as Cholesky factorisation. QR factorisation can be described as expressing a matrix as the product between an orthogonal matrix and an upper triangular matrix. An elaborate description of QR factorisation can be found in [25]. In GTSAM it is done by first using the augmented matrix $[\bar{A}_j|\bar{q}_j]$ based on the product factor in step 4 as

   $$[\bar{A}_j|\bar{q}_j] = Q\begin{bmatrix}\tilde{R}_j & \tilde{T}_j & \tilde{q}_j\\ & \tilde{A}_\tau & \tilde{q}_\tau\end{bmatrix}$$

   where K is an orthogonal rotation matrix. $\tilde{Q}_j$, $\tilde{T}_j$ and $\tilde{q}_j$ are contributions to the square root information matrix Q with its equivalent graphical model, the Bayes net. $\tilde{A}_\tau$ and $\tilde{q}_\tau$ gives information about new factors created based on the eliminated variable's separator. From here $\Gamma(\zeta_j, S_j)$ can be factorised as

   $$\begin{aligned}\Gamma(\zeta_j, S_j) &= \exp\Big\{-\frac{1}{2}\|\bar{A}_j \begin{bmatrix}\zeta_j & S_j\end{bmatrix}^T - \bar{q}_j\|_2^2\Big\}\\ &= \exp\Big\{-\frac{1}{2}\|\tilde{R}_j\zeta_j + \tilde{T}_j S_j - \tilde{q}_j\|_2^2\Big\}\exp\Big\{-\frac{1}{2}\|\tilde{A}_\tau S_j - \tilde{q}_\tau\|_2^2\Big\}\\ &= P(\zeta_j|S_j)\tau(S_j)\end{aligned}$$

   where $P(\zeta_j|S_j)$ is a conditional probability added to the Bayes net while $\tau(S_j)$ is a new factor that is added to the factor graph and will be removed when one of the variables in the separator S_j is eliminated.

6. Repeat the steps with the next variable subject to elimination.

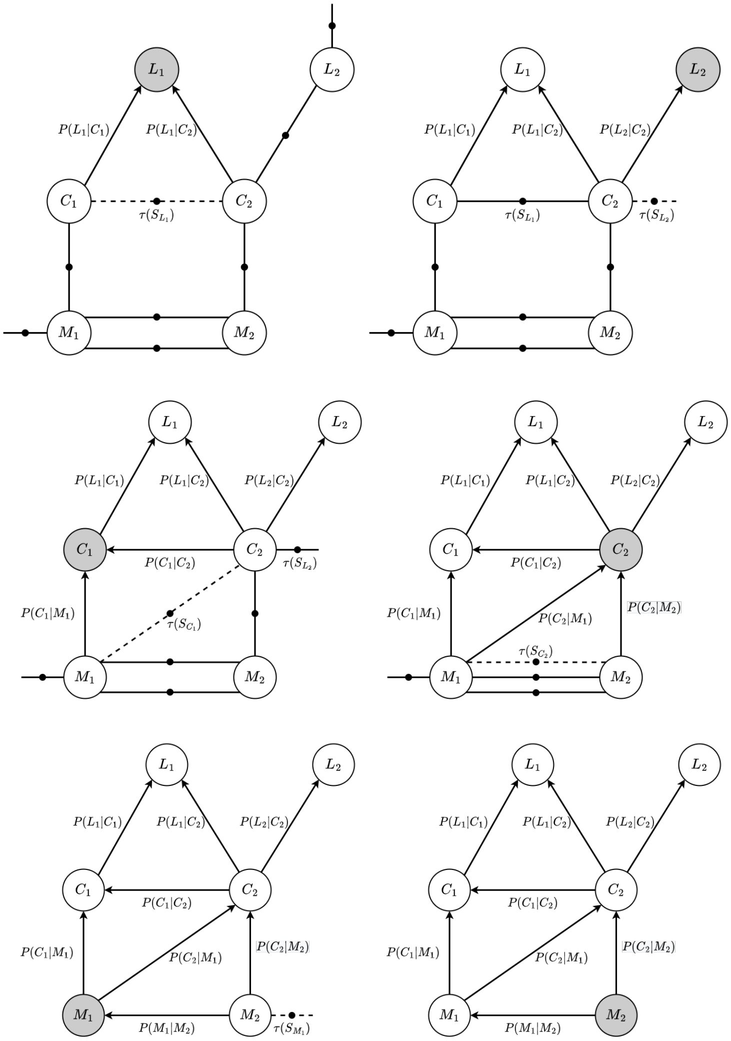

Figure 3.2: *The process of variable elimination with the ordering $(L_1, L_2, C_1, C_2, M_1, M_2)$ based on the factor graph in Figure 3.1. The steps consist of a combination between factor graphs and Bayes nets in order to show the procedure. The grey variable in each graph is the variable subject to elimination and dotted lines show the factors created according to step 5 as previously described. The final product is a Bayes net made up of conditionals.*

After these steps, both a Bayes net and its matrix equivalence Q, has been obtained. The structure of the Bayes net can differ depending on the order of the elimination since the new factor τ will differ depending on the separators. The effects of this were however not analysed in this thesis, instead, the built-in elimination process in GTSAM was used.

Further, the matrix Q can be used to obtain all covariances according to $\Sigma \triangleq (Q^T Q)^{-1}$. This operation has a high computational cost since Q becomes a large matrix over time, hence marginal covariances are computed according to [26]. The process and the final Bayes net obtained from variable elimination, with the elimination order $(L_1, L_2, C_1, C_2, M_1, M_2)$, on the factor graph in Figure 3.1 can be seen in Figure 3.2. Figure 3.2 shows the process of building the Bayes net of conditionals according to the described steps. It results in a graph where all factors have been removed and the entire net is described through conditional probabilities. In short, the elimination process factorised the factors $\psi(X)$ to a probability distribution $P(X)$, which for the Bayes net in Figure 3.2 looks like

$$\begin{aligned} P(X) =& P(L_1|C_1, C_2)P(L_2|C_2) \\ & P(C_1|M_1, C_2)P(C_2|M_1, M_2) \\ & P(M_1|M_2)P(M_2), \end{aligned} \tag{3.23}$$

where the prior on M_2 comes from the fact that the separator is empty when eliminating the last variable, in this case, M_2, hence resulting in a prior. This will be true regardless of what variable is eliminated last. From this point, back-substitution can be performed in the reverse order of the elimination to obtain the MAP estimate, for a detailed explanation see [25].

3.6 Incremental smoothing and mapping

In order to obtain an estimate of the trajectory after each step given the available sensor data, an incremental inference algorithm from the optimization library incremental smoothing and mapping (iSAM) was used. The incremental attribute of the algorithm means that it updates the estimate when new measurements are obtained. In this thesis, the algorithm used is called iSAM2 and is based on [18]. It uses a data structure in form of a Bayes tree in order to improve efficiency. This section describes the Bayes tree, how its updated, and the Bayes tree relinearisation.

3.6.1 Bayes tree

From the Bayes net obtained by variable elimination in Section 3.5, a Bayes tree will now be derived. This is possible since the net is chordal [18]. The nodes in the Bayes net correspond to cliques $\tilde{C}_k$ in the Bayes tree, thus the Bayes net contains the information from the obtained square root information matrix R. A clique can be described as a subset of nodes to which every node is connected. Further, a distribution $P(\tilde{F}_k|S_k)$ is established where S_k can be described as the variables that are contained in both a clique C_k and its parent Π_k, i.e. $\tilde{C}_k \cap \Pi_k$. $\tilde{F}_k$ contains the remaining variables defined as $\tilde{F}_k \doteq \tilde{C}_k \setminus S_k$. The notation $\setminus$ means that $\tilde{F}_k$ is defined as the parts of $\tilde{C}_k$ that are not contained within S_k. The clique notation is $\tilde{C}_k = \tilde{F}_k : S_k$ which shows what variables in a clique also is a part of its parent.

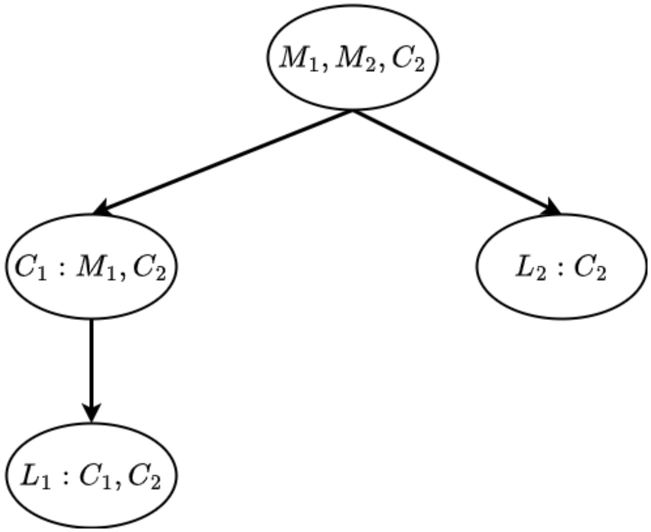

Figure 3.3: *Bayes tree, based on the Bayes net in Figure 3.2.*

As an example, the cliques for the final Bayes net in Figure 3.2 would be

$$\tilde{C}_r = \{M_1, M_2, C_2\}, \tag{3.24}$$

$$\tilde{C}_2 = \{C_1\} : \{M_1, C_2\}, \tag{3.25}$$

$$\tilde{C}_3 = \{L_1\} : \{C_1, C_2\}, \tag{3.26}$$

$$\tilde{C}_4 = \{L_2\} : \{C_2\}, \tag{3.27}$$

where the separator is empty for C_r since it is the root of the tree. These cliques result in the Bayes tree presented in Figure 3.3. Note that the Bayes tree will differ depending on the elimination order just as the Bayes net.

3.6.2 Bayes tree update and relinearisation

This part will cover a brief description of the problem of updating the Bayes tree and dealing with nonlinear factors by only performing relinearisation when needed. For a more elaborate theory, see [18] and [25]. An update is equivalent to receiving a new measurement which means the addition of a new factor to the factor graph. Since all information is stored in the Bayes tree we need to convert the parts of the tree that are affected back into the form of a factor graph. The parts that shall be converted are found from the variables in the factor about to be added. The cliques in the Bayes tree that contain these variables are removed as well as all the parent cliques up to the root and converted to a factor graph. The parts of the tree that are unaffected are stored. The new factor can then be added to the converted factor graph which goes through the process of variable elimination and creates a Bayes net. A new Bayes tree is then created from this Bayes net in combination with the previously stored parts. In Algorithm 1 these steps are described in a simplified update process, and for a more detailed version see [18].

Algorithm 1 Bayes tree update

```
In: Bayes tree $\mathcal{T}$, new factor $\psi_{new}(\zeta_i, \zeta_j)$
Out: Updated Bayes tree $\mathcal{T}'$
    1. Remove cliques containing the variable $\zeta_i$ or $\zeta_j$ and all the parent cliques up to the root.
    2. Store the unaffected parts of the tree as $\mathcal{T}_{store}$.
    3. Create a factor graph from the removed cliques and add the factor $\psi_{new}(\zeta_i, \zeta_j)$.
    4. Retrieve a Bayes net by performing variable elimination.
    5. Create a Bayes tree, $\mathcal{T}'$ from the Bayes net.
    6. Insert the previously stored parts, $\mathcal{T}_{store}$, into $\mathcal{T}'$.
```

When the Bayes tree has been updated with additional measurements, the solution needs to be updated as well. This could be solved by performing back-substitution, beginning at the root and proceeding downwards through the entire tree. It is however not desirable to perform this operation for the entire tree since it over time becomes inefficient. Thus, a process called fluid relinearisation is implemented [18]. Relinearisation can be explained as changing the values of already computed variables when new measurement factors have been added. And fluid refers to only performing relinearisation where it is needed. The motivation behind this is that updates to the tree usually only have an impact on the local parts of the tree. If a variable is to be relinearised or not depends on if its deviation from the linearisation point exceeds a threshold value. If the deviation exceeds this value, all parts of the tree containing this variable need to be modified. In short, this is done by replacing the affected parts of the tree with the information obtained from relinearising the corresponding nonlinear factors. This means finding a new linearisation point for the affected variables and inserting the result into the Bayes tree. For further information, see [18].

3.7 Inertial navigation system

Inertial navigation is used to track position and orientation relative to an initial state. The INS receives measurements from an IMU containing a gyroscope and accelerometer, oriented in the vehicle's local coordinate system. The coordinate systems for the INS can be seen in Figure 3.4 where B corresponds to the local coordinate system, referred to as the body frame. Figure 3.4 also shows T_{WB} which is the transformation between the body frame and the navigation frame

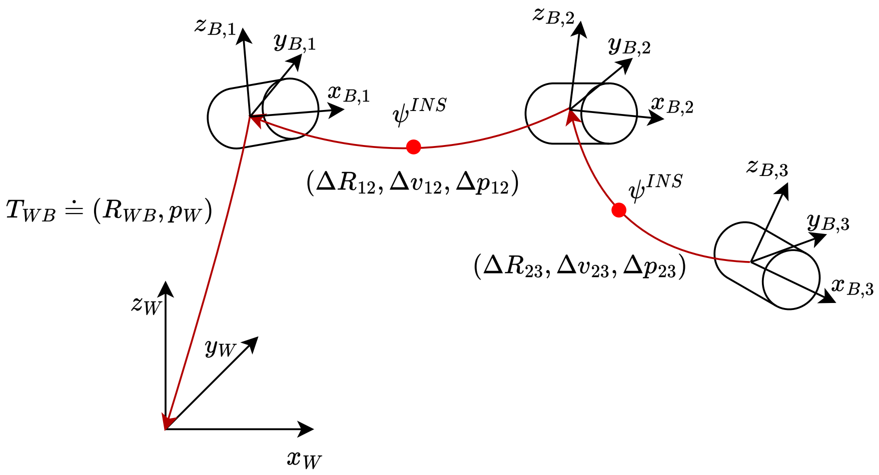
$z_{B,1}$
$y_{B,1}$
$x_{B,1}$
ψ^{INS}
$z_{B,2}$
$y_{B,2}$
$x_{B,2}$
$(\Delta R_{12}, \Delta v_{12}, \Delta p_{12})$
$T_{WB} \doteq (R_{WB}, p_W)$
ψ^{INS}
$z_{B,3}$
$y_{B,3}$
$x_{B,3}$
$(\Delta R_{23}, \Delta v_{23}, \Delta p_{23})$
z_W
y_W
x_W

Figure 3.4: *Coordinate systems and transformations for the INS. The three vehicle poses each correspond to the pose at a time step when an image is captured. A factor ψ^{INS} is created from preintegrated measurements between the time steps.*

W in terms of rotation R_{WB} and position p_W. The purpose of the INS is to find information about the pose of the vehicle (R, p) and its velocity v.

The INS used in the thesis comes from [27], where all possible rotations of a vehicle can be represented as tangents of a spherical manifold, that locally resembles Euclidean space. Exponential mapping is used to connect the tangent space to the manifold itself, according to Figure 3.5. Preintegration is performed on the manifold by using the exponential map, which means that the IMU measurements between time steps with images are contracted to create one INS factor using the rotational manifold expression, accessible by the exponential map conversion. The orientation will be defined with a $3x3$ rotation matrix. In the context of the INS the exponential map transforms small rotation vectors, expressed as $3x3$ skew-symmetric matrices, into rotation matrices. A skew-symmetric matrix is defined as

$$\omega^{\wedge} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{3.28}$$

The inverse of the exponential map is called the logarithmic map and transforms a rotation matrix into a skew-symmetric matrix. The exponential map is notated with $(\cdot)^{\wedge}$, while the logarithmic map is notated with $(\cdot)^{\vee}$. The full notations for the map transformations can then be expressed as

$$\begin{aligned} \text{Exp} &: \phi \rightarrow \exp(\phi^{\wedge}), \\ \text{Log} &: R \rightarrow \log(R)^{\vee}, \end{aligned} \tag{3.29}$$

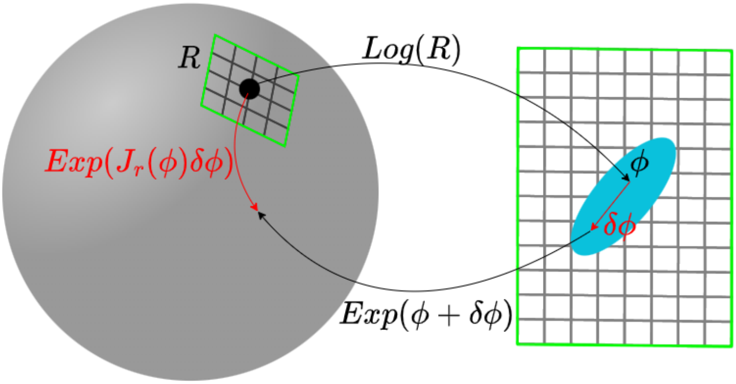

Figure 3.5: *Visualisation of the manifold to the left, the tangent space to the right and the mapping between them.*

Note that Exp denotes the exponential mapping, while exp denotes the exponential function. A visualisation of the manifold, the tangent space and its mapping can be seen in Figure 3.5. The term $\delta\phi$ is called perturbation which is a small incremental change to the skew-symmetric matrix. The variable J_r is the right Jacobian which is the derivative of the exponential map with respect to a small rotation. It is used to relate the additive perturbation in the tangent space to multiplicative perturbation on the manifold and is defined as

$$J_r(\phi) = I - \frac{1 - \cos(\|\phi\|)}{\|\phi\|^2}\phi^\wedge + \frac{\|\phi\| - \sin(\|\phi\|)}{\|\phi^3\|}(\phi^\wedge)^2. \tag{3.30}$$

This approach is used in this thesis to ensure that the INS estimates of the vehicle's motion are consistent with the constraints on the vehicle and improve the estimate's accuracy. Since the usage of manifolds and exponential mapping is not the focus of this thesis, see [28] for a more detailed description. There are however some important relationships that are needed, such as the first-order approximation of the exponential map

$$\exp(\phi^\wedge) \approx I + \phi^\wedge, \tag{3.31}$$

and the first-order approximation of the mapping of perturbations

$$\mathrm{Exp}(\phi + \delta\phi) \approx \mathrm{Exp}(\phi)\,\mathrm{Exp}(J_r(\phi)\delta\phi). \tag{3.32}$$

The IMU measures the rotation rate $\tilde{\omega}$ and acceleration $\tilde{a}$ of the vehicle in the body frame as

$$\bar{\omega}_B = \omega_B(t) + b^g(t) + \eta^g(t), \tag{3.33a}$$

$$\bar{a}_B(t) = R_{WB}^T(t)(a_W(t) - g_W) + b^a(t) + \eta^a(t), \tag{3.33b}$$

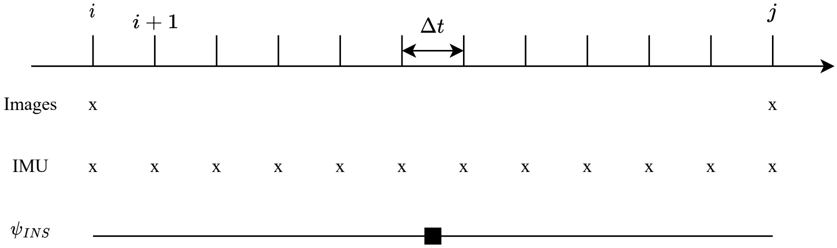

Figure 3.6: *Visualisation of the sampling rates for the IMU and camera where each x corresponds to a measurement.*

where b^g and b^a correspond to the gyroscope and accelerometer bias, respectively, while η^g and η^a are assumed as additive white noise. To find the desired information a kinematic model is defined as

$$\dot{R}_{WB} = R_{WB}\omega_B^\wedge, \tag{3.34a}$$

$$\dot{v}_W = a_W, \tag{3.34b}$$

$$\dot{p}_W = v_W, \tag{3.34c}$$

where $a_w = R_{WB}a_B$. Discretisation and assuming that the measurements a_W and ω_W are constant between t and $t + \Delta t$, while using the measurement equations (3.33) yields

$$R(t + \Delta t) = R(t)\,\mathrm{Exp}\Big((\tilde{\omega}(t) - b^g(t) - \eta^{gd}(t))\Delta t\Big), \tag{3.35a}$$

$$v(t + \Delta t) = v(t) + g\Delta t + R(t)\Big(\tilde{a}(t) - b^a(t) - \eta^{ad}(t)\Big)\Delta t, \tag{3.35b}$$

$$p(t + \Delta t) = p(t) + v(t)\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}R(t)\Big(\tilde{a}(t) - b^a(t) - \eta^{ad}(t)\Big)\Delta t^2, \tag{3.35c}$$

where the subscripts for the body and world frame has been dropped for simplicity. Note that the rotation matrix R refers to R^{car} for the INS model. It is also worth noting that the orientation $R(t)$ is assumed to be constant for the integration between two measurements. The superscript d in η^{gd} and η^{ad} corresponds to it being discrete-time noise.

Since the IMU has a high sample rate and the camera yields measurements at a low rate we only want to add the states, obtained from measurements, when an image is taken. This is shown in Figure 3.6 where the IMU measurements between two images taken at time step i and j, as previously stated, are contracted to one measurement which yields a single factor. This factor, called preintegrated IMU factor, sets constraints on the pose between the time steps of two images. If we assume that the measurements from the IMU are synchronised with the images from the camera we can summarise all IMU measurements by calculating the integration in Equation (3.35) for each Δt between the image taken at time step i and j. This gives

$$R_j = R_i \prod_{k=i}^{j-1} \mathrm{Exp}\Big((\tilde{\omega}_k - b_k^g - \eta_k^{gd})\Delta t\Big), \tag{3.36a}$$

$$v_j = v_i + g\Delta t_{ij} + \sum_{k=i}^{j-1} R_k(\tilde{a}_k - b_k^a - \eta_k^{ad})\Delta t, \tag{3.36b}$$

$$p_j = p_i + \sum_{k=1}^{j-1}(v_k\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}R_k(\tilde{a}_k - b_k^a - \eta_k^{ad})\Delta t^2), \tag{3.36c}$$

where $\Delta t_{ij} = t_j - t_i$. As stated in [27], the problem with (3.36) is that it would have to be computed each time the linearisation point at time t_i changes. In order to avoid this we formulate equations, that are independent of both the pose and the velocity at time t_i, as

$$\Delta R_{ij} \doteq R_i^T R_j = \prod_{k=i}^{j-1} \exp(\tilde{\omega}_k - b_k^g - \eta_k^{gd}), \tag{3.37a}$$

$$\Delta v_{ij} \doteq R_i^T(v_j - v_i - g\Delta t_{ij}) = \sum_{k=i}^{j-1} \Delta R_{ik}(\tilde{a}_k - b_k^a - \eta_k^{ad})\Delta t, \tag{3.37b}$$

$$\Delta p_{ij} \doteq R_i^T(p_j - p_i - v_i\Delta t - \frac{1}{2}g\Delta t_{ij}^2) = \sum_{k=i}^{j-1}(\Delta v_{ik}\Delta t + \frac{1}{2}\Delta R_{ik}(\tilde{a}_k - b_k^a - \eta_k^{ad})), \tag{3.37c}$$

where

$$\begin{aligned} \Delta R_{ik} &\doteq R_i^T R_k, \\ \Delta v_{ik} &\doteq R_i^T(v_k - v_i - g\Delta t_{ik}). \end{aligned} \tag{3.38}$$

(3.37) gives a relation between the images taken at two different time steps with the measurements from the IMU. In this thesis the bias is assumed to be constant between two images.

In order to reach the MAP estimate we need to define the densities of the measurements. We do this by manipulating (3.37), which means to isolate the noise terms. For the rotation increment ΔR_{ij} we use (3.32) and then a property of the exponential map which is

$$\mathrm{Exp}(\phi)R = R\,\mathrm{Exp}(R^T\phi), \tag{3.39}$$

and then we get

$$\begin{aligned}\Delta R_{ij} &\approx \prod_{k=i}^{j-1}\Big(\mathrm{Exp}\big((\tilde{\omega}_k - b_i^g)\Delta t\big)\mathrm{Exp}\big(-J_r^k \eta_k^{gd}\Delta t\big)\Big),\\ &= \Delta\tilde{R}_{ij}\prod_{k=i}^{j-1}\mathrm{Exp}\big(-\Delta\tilde{R}_{k+1j}^T J_r^k \eta_k^{gd}\Delta t\big),\\ &\doteq \Delta\tilde{R}_{ij}\mathrm{Exp}(-\delta\phi_{ij}),\end{aligned} \tag{3.40}$$

where the preintegrated rotation measurement $\Delta\tilde{R}_{ij} \doteq \prod_{k=i}^{j-1}\mathrm{Exp}((\tilde{\omega}_k - b_i^g)\Delta t)$ and its noise $\delta\phi_{ij}$ has been defined. Then we can substitute (3.40) to the expression for Δv_{ij} in (3.37) and use the first order approximation from (3.31) to replace $\mathrm{Exp}(-\delta\phi_{ij})$. From this we obtain

$$\begin{aligned}\Delta v_{ij} &\approx \sum_{k=i}^{j-1}\Delta\tilde{R}_{ik}(I - \delta\phi_{ik}^{\wedge})(\tilde{a}_k - b_i^a)\Delta t - \Delta\tilde{R}_{ik}\eta_k^{ad}\Delta t,\\ &= \Delta\tilde{v}_{ij} + \sum_{k=1}^{j-1}\Big(\Delta\tilde{R}_{ik}(\tilde{a}_k - b_i^a)^{\wedge}\delta\phi_{ik}\Delta t - \Delta\tilde{R}_{ik}\eta_k^{ad}\Delta t\Big),\\ &\doteq \Delta\tilde{v}_{ij} - \delta v_{ij},\end{aligned} \tag{3.41}$$

where the preintegrated velocity measurement $\Delta\tilde{v}_{ij} \doteq \sum_{k=i}^{j-1}\Delta\tilde{R}_{ik}(\tilde{a}_k - b_i^a)$ and its noise δv_{ij} is defined. We then substitute (3.40) and (3.41) in to (3.37) to express Δp_{ij} as

$$\begin{aligned}\Delta p_{ij} &\approx \sum_{k=1}^{j-1}\Big((\Delta\tilde{v}_{ik} - \delta v_{ik})\Delta t + \frac{1}{2}\Delta\tilde{R}_{ik}(I - \delta\phi_{ik}^{\wedge}(\tilde{a}_k - b_i^a)\Delta t^2 - \frac{1}{2}\Delta\tilde{R}_{ik}\eta_k^{ad}\Delta t^2\Big),\\ &= \Delta\tilde{p}_{ij} + \sum_{k=i}^{j-1}\Big(-\delta v_{ik}\Delta t + \frac{1}{2}\Delta\tilde{R}_{ik}(\tilde{a}_k - b_i^a)^{\wedge}\delta\phi_{ik}\Delta t^2 - \frac{1}{2}\Delta\tilde{R}_{ik}\eta_k^{ad}\Delta t^2\Big),\\ &\doteq \Delta\tilde{p}_{ij} - \delta p_{ij},\end{aligned} \tag{3.42}$$

where the preintegrated position measurement $\Delta\tilde{p}_{ij}$ and its noise δp_{ij} has been defined by again using the first-order approximation from (3.31). By finally substituting the three preintegrated measurements to the definitions of ΔR_{ij}, Δv_{ij} and Δp_{ij} in (3.37) we get

$$\Delta\tilde{R}_{ij} = R_i^T R_j\,\mathrm{Exp}(\delta\phi_{ij}), \tag{3.43a}$$

$$\Delta\tilde{v}_{ij} = R_i^T(v_j - v_i - g\Delta t_{ij}) + \delta v_{ij}, \tag{3.43b}$$

$$\Delta\tilde{p}_{ij} = R_i^T(p_j - p_i - v_i\Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2) + \delta p_{ij}, \tag{3.43c}$$

which is our preintegrated measurement model where the contracted measurements are defined as a function of the states to be estimated and added noise.

3.8 Odometry motion model

The odometry motion model uses measurements from wheel encoders mounted on the rear wheels of the vehicle. The odometry model generates a two dimensional pose, which then is transformed into a three dimensional pose to be used in the factor graph. The measurements contain information about the angular velocity ω^{wheel} of the individual wheels through pulse signals, from which a velocity V^{wheel} for each individual wheel can be calculated. In order to calculate the velocity, a conversion from the wheel encoder analog signal has to be done to obtain a digital signal. As the measurement noise gets converted accordingly, the noise is not Gaussian due to quantisation effects which can be summarized as errors in the conversion from analog to digital signals. However, for simplicity, the noise is still assumed Gaussian. The vehicle's absolute velocity is estimated to be the mean value of the rear wheel's velocities. The vehicle yaw rate can be calculated from the difference between the velocities obtained from the two rear wheels, divided by the length of the rear axis d^{axis}. The left rear wheel is denoted V_3^{wheel} and the right rear wheel is denoted V_4^{wheel} These two measurements corresponds to V and ω and are found from

$$V = \frac{V_3^{wheel} + V_4^{wheel}}{2}, \tag{3.44}$$

$$\omega = \frac{V_4^{wheel} - V_3^{wheel}}{d^{axis}}. \tag{3.45}$$

The odometry model is derived from Figure 3.7, where the vehicle orientation in time step k and $k-1$ is modelled in the local coordinate system $[X_0, Y_0, \theta_0]^T$, as well as its trajectory between these orientations. Three states are defined in each time step; $[x, y, \theta]^T$. Between time step k and $k-1$ it is assumed that the vehicle travels in an arc with constant velocity. The angle $\omega_{k-1}\Delta t$ determines the rotation center for the trajectory.

The resulting distance travelled is denoted D_{k-1}. The body frame in which the measured velocity was collected is denoted $[X_{k-1}, Y_{k-1}, \theta_{k-1}]^T$, where X_{k-1} is oriented in the vehicle travelling direction while Y_{k-1} is perpendicular to X_{k-1}. If the relationship between the global navigation frame and $[X_0, Y_0, \theta_0]^T$ is known, then $[x, y, \theta]^T$ can be expressed in the global navigation frame, resulting in the odometry model as

$$p_k^x = p_{k-1}^x + \frac{2V_{k-1}}{\omega_{k-1}} \sin\left(\frac{\omega_{k-1}\Delta t}{2}\right) \cos\left(\frac{\omega_{k-1}\Delta t}{2} + \theta_{k-1}\right), \tag{3.46a}$$

$$p_k^y = p_{k-1}^y + \frac{2V_{k-1}}{\omega_{k-1}} \sin\left(\frac{\omega_{k-1}\Delta t}{2}\right) \sin\left(\frac{\omega_{k-1}\Delta t}{2} + \theta_{k-1}\right), \tag{3.46b}$$

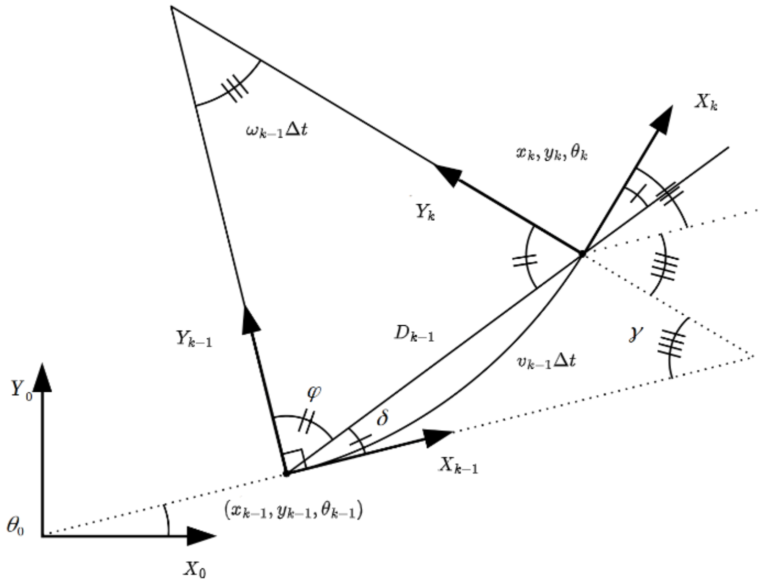

Figure 3.7: *Relevant angles from which to derive the odometry model.*

$$\phi_k^z = \phi_{k-1}^z + \omega_{k-1}\Delta t, \tag{3.46c}$$

which corresponds to a two dimensional pose, where ϕ^z is a part of R^{car}, which can be transformed to a three dimensional pose containing the vehicle states p and R^{car} with a high uncertainty in the unknown directions. A detailed description of how the odometry motion model and its Jacobian was derived can be found in Appendix A.

3.9 Camera model

The camera model used in the GTSAM toolbox is given by

$$\begin{bmatrix} u \\ w \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathcal{K}} \underbrace{\begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \end{bmatrix}}_{[R|T]} \begin{bmatrix} L^x \\ L^y \\ L^z \\ 1 \end{bmatrix}, \tag{3.47}$$

where

$$T_x = -\begin{bmatrix} R_{11} & R_{12} & R_{13} \end{bmatrix} p_k, \tag{3.48a}$$

$$T_y = -\begin{bmatrix} R_{21} & R_{22} & R_{23} \end{bmatrix} p_k, \tag{3.48b}$$

$$T_z = -\begin{bmatrix} R_{31} & R_{32} & R_{33} \end{bmatrix} p_k. \tag{3.48c}$$

It is possible to find the pixel coordinates $[u, w]^T$ of a projected landmark with the global navigation frame coordinates $[L^x, L^y, L^z]^T$, given the camera matrix $\mathcal{K}$ and rotation translation matrix $[R|T]$ which corresponds to the camera's pose. The f_x and f_y elements in the camera matrix $\mathcal{K}$ denote the focal lengths of the camera expressed in pixels, while the elements c_x and c_y denote the principal point in pixels, which is the center of the captured images. The rotation translation matrix $[R|T]$ transforms the landmark's global coordinates $[L^x, L^y, L^z]^T$ to the camera-oriented coordinate system $[L^{c,x}, L^{c,y}, L^{c,z}]^T$. The camera coordinate system is oriented in such a way that the x and y axis forms the image plane, while the z-axis is perpendicular to the image plane. The camera model (3.47) can be written in a simpler form to clearly show the relation between the camera coordinate system and the pixel coordinates $[u, w]^T$ as

$$\begin{cases} L^c = R(L - r) \\ L^{c,x'} = \frac{L^{c,x}}{L^{c,z}} \\ L^{c,y'} = \frac{L^{c,y}}{L^{c,z}} \\ u = f_x \cdot L^{c,x'} + c_x \\ w = f_y \cdot L^{c,y'} + c_y \end{cases}, \tag{3.49}$$

where u and w correspond to the output from the measurement function h^{proj} that gives the landmark's estimated pixel coordinates based on the current camera pose. The measurement y^{proj}, on the other hand, simply is the landmark's pixel coordinates in the image. The camera model Jacobian for a monocular pinhole camera can be found by first expressing a landmark in the local camera coordinate system with

$$L^c = R(L - r). \tag{3.50}$$

A projection of the landmark is then given by

$$z = \begin{bmatrix} \frac{L^{c,x}}{L^{c,z}} & \frac{L^{c,y}}{L^{c,z}} & 0 \end{bmatrix}^T, \tag{3.51}$$

where the principal point c has been dropped since it is placed in $c_x = c_y = 0$ in this thesis. The measured projection coordinates y^{proj} from the measurement model h^{proj} are found from (3.51) and the camera matrix $\mathcal{K}$ according to

$$y^{proj} = \mathcal{K}z. \tag{3.52}$$

From (3.50)–(3.52) the Jacobian is then

$$\frac{\partial y^{proj}}{\partial L} = \frac{\partial y^{proj}}{\partial z}\frac{\partial z}{\partial L^c}\frac{\partial L^c}{\partial r}\frac{\partial r}{\partial L} = -\mathcal{K}\frac{\partial z}{\partial L^c}R, \tag{3.53}$$

where

$$\frac{\partial y^{proj}}{\partial z} = \mathcal{K}, \tag{3.54a}$$

$$\frac{\partial L^c}{\partial r} = -R, \tag{3.54b}$$

$$\frac{\partial z}{\partial L^c} = \begin{bmatrix} \frac{1}{L^{c,z}} & 0 & -\frac{L^{c,x}}{(L^{c,z})^2} \\ 0 & \frac{1}{L^{c,z}} & -\frac{L^{c,y}}{(L^{c,z})^2} \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.54c}$$

$$\frac{\partial r}{\partial L} = 1. \tag{3.54d}$$

3.9.1 Map data

The map data consists of topographical data and images. The topographical data is a point-cloud data set with a density of up to 15-20 points per square-meter. This data was collected by a private company, ordered by Linköping municipality, in 2013 [29].

The map image data is a combination of global coordinates and aerial footage that are distributed by the Swedish Authority Lantmäteriet [30]. The image data makes it possible to associate each landmark L_i with an estimated global coordinate $[L_i^x, L_i^y, L_i^z]^T$. An uncertainty is added to these coordinates as the exact global coordinate can not be extracted from the map, which enables the SLAM algorithm to alter the landmark coordinates.

3.10 Performance metrics

To analyse the estimation a few different metrics regarding the performance were computed. The first is the well known mean squared error (MSE)

$$MSE = \frac{1}{N}\sum_{i=1}^{N} \left\| p_i - \hat{p}_i \right\|_2^2 \tag{3.55}$$

where $p_i - \hat{p}_i$ corresponds to the error between the true and estimated vehicle positions. The squared Euclidean norm is then used to find the MSE for the entire trajectory. The second method is called average normalised estimation error squared (ANEES) and takes the estimated covariance into account and is formulated as

$$ANEES = \frac{1}{N}\sum_{i=1}^{N} (p_i - \hat{p}_i)^T \Sigma_i^{-1} (p_i - \hat{p}_i) \tag{3.56}$$

with the marginal covariance matrix Σ_i, given by the localisation system at iteration i. The covariance was obtained after the entire simulation since it may change when new measurements are obtained due to the smoothing implementation. The optimal value of ANEES in this case is 3 (because we analyse the error with 3 degrees of freedom) which means that the estimated error and covariance are equal. ANEES equal to 3 means that the model is credible while values larger or smaller mean that the model is either optimistic or pessimistic [31].

The third and final method was to analyse the indicated standard deviation. The standard deviation was obtained by taking the square root of the diagonal in the covariance matrix Σ_i at each iteration i.

4 Method

To be able to evaluate the proposed localisation system, data has been collected with a real vehicle. The vehicle was equipped with cameras, GNSS receivers, a controllable board fixture, IMU, wheel encoders, and computers during the data collection. Data were collected along three different tracks on the public road with different speed limits, amount of turns, and landmarks. The collected data contained IMU measurements for the inertial navigation, wheel encoder data for the odometry model, and GNSS measurements to generate ground truth data for the tracks and images along the tracks from which to find visible landmarks. A simulated camera was later used to project the landmarks instead of finding the landmark pixel coordinates in the real images. This was done in order to enable better comparison between different landmark setups as the IMU and odometry data would be the same for the different camera settings, making it possible to draw conclusions based on the camera and landmark constraints alone. The simulated camera is assumed to have no restrictions regarding the speed at which it can rotate in different directions. The simulated camera data consisted of each image's landmark projection pixel coordinates and the corresponding camera pose relative to the vehicle. The images collected with the real camera contained information on what was seen in the real scenario, which was used to implement realistic landmark selections and compare them with the map data. This made it possible to present realistic localisation estimates for the vehicle along the three tracks.

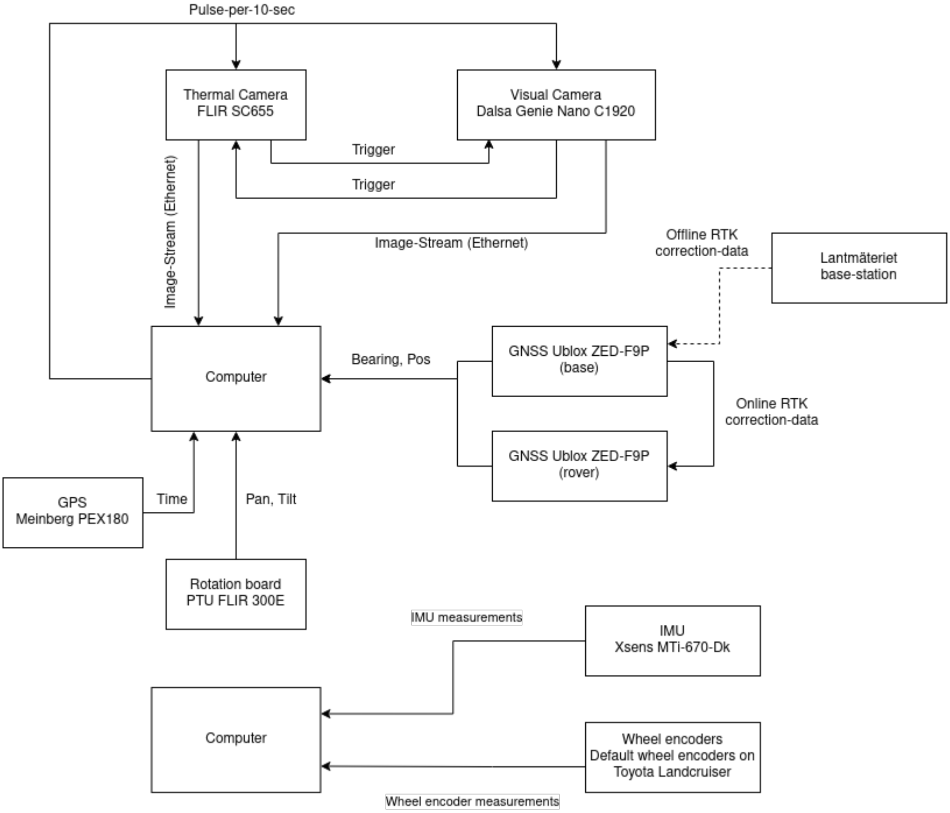

Figure 4.1: *The data collection system.*

4.1 Data collection

The general system used to collect the image, GNSS, odometry and IMU data can be found in Figure 4.1. The data was collected in and around the suburb Lambohov, near Linköping in Sweden.

Here follows a description of the different system components and their interactions with each other, as found in Figure 4.1.

Vehicle

The vehicle used during the data collection was a Toyota Landcruiser, as seen in Figure 4.2. This vehicle had been used for similar purposes before and was therefore already prepared with various mountings on the roof.

Computers

Two regular computers were used for the data collection. One computer was used to collect the images, GNSS signals, and steer the control board, while the

__Figure 4.2:__ The vehicle used during the data collection, equipped with the data collection equipment.

other computer stored the IMU and odometry data. These computers acted as two separate data collection systems.

GPS

One computer was fitted with a GPS time receiver (Meinberg PEX180). This was to prevent any clock drifts during the data collection. The vehicle was also fitted with two GNSS receivers (Ublox Zed-F9P) which were used to generate the RTK data. The RTK data processing was handled by FOI.

Landmäteriet base-station

In order to increase the accuracy of the RTK data, it was post-calibrated with the phase of the GNSS carrier wave, which Lantmäteriet collected. This data post-calibration was handled by FOI.

Cameras

Both a thermal camera (FLIR SC655) and a visual camera (Dalsa Genie Nano C1920) were used. The thermal camera would capture an image when it received a signal from the computer, and would in turn send a trigger signal to the visual camera. The visual camera would then capture an image and send a trigger signal back to the thermal camera, confirming that the sequence was completed and the computer would then store the paired images. The image signal stream was handled through ethernet. The thermal camera was used to complement the data collection for future use for FOI, and the only image data of interest for the estimation model was from the visual camera. The visual camera data was mainly of interest during the evaluation of a realistic landmark setting as a simulated camera was ultimately implemented to replace an actual implementation of the captured images.

Pan/tilt unit

Both cameras were mounted on a pan/tilt unit (PTU FLIR 300E). The pan/tilt unit was controlled through a joystick, enabling the cameras to be controlled independently of the vehicle's orientation. This makes it possible to aim the camera at landmarks, or in general directions of interest.

IMU

The IMU used was an MTi-670-Dk from the supplier Xsens, and it was mounted in the vehicle above the center of the rear wheel axis to minimise any leverage interactions from the vehicle's rotation. The IMU data collection was done using a different computer than what was used to collect GPS and camera data.

Wheel encoders

The wheel encoders were pre-installed in the vehicle by the manufacturer. The data was collected from the controller area network bus. The same computer that was used to collect IMU data was used to collect the wheel encoder data.

4.2 Data collection trajectories

Three different tracks were used to gather data, hereafter referred to as the country roads track, the highway track, and the urban track in accordance with their distinguishable characteristics.

4.2.1 Country roads track

The country roads track data was mainly collected on country roads. The path was 3930 meters long and took 448 seconds to complete. The RTK path can be found in Figure 4.3. The track characteristics can be divided into segments along the path. Initially, houses are visible around the vehicle which provides multiple clear landmarks. After approximately 100 meters, there are no close visible houses as the initially visible houses are hidden behind a small hill. For the following 1100 meters, the landscape consists of open fields and sparse vegetation. A few buildings are visible along the way but most of the time the scenic view consists of open fields, traffic signs, power lines, and sparse vegetation. Close to the first roundabout, there is a large building that is easy to identify from approximately 300 meters. After the first roundabout, there is a fair amount of

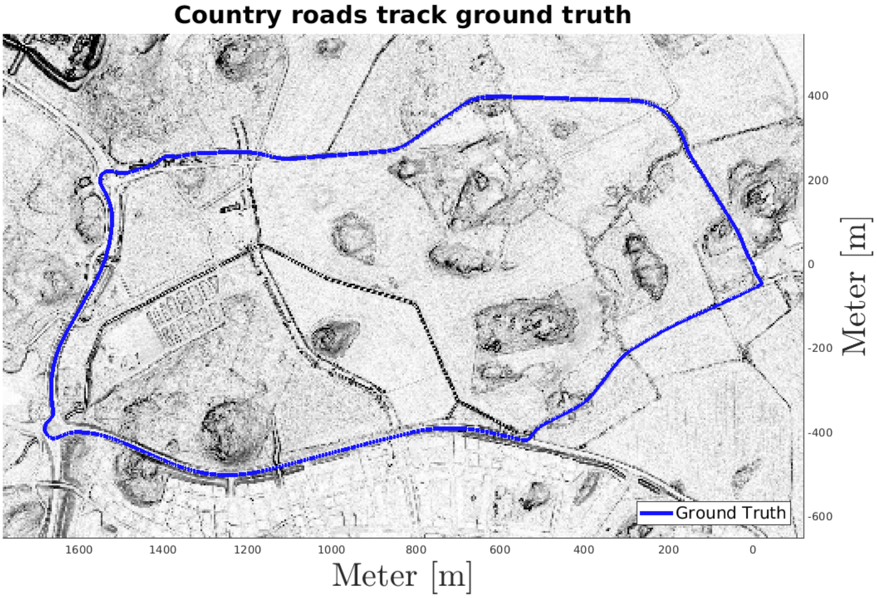

Figure 4.3: *The ground truth trajectory for the country roads track. The vehicle travelled counterclockwise along the path.*

distinguishable landmarks until the last 400 meters of the track. In these last 400 meters, there are once again mostly open fields and sparse vegetation that are visible. The houses that were initially visible are hidden behind vegetation and are therefore not visible until the end of the path. This track was selected as it contained different segments which were of interest, both the open fields, the roundabouts, and various speed sections. Generally, along the open fields, there was not much traffic to take into account, enabling a constant speed to be maintained during large sections of the path.

4.2.2 Highway track

The highway track data set contained both low and high-speed sections. The path was 6568 meters long and took 675 seconds to complete. The RTK path can be found in Figure 4.4. The vehicle travelled at a higher speed during sections of this data set compared to the other data sets. Many landmarks were visible along the track, both houses, bushes, bus stops, trees, and power lines. The vehicle travelled clockwise during the data collection. The highway track begins and ends in a parking lot, and stretches towards an industrial area after approximately 1000 meters. Here the track circle around an area containing both vegetation and buildings, only to reconnect with itself. The track then leads to the highway, where a much higher speed is maintained compared to the other

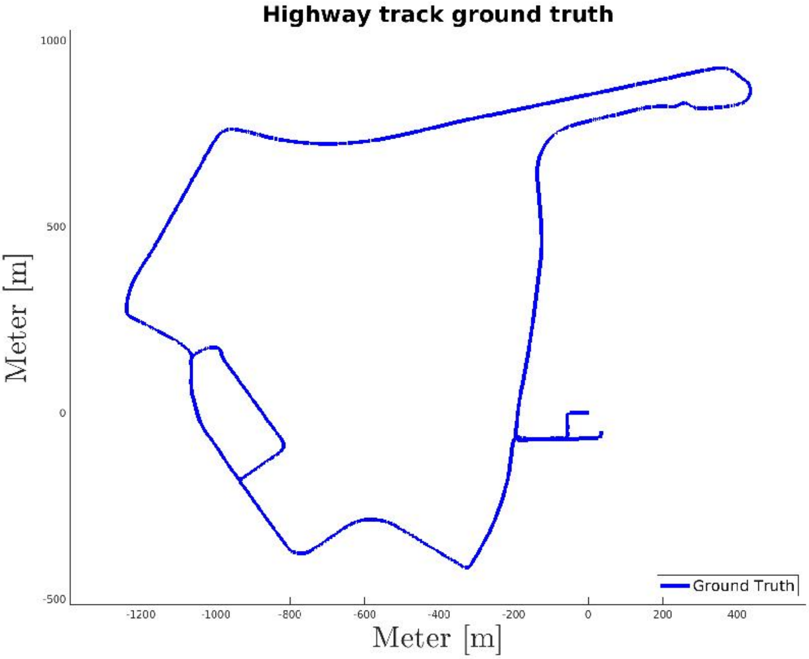

Figure 4.4: The ground truth trajectory for the highway track.

tracks, ultimately reaching a roundabout from where the initial parking lot is the final destination during a path with moderate speed and partial vegetation.

The highway track was selected mainly due to the high-speed sections, as an addition to the other tracks. Due to issues with the data, it was not possible to give a visual representation of the highway track terrain, so the background is disregarded from these plots.

4.2.3 Urban track

In the urban track data set, the vehicle maintained a very low speed with many sharp turns and regular stops in an urban area. The path was 2489 meters long and took 587 seconds to complete. The RTK path can be found in Figure 4.5. The urban track characteristics were unique compared to the country roads track and the highway track, as an abundance of clearly distinguishable landmarks are visible in each image, which all were very close to the path of the vehicle. There are many houses, road signs, intersections, and vegetation visible in each image. Due to the proximity of the houses, few images capture possible landmarks at further distances. At the beginning of the path, the vehicle travelled north until a roundabout was encountered, at which point the vehicle returned from where it came. It then travelled along narrow roads with many close buildings, until it eventually returned to the approximate starting area.

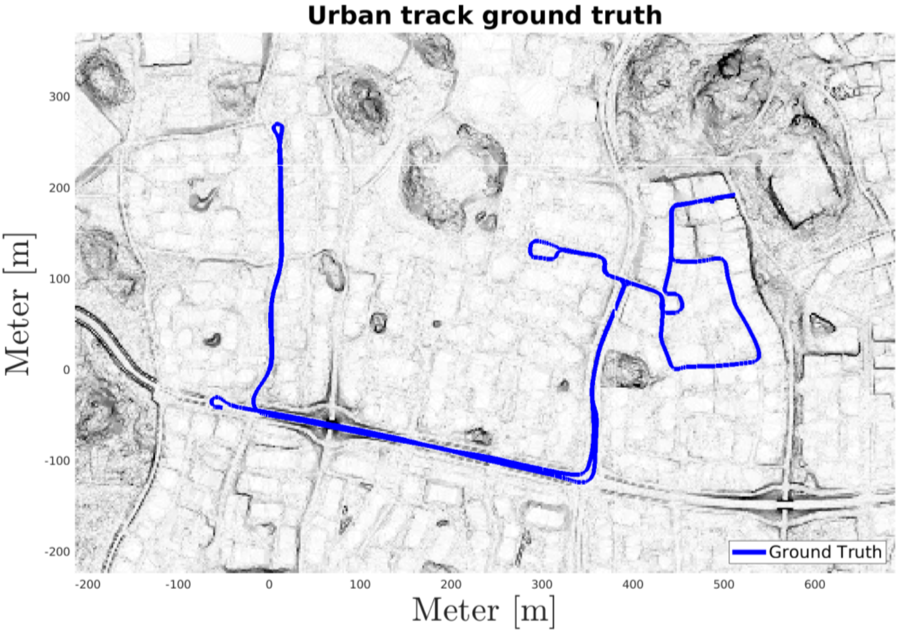

Figure 4.5: *The ground truth trajectory for the urban track.*

The urban track was selected due to the low speed, the many sharp corners, and the complex path.

4.3 Data processing

Due to the usage of two separate computers during the data collection, there was an issue when matching the timestamps at which the data was collected between these computers. This has in turn most likely led to small deviations between the data sets. This has affected the result to some degree and is difficult to accurately compensate for due to the data collection setup.

4.3.1 Generating the RTK

RTK data was generated from the two GNSS measurements. From the RTK data, a heading could also be calculated. The data was also post calibrated using Lantmäteriet Base-station to generate accurate data.

4.3.2 Ground truth poses

In order to generate simulated camera data, both the relative pose between the vehicle and camera, as well as the landmark projection in the images had to be generated. This information would be found from the control signal to the pan/tilt unit and the landmarks found in the images respectively during real data collection. Both the true vehicle poses and simulated landmarks was required to generate this data. The first part of retrieving the poses was to make sure that the measurements from the different sensors were in sync. All data from the three different tracks were collected on the same day, and the GNSS measurements were collected continuously throughout the entire data collection session of all three tracks. The IMU and odometry data however, were only collected while driving along the three tracks. To synchronize the measurements, the RTK data was generated from the GNSS measurement and was then differentiated to estimate a velocity, which was matched to the velocity obtained from the odometry model. These were lined up for each trajectory, making sure that the measurements with the right time stamps were used with as good accuracy as possible.

The vehicle poses were generated from IMU, odometry, and RTK data, and were found through iSAM2, generating an estimate of the pose with each RTK measurement point. These vehicle pose estimates were then treated as ground truth and were used to generate the relative camera pose. Camera poses could then be set with zero translation difference from the ground truth vehicle poses and rotated independently depending on which landmark it was supposed to project.

By placing landmarks along the tracks and aiming the simulated camera towards these, it was possible to define which landmarks were visible from the camera poses, and the simulated camera could project these landmarks in an image as if captured from the real path, to generate the pixel coordinates $(u, w)^T$. The generated pixel coordinates are then used as the simulated images with landmark projections for the different setups. The relative pose between the vehicle and camera, and the landmark projections were estimated for all ground truth vehicle poses along the track in this manner.

4.4 Description of the localisation system

The localisation system generates and appends state information to a factor graph, which then is transformed into Bayes net, from which the MAP estimate can be found. The general localisation system algorithm can be found in Algorithm 2.

Algorithm 2 The localisation algorithm

The localisation system creates factor graphs with each new image, containing information about the variables at that time. The iSAM2 solver transforms these factor graphs into a Bayes net and finds the MAP estimate. The algorithm variable notation coincides with Figure 3.1.

1. Landmark positions are generated from the map data or simulated according to the test scenario.
2. A factor graph is generated and the landmarks are appended as variables L_i, with a priori factors $\psi_{landmark}(L_i)$ to constrain the landmark positions. Each landmark has a unique identifier.
3. A priori which will be set on the first vehicle variable M_1 is appended to the factor graph as a factor $\psi_{car}(M_1)$.
4. An image j is captured, and a camera and vehicle variable is added to the factor graph. The j:th image will generate the j:th camera and vehicle variable, i.e, the first image will generate C_1 and M_1 and so forth.
5. The INS and odometry model is appended as the factors $\psi_{INS}(M_{j-1}, M_j)$ and $\psi_{odo}(M_{j-1}, M_j)$ between the vehicle´s new and previous pose, M_j and M_{j-1}.
6. The control board settings are used to generate the relative pose factor $\psi_{cc}(M_j, C_j)$ between the vehicle M_j and camera C_j variables.
7. Landmarks are identified in the captured image, and the pixel coordinates in the image which best denote the landmark positions from the map data are selected and added to the factor graph as the projection factor $\psi_{proj}(C_j, L_i)$. If there are image data denied periods, the landmark projection data will be discarded during these measurement periods.
8. The factor graph is added to the iSAM2 solver which transforms the graph into a Bayes net and stores it in a Bayes tree.
9. The iSAM2 solver finds the MAP estimate and updates the uncertainties and variable estimates, including the landmark positions.
10. With each new image captured, the algorithm returns to step 4.

4.5 Parameters

Three important aspects that had an impact on the result were image frequency, landmark selection, and covariances. Different image frequencies and landmark setups were tested for the three tracks.

4.5.1 Image frequency

To answer the problem statement and evaluate the impact of the image frequency, both constant and varied image frequency was tested. The image frequency was set to 0.1 Hz, 0.5, Hz, 1 Hz, and 2 Hz for the constant image frequency tests.

While testing varied image frequencies, two different setups were tested. Firstly, the image frequency was randomly varied in an interval of 0.1-2 Hz. The purpose behind this test was to simulate a camera trigger based on landmark identification, which most likely would not remain constant throughout the tracks due to varied visibility. Secondly, longer periods of no visible landmarks were simulated, denying the system any landmark observations. The purpose was to be able to evaluate if it is possible to see landmarks and capture images for a period of time, followed by a period of time where no landmarks can be seen. This would be repeated during the path 6 times. The periods where no landmarks could be identified were set to 10, 20, and 40 seconds for each track. During these periods of no visual landmarks, the factor graph still received new information at the same rate as when landmarks were visual, with the difference that all landmark projection data was discarded. This allowed the estimate to be updated based on the INS and odometry model alone.

4.5.2 Landmark setups

To adequately answer the problem statement, different landmark setups were used for the same track in the simulations.

Evenly distributed landmarks

The evenly distributed landmarks setup corresponds to landmarks appearing evenly along the ground truth trajectory. Further, the landmarks alter between being on the right and left side of the path, and are located approximately the same distance from the path. This setup was chosen as it gives uniform constraints for the model, meaning it always has available landmarks that follow the same distance pattern between landmark and vehicle path. This landmark setup was used while testing the image frequency impact on the estimate.

Different number of landmarks

In the different number of landmarks setup, multiple landmarks could be found in one image. More landmark observations in each image should intuitively result in more constraints and therefore also a better estimate of the camera pose. For this test, it is of interest to analyze how the number of landmarks seen in each image affects the localisation estimate. The setup was implemented by adding additional landmarks a few meters next to each landmark in the evenly distributed landmarks setup.

Sparse landmark selection

The sparse landmark setup has much fewer landmarks compared to the other setups. The setup can be used to evaluate the localisation performance if only a distant landmark, such as a church tower, can be identified from the map. To evaluate the localisation performance in these cases, two landmark setups will be tested. Firstly when only one distant landmark is identified along the entire track, and secondly when two supporting landmarks are added along the track but only visible within a short range.

Realistic landmarks selection

The last landmark setup, realistic landmarks selection, is chosen to be as realistic as possible. This is done by identifying landmarks in the real images and correlating these to the map data. Some landmarks are assumed to be visible in the simulated images despite not being visible in the real images due to the orientation of the camera.

4.5.3 Model covariance, uncertainties, and initialization

The covariance and bias uncertainties used for the IMU can be seen in Table 4.1-4.2. The covariance is taken from the datasheet while the bias uncertainties were initialised with high values since the system computes the bias for each set of preintegrated measurements and low initial uncertainties showed poor results. Note that there are only three covariance values each for the gyroscope and accelerometer. These are used as the diagonal of a covariance matrix where the non-diagonal values are set equal to zero.

Table 4.1: *IMU covariance*

	$\sigma_x^{\dot{\phi}}$	$\sigma_y^{\dot{\phi}}$	$\sigma_z^{\dot{\phi}}$
Gyroscope $[\frac{rad^2}{s^2}]$	$0.17 \cdot 10^{-3}$	$0.17 \cdot 10^{-3}$	$0.17 \cdot 10^{-3}$
	σ_x^a	σ_y^a	σ_z^a
Accelerometer $[\frac{m^2}{s^4}]$	$0.87 \cdot 10^{-5}$	$0.87 \cdot 10^{-5}$	$0.87 \cdot 10^{-5}$

Table 4.2: *IMU bias uncertainties*

	$\dot{\phi}_x\,[\frac{rad}{s}]$	$\dot{\phi}_y\,[\frac{rad}{s}]$	$\dot{\phi}_z\,[\frac{rad}{s}]$	$a^x\,[\frac{m}{s^2}]$	$a^y\,[\frac{m}{s^2}]$	$a^z\,[\frac{m}{s^2}]$
σ_{bias}	1	1	1	0.5	0.5	0.5

Moreover, the implemented initial pose uncertainties can be seen in Table 4.3. The uncertainties σ, similar to the IMU covariance, are the elements in the diagonal matrix corresponding to the factor graph variables. The values are arbitrarily selected with respect to the order of the expected values. How these are implemented is described in Section 4.4.

Table 4.3: *Initial pose uncertainties*

	ϕ^x $[rad]$	ϕ^y $[rad]$	ϕ^z $[rad]$	p^x $[m]$	p^y $[m]$	p^z $[m]$
σ_{car}	0.05	0.05	0.05	0.1	0.1	0.1
σ_{camera}	0.05	0.05	0.05	0.1	0.1	0.1
$\sigma_{landmark}$	-	-	-	0.1	0.1	0.1
$\sigma_{carCamera}$	0.0001	0.0001	0.0001	0	0	0

Furthermore, the projection uncertainty, presented in Table 4.4, is a normally distributed noise of 0.2 pixels that are added to the projection received from the camera projection factor. This is done to simulate some measurement noise to the projected landmarks. The purpose of this is to get a closer correspondence to the projections from a real camera. The initial velocity uncertainty can be seen in Table 4.5.

Table 4.4: *Projection uncertainty*

	$u\,[pixel]$	$w\,[pixel]$
$\sigma_{projection}$	0.2	0.2

Table 4.5: *Initial velocity uncertainty*

	v^x $[\frac{m}{s}]$	v^y $[\frac{m}{s}]$	v^z $[\frac{m}{s}]$
$\sigma_{velocity}$	0.0028	0.0028	0.0028

5

Result

The result section has been divided into segments corresponding to the problem, where each section consists of a different setup regarding the image frequency or landmark setup. The results from the country roads track can be found in the result section, while the urban and highway tracks results can be found in Appendix B. The vehicle pose translation that was created from odometry, IMU, and RTK data as described in Section 4.3.2 is considered to be the ground truth path. The performance is measured by absolute positional error, the estimates' standard deviation, MSE, and ANEES as described in Section 3.10. These performance data are calculated on an image frequency basis, so each image captured generates a data point. During periods of denied image data, the landmark projections are disregarded but the model still receives new data from the INS and Odometry model. The p^x, p^y, and p^z-positions in the figures regarding the absolute positional error and standard deviation refer to the global navigation frame coordinate system as previously described.

5.1 Ground truth result

In Table 5.1 the MSE values between the ground truth poses, generated as described in Section 4.3.2, and the RTK measurements are shown.

Table 5.1: *MSE between the ground truth translation and RTK data.*

	MSE [m]
Country roads track	$7.6183 \cdot 10^{-5}$

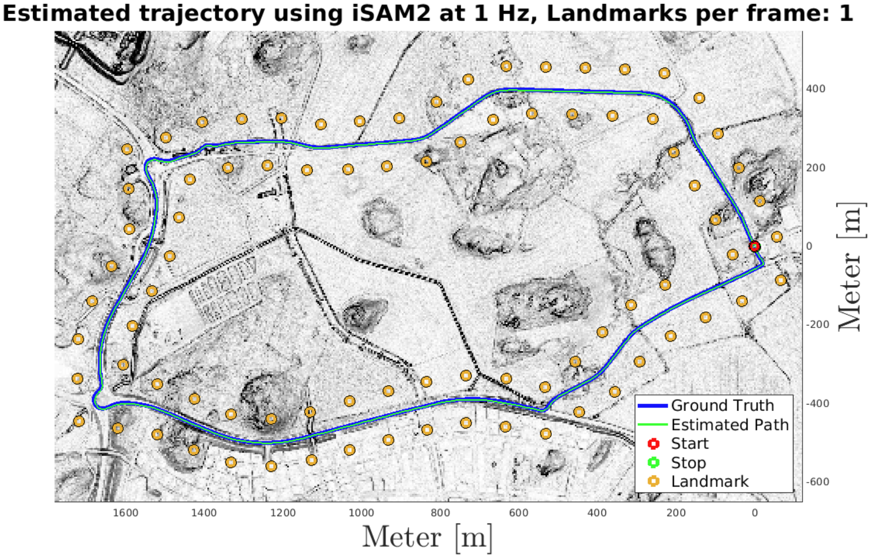

Figure 5.1: *The estimated path for the country roads track compared to ground truth with 1 Hz image frequency.*

5.2 Image frequency

The image frequencies tested were 0.1 Hz, 0.5 Hz, 1 Hz, and 2 Hz. The landmark setup remained the same while testing the different image frequency settings, with the landmarks being placed 40-60 meters from the path, with at least 110 meters between each landmark. Only one landmark was visible in each image. The resulting MSE and ANEES values can be found in Table 5.2 for the country roads track at the different image frequency settings.

Table 5.2: *MSE and ANEES of the estimated path at different image frequencies.*

	0.1 Hz	0.5 Hz	1 Hz	2 Hz
MSE Country roads track [m]	5.4643	1.154	0.7266	0.56464
ANEES Country roads track [-]	3.8592	9.2137	12.5022	17.0959

The landmark setup for the country roads track can be seen in Figure 5.1

The error between the estimate and ground truth, as well as the approximated standard deviation in each image captured along the country roads track, can be found in Figure 5.2. During the data collection, 45, 224, 448, and 896 simulated images were captured at 0.1, 0.5, 1, and 2 Hz respectively.

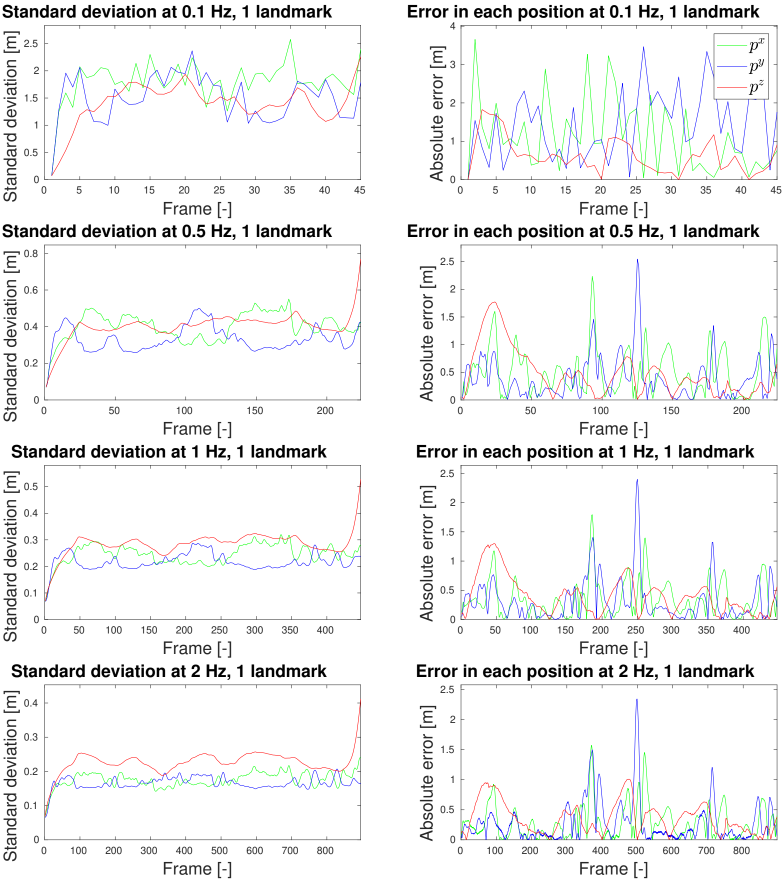

Figure 5.2: *Absolute errors and the standard deviations of the estimates at 0.1, 0.5, 1 and 2 Hz image frequency during the country roads track.*

5.3 Varied image frequency

The image frequency was varied along the path. This was done in two different setups. For the first setup, an image was captured in an interval of 0.5 - 10 seconds after the previous image. The time in the interval was randomised, hence a slightly different estimation can be expected each time. For the second setup, the localisation system was denied new images for a fixed amount of time during certain parts of the track.

5.3.1 Frequency interval 0.1 - 2 Hz

With a varied image frequency in the interval of 0.1 - 2 Hz, a of total 448 simulated images were captured for the country roads track, which is the same amount of images as a constant image frequency of 1 Hz would generate. The path was estimated five times in order to describe the characteristics of the varied frequency impact more accurately, as the varied frequency would yield different results for each iteration. The generated MSE and ANEES values for these tests can be found in Table 5.3.

Table 5.3: *MSE and ANEES of the estimated country roads track at varied image frequencies.*

	MSE [m]	ANEES [-]
Test 1	1.5233	22.5381
Test 2	1.3137	18.4358
Test 3	1.419	22.4734
Test 4	1.571	22.3003
Test 5	1.1907	18.2362

5.3.2 Denied images

The results when the system was denied new image data for parts of the path, in terms of MSE and ANEES, can be seen in Table 5.4.

Table 5.4: *MSE and ANEES of the estimated country roads track with periods of denied image data.*

	Denied image time	MSE [m]	ANEES [-]
Test 1	10	0.8675	11.0352
Test 2	20	1.2363	10.5455
Test 3	40	2.2984	10.0073

The estimated path for the country roads track where the localisation system is denied image data for 40 seconds 6 times along the track can be found in Figure 5.3, along with the covariance ellipse for the estimated path.

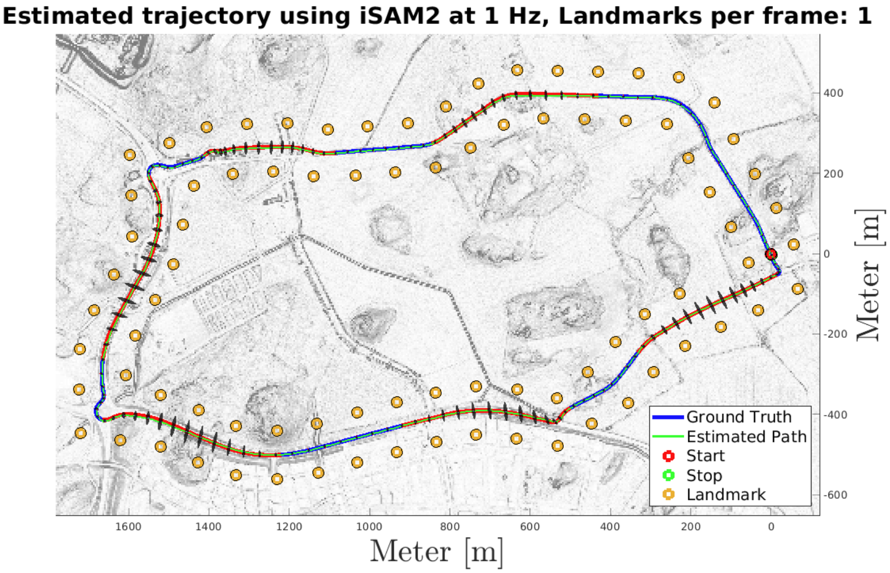

Figure 5.3: *Estimated path for the country roads track with uncertainties and 40 seconds of denied image data at 6 instances.*

In Figure 5.4 the absolute errors and standard deviations are presented for 10, 20 and 40 second periods of denied image data.

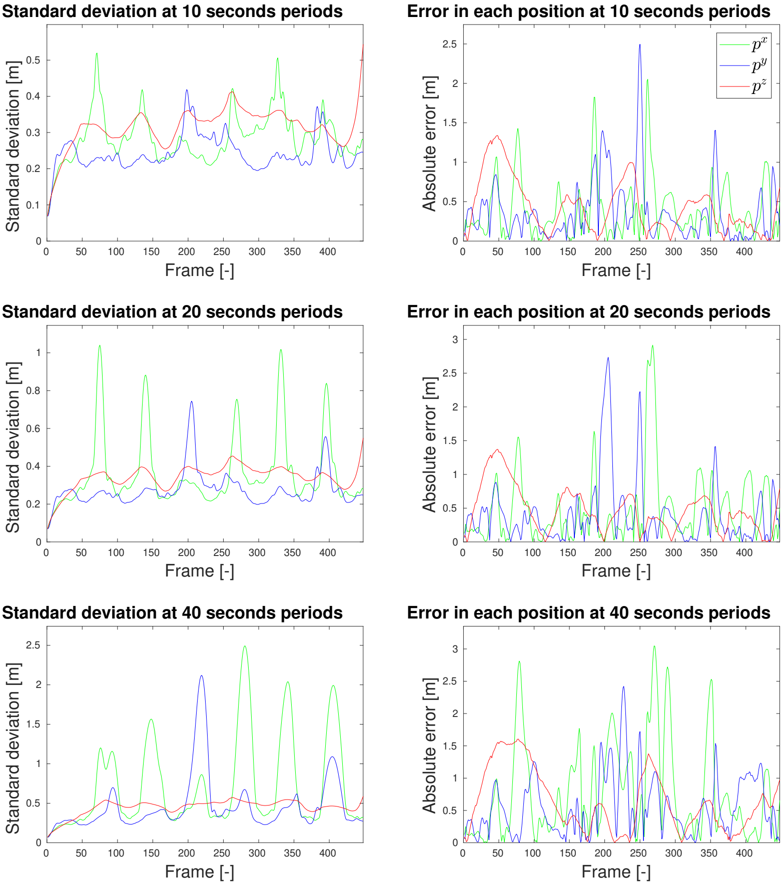

Figure 5.4: *Absolute errors and the standard deviations of the estimates at 1 Hz image frequency for the country roads track, with 6 instances of 10, 20, and 40 seconds denied image data.*

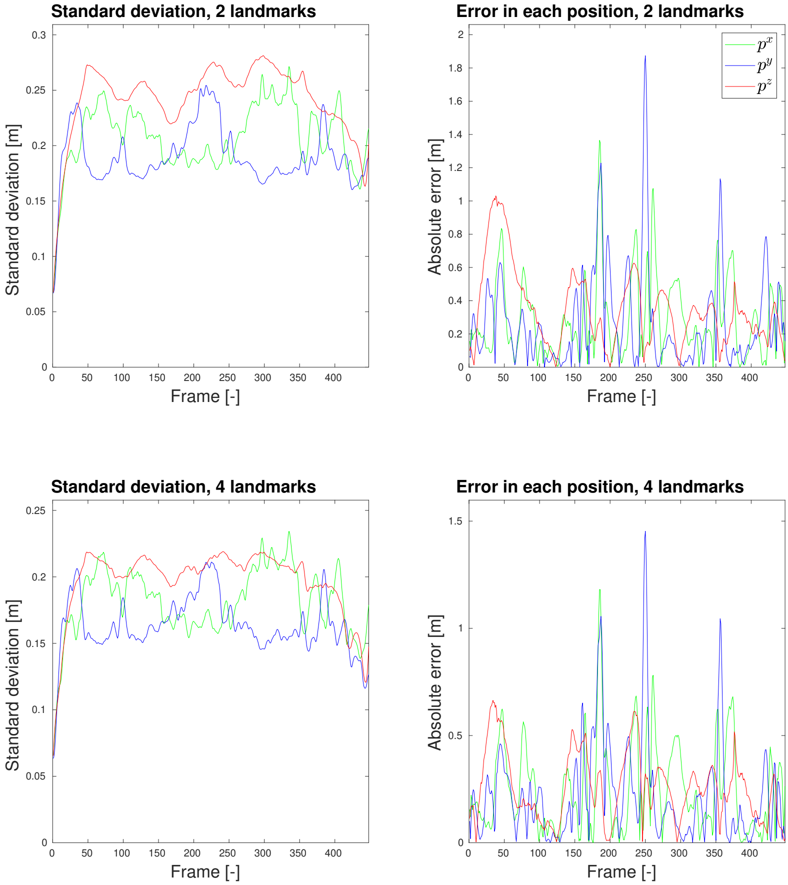

Figure 5.5: *Absolute errors and the standard deviations of the estimates at 1 Hz during the country roads track with 2 and 4 landmarks visible in each image.*

5.4 Amount of visible landmarks per image

Using the same landmark setup as in Section 5.2, different amounts of visible landmarks in each image were tested. Either two or four visible landmarks in each image were simulated. The MSE and ANEES values for the country roads track can be seen in Table 5.5. In Figure 5.5 the resulting standard deviation and absolute error using 2 and 4 landmarks can be seen for the country roads track.

Table 5.5: *MSE and ANEES of the estimated path with multiple visible landmarks.*

Visible in each image	2 Landmarks	4 Landmarks
MSE Country roads track [m]	0.56138	0.47978
ANEES Country roads track [-]	11.9004	11.5932

5.5 Sparse landmark selection

For these tests, a landmark selection as described in Section 4.5.2 was used. Two different setups were tested for the tracks. In the first setup, only a distant landmark was visible in each image, and in the second setup two additional landmarks were added, but they were only visible if the vehicle was within a range of 30 meters from the landmarks. These tests were performed with an image frequency of 1 Hz. The country roads track presents figures for both the standard deviation and absolute error in the estimates for one distant landmark and where two additional landmarks are used, to visualize the difference.

While testing these setups on the country roads track, a landmark was placed 5 km west, 1 km north and 100 meters above the vehicle starting position. The results for this sole landmark can be seen in Figure 5.7. For this track, two complementary landmarks were added at one roundabout and one cross-section and the results can be seen in Figure 5.8. The values of MSE and ANEES for the two tests can be found in Table 5.6.

Table 5.6: *Values of MSE and ANEES for the two sparse landmark setups.*

	MSE [m]	ANEES [-]
Distant landmark	88.9824	53.7327
3 landmarks	13.6859	40.5009

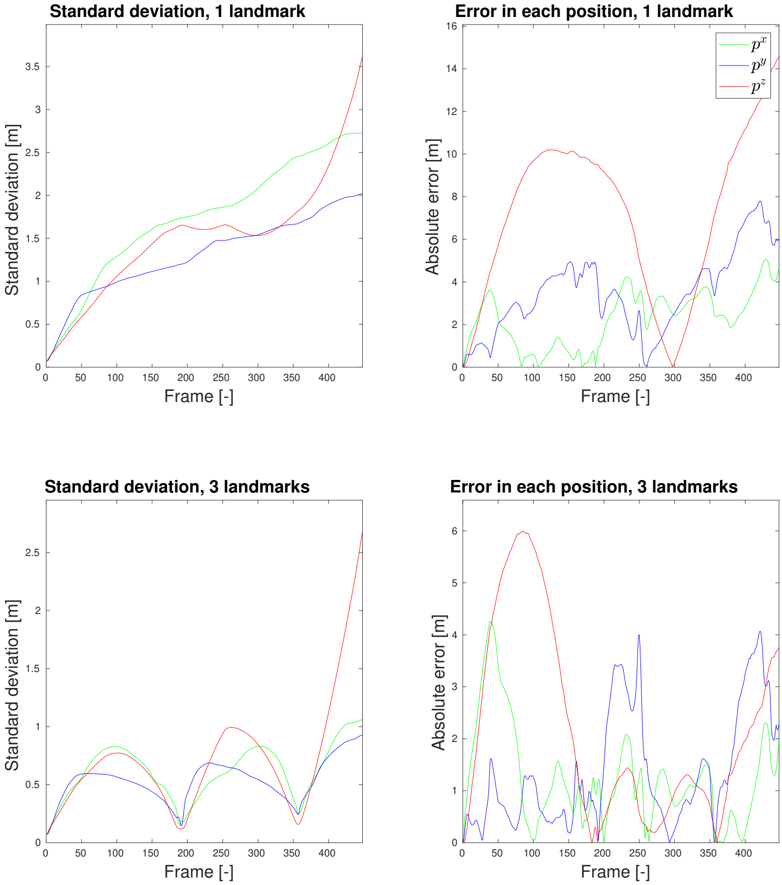

Figure 5.6: *Absolute errors and the standard deviations of the estimates at 1 Hz during the country roads track with one unique landmark, and where two complementary landmarks are added along the track.*

Furthermore, the path when using one distant landmark during the entire track can be found in Figure 5.7, and the path with the additional complementary landmarks can be seen in Figure 5.8, both with their covariance ellipses.

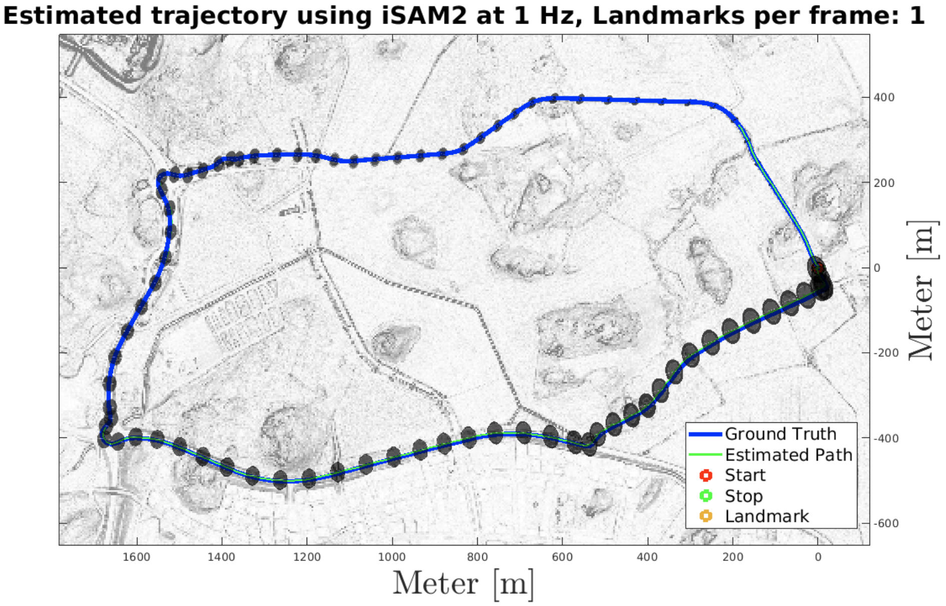

***Figure 5.7:** Estimated path with uncertainties using a distant landmark the entire trajectory.*

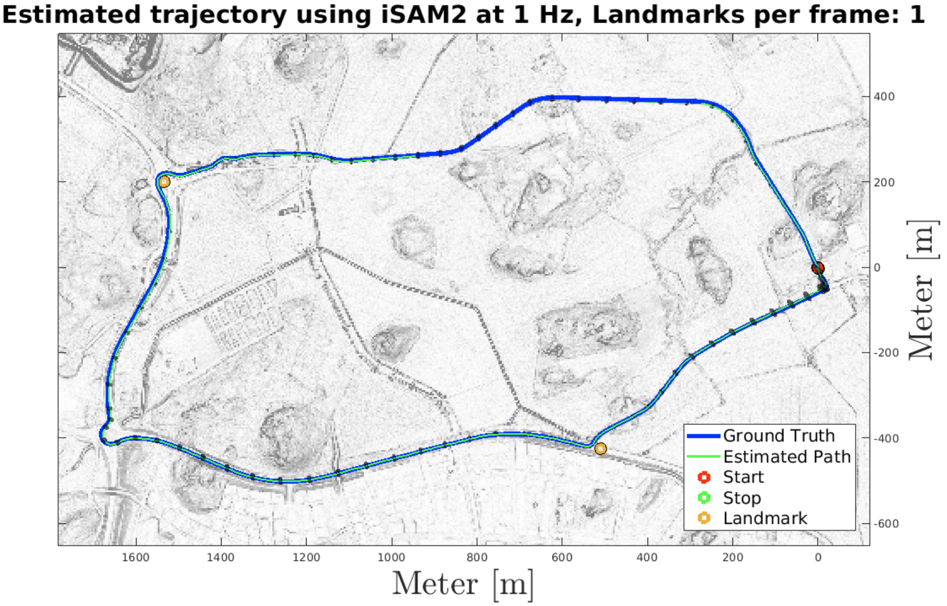

***Figure 5.8:** Estimated path with uncertainties using a distant landmark and two complementary landmarks.*

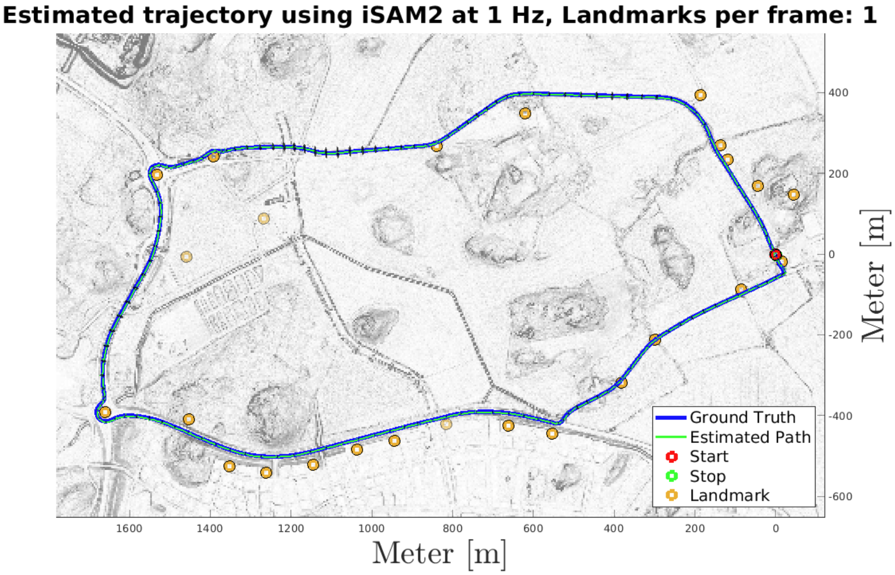

Figure 5.9: *The estimated path with realistic landmarks for the country roads track.*

5.6 Realistic landmark selection

For this part, the landmarks were selected according to Section 4.5.2 from the real images. The results are presented as an estimated path, absolute error, and standard deviation for the country roads track. The values for MSE and ANEES can be found in Table 5.7.

Table 5.7: *MSE and ANEES of the estimated country roads track at realistic landmark setups.*

	MSE [m]	ANEES [-]
Country roads track	0.54252	12.6524

The estimated path for the country roads track with realistic landmarks is shown in Figure 5.9 while the absolute errors and standard deviations are shown in Figure 5.10.

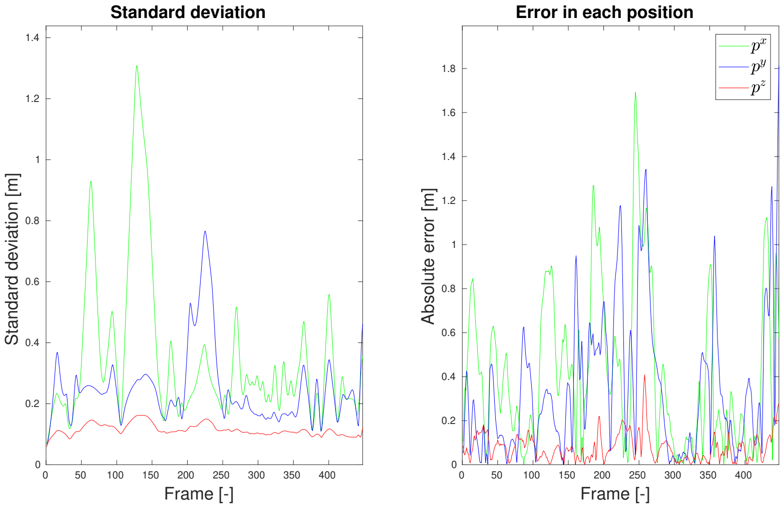

Figure 5.10: *The estimated path with realistic landmarks for the country roads track.*

6 Discussion

Here, the findings in Section 5 and how the findings were obtained in Section 4, are discussed and analysed. The general work in a wider context is also discussed.

6.1 Results

The different setups in Section 5 are discussed with respect to the problem statement.

6.1.1 Ground truth

The estimated ground truth vehicle poses had a similar translation compared to the RTK data, as seen by the MSE between the two data sets in Tables 5.1 and B.1. This can be interpreted as the ground truth having high accuracy with regard to the vehicle's position during the tracks. The MSE value for the urban track is higher compared to the other tracks which could suggest that some interference occurred during the data collection session. During the urban track, there were multiple speed bumps that could introduce noisy measurements compared to the other tracks, but it should not be significant enough to affect the entire estimation to this degree.

Furthermore, it is difficult to determine the credibility of the vehicle's orientation estimate regarding roll and pitch since they do not get any constraints from the odometry model as the yaw does. The RTK data generated from the GNSS measurements needed to be synced with the odometry model and IMU measurements to generate the vehicle pose estimates. This sync was sensitive, where just a small sync error would generate poor pose estimates. As these poses were then used for the simulated camera, they could have had an impact on the estimation.

6.1.2 Image frequency

Three different image frequency settings were tested to evaluate the impact of the image frequency on the path estimation.

Constant frequencies

The general correlation between image frequency, MSE, and ANEES are found in Table 5.2 for the country roads track, and in Tables B.2 and B.3 for the highway and urban tracks respectively. The findings show that the estimate becomes better with higher frequency, but also more overconfident. The increased accuracy is intuitive as more constraints and smaller integration drifts should yield a better estimate. The reason the ANEES increases is however an issue that can have multiple explanations. The optimal trend would be that the ANEES remains the same or approaches its optimal value as more images are included in the estimation.

One of the probable reasons is that the measurements consist of both collected data and simulated data, which do not fully coincide. Despite the high accuracy between the estimated ground truth path and RTK data, there is a pose difference. Another possible reason is that with higher image frequency, the preintegrated IMU factor and odometry factor gets lower uncertainties due to fewer integrated measurements between images. This is also taken into account by the estimation. While the error decreases between the estimate and ground truth, the uncertainty decreases even more. This could result in higher ANEES for the higher frequencies. Another possible explanation is that the camera-related uncertainties are set too low, and do not yield the increased accuracy that the model expects. Comparing the tracks, a significant improvement can be found at 0.5 Hz compared to 0.1 Hz. At 1 Hz the MSE is below 1 m for the country roads and highway tracks with ANEES values of 12.5 and 14.2. At 2 Hz the MSE is lower but the ANEES values become 17.1 and 22.4 which states higher estimation overconfidence.

Looking at the country roads track error and standard deviation in Figure 5.2, it is clear that the general characteristics are similar between 0.5, 1, and 2 Hz. The resulting error at 0.1 Hz has poor resolution and does not follow the same characteristics as the higher frequencies, particularly after 25 images. The error peaks are visible throughout each of these plots, where the most distinguishable error peaks are reflected as an increased standard deviation for those images. Looking at the uncertainties, they also get lower with a higher frequency with similar characteristics. It is clear however that the higher frequency affects the p^x- and p^y-position uncertainties more than p^z-position uncertainty. A likely explanation for this is that the landmarks are set with no significant translation in p^z-position compared to the track, which would pose less of a constraint in this direction.

The highway track characteristics found in Figure B.2, are harder to interpret. There are fewer characteristics in the error plot, and the errors are poorly represented in the standard deviation plot. There is one clear characteristic across the standard deviation plots, that the uncertainty at images 45, 230, 470, and 920 are surrounded by a spike in p^y position at 0.1, 0.5, 1, and 2 Hz respectively. A correlating absolute error for this increased uncertainty can be found at 0.1 and

0.5 Hz, but not at 1 and 2 Hz in the figure. The error coincides with the turn in the top right of Figure B.1. With higher frequency, this uncertainty becomes less distinct, as does the error.

The urban track characteristics found in Figure B.4 show the same characteristics in the absolute error and standard deviation plots for 0.5, 1, and 2 Hz. Similar to the country roads track 0.1 Hz provides a poor resolution with a significantly worse estimate. During the midsection of the urban track, a significant error spike occurs in both p^x- and p^z-position, which is reflected in the standard deviation.

These findings show that the estimation accuracy increases with frequency, but also that the estimate becomes more overconfident. This could be due to the combination of simulated and collected data, which slightly contradicts each other, resulting in growing overconfidence with frequency. Another reason could be that the camera measurement noise is underestimated in the model. To properly make use of higher frequency without compromising the estimate credibility, the model parameters should be better tuned with uncertainties and only use collected data to prevent these kinds of errors. Arguably, the trend shows that the estimation accuracy for 0.1 Hz is lower and that 0.5 Hz is the minimum frequency from which to generate a reasonable estimate. The trade-off between low MSE and the most optimal ANEES makes the frequency selection of 1 Hz the best setting for the localisation model in its current state.

Varied frequencies

Tables 5.3, B.4 and B.5 shows the MSE and ANEES values where a random uniformly distributed image frequency was used along the country roads, highway, and urban tracks respectively. Each image timestamp varied between 0.5 - 10 seconds after the previous image, resulting in the same total amount of images as if a constant frequency of 1 Hz was used, which makes it possible to compare the varied frequency setup with the constant 1 Hz setup. In Table 5.2 the MSE was found to be 0.7266 m at 1 Hz for the country roads track, and 0.69463 m and 1.4292 m at 1 Hz for the highway and urban track in Table B.2. Comparing this with the findings in Tables 5.3, B.4 and B.5, the MSE becomes worse for the country roads track and the highway track with varied image frequencies, but very similar with the urban track. The ANEES values in Table B.3 was 12.5022, 14.2338, and 28.6026 at 1 Hz. These values compared with Tables 5.3, B.4 and B.5 shows that the ANEES became significantly worse for the country roads track, slightly worse for the highway track and better for the urban track.

While the varied image frequency is difficult to interpret, the MSE becomes worse for the country roads track and urban track which has the most reliable simulated data, from which one arguably can conclude that a constant image frequency yields a better result. When it comes to a realistic scenario, it is important to know how accurate the estimate is, and with much more widely distributed MSE and ANEES values at varied frequencies, it is more reliable to use a constant frequency.

Periods of denied image data

In Tables 5.4, B.6 and B.7, the MSE and ANEES can be found from the different tracks, with a denied image data period of 10, 20 and 40 seconds 6 times along the tracks. The MSE values become higher with an increased amount of denied images, however, the ANEES values stay approximately the same for each track. This implies that the IMU and odometry uncertainties propagate fairly similarly to the error during the denied image data period. In Figures 5.3, B.5 and B.7, the estimates' covariance ellipses are plotted along the tracks when image data is denied to the model for 40 seconds intervals 6 times along the tracks. In Figure 5.3 and B.5 the ellipse sizes can be seen to increase when image data is denied and the ellipses reach their maximum size during the middle of the image data denied path, only to decrease towards the section where image data is available again. In straight parts of the paths, the uncertainty propagates faster sideways than it does in the vehicle's moving direction. For the country roads track, Figure 5.4 confirms that the standard deviation for an estimate increases significantly 6 times along the track, and scales correctly with the absolute error. The same can be said for the highway track in Figure B.6.

One interesting section is the fifth period of denied image data during the highway track in Figure B.5. It is represented in Figure B.6 as an increased absolute error and standard deviation in p^y-position. This reflects a probable reason why Figure B.7 shows higher standard deviations. The urban path contains many narrow corners, starts, and stops which makes the INS and odometry uncertainty propagate in all directions since the vehicle turns frequently. Comparing the MSE of the path estimate with denied image data from Tables 5.4, B.6 and B.7 with the MSE of the path estimate where images were captured continuously at 1 Hz in Tables 5.2 and B.2, shows that the MSE becomes higher with denied image data.

These findings show that it is possible to get a decent path estimate even when no landmarks can be found, or images captured for a period of time. However, the estimated error increases with the duration of the denied image data period and strongly depends on the track characteristics. It is better to deny image data to the localisation system during straight sections of a path than when there are many corners present.

6.1.3 Landmark selection

Three different landmark setups were investigated as presented in the result. Each of these is described for their respective section from Chapter 5.

Amount of visible landmarks per image

The first landmark setup was tested to analyse how the performance depends on the number of visible landmarks in each image. One can see that the MSE, presented in Tables 5.5 and B.8, is small for all tracks meaning that the estimation is close to the ground truth. The MSE decreases for all tracks when increasing the number of landmarks per image. This corresponds to the intuitive result; that more information gives a better estimate. This is intuitive since the landmark placement is set with high accuracy, and the camera uncertainty is relatively low compared to the IMU measurements. The values for ANEES, presented in Tables 5.5 and B.9, are higher than the optimal value of 3 meaning that the estimates are optimistic, i.e, overconfident. The ANEES values for the country roads and highway track are similar for both two and four visible landmarks per image, while the ANEES value decreases for the urban track. One reason why the urban track differs from the other two could be because of what was described in Section 6.1.1. One possible explanation for the high ANEES values can be seen from the absolute errors and standard deviations in Figure 5.5. The figures show that the standard deviations does not increase significantly at the high spikes for the errors. Since the values for standard deviation are correlated to the covariance matrix this means that the system is too certain about the estimation compared to how close to the ground truth the estimation is. Based on the findings, more landmarks per image result in a better estimate without compromising confidence, but the estimate is overconfident for all setups.

Sparse landmark selection

The sparse landmark setup was tested to see how the estimate was affected by not having many visible landmarks along the tracks. In Figure 5.7 one can see the implications when only having one landmark several kilometers away from the vehicle. The figure shows that the standard deviation is low at the start, but increases over the entire path. When comparing this with Figure 5.8 one can see the impact of identifying another landmark closer to the vehicle for just a few seconds. In the figure, one can see that the standard deviation increases between the complementary landmarks and becomes small for all nearby estimates. This happens because the entire trajectory gets re-estimated after each image, meaning that the standard deviation of the estimate will decrease both before and after a complementary landmark is seen. This can also be seen from the standard deviation result in Figure 5.6 where the standard deviation decreases distinctly when the two complementary landmarks are seen in the images. The same characteristics can be found in the results for both the highway and urban tracks.

Furthermore one can see that the MSE values in Tables 5.6, B.10 and B.11 becomes lower with the additional landmarks. The most significant improvement is for the country roads track where the MSE is six times higher while only seeing one distant landmark, compared to also seeing the two complementary landmarks. The high MSE is mainly due to the error in the p^z-direction. Despite the high error, the estimate is still overconfident as seen in the ANEES values. One explanation for this could be poor initialisation, meaning that the pose is estimated with a faulty initial pitch with too high a certainty. Since the odometry model has high uncertainties regarding the pitch, the camera becomes more important in this estimation. When adding two landmarks for just a few seconds at ground level this error becomes much smaller. However, the estimate still is still too confident compared to the resulting error. The estimation for the highway and urban track does have similar characteristics with MSE decreasing and the ANEES increasing. Only having one distinct landmark generates an estimate which deviates significantly from the ground truth track compared to the previously discussed tests. By finding two complementary landmarks, the absolute error, and standard deviation decrease significantly. Based on this it is arguably necessary to have a few complimentary landmarks if one distant landmark is mainly used.

Realistic landmark selection

The realistic landmark setup was chosen to give a closer resemblance of how the localisation system would perform in real-world settings. From Figures 5.9, B.15 and B.17 one can see the landmark setups derived from the real images. The country roads track does have some parts with sparse landmark availability. The impact of this can be seen in Figure 5.10 where path sections with sparse landmarks contain corresponding spikes in the standard deviation plot. The MSE and ANEES values, as seen in Tables 5.7 and B.12, are low which means that the path estimates overall are close to the ground truth tracks. The values for ANEES show an optimistic system that is overconfident in its estimates as previously stated. In general, the realistic landmark setup confirms that the estimates are accurate and generate a sufficient localisation estimate for the vehicle.

6.2 Method

The method used contained both positive and negative aspects. The data collection tracks and setups were performed in a structured manner, but the estimated ground truth poses for the vehicle led to issues. This could have been avoided by finding the real landmark projections instead of using a simulated camera, but that in turn would make it difficult to evaluate the different landmark setups. By using the real images, some kind of algorithm would have to be implemented to efficiently find the landmark pixel coordinates in each image. This would significantly increase the scope of this thesis, and should instead be investigated separately in future works.

In order to minimize the potential impact of temporary disturbances during the data collection, it would be optimal to collect data multiple times on the same tracks. The collected data could then be compared for different sessions on the track to potentially identify data collection issues or outside factors that should be considered. The tracks were only driven once with data being collected in this thesis. The reasoning behind this was that it would have been more time-consuming and result in too much data to analyze. However, it would be beneficial to investigate this in future works.

The data should also have been collected on one computer instead of two, so the timestamps can be accurately used to structure the data. This was a mistake during the data collection which proved to take a lot of time to counteract. It is also probable that this affected the outcome of the result.

The result was also affected by the set system parameters. The uncertainties set in the system need to be tuned further, to ensure the camera, IMU, and odometry uncertainties propagate realistically throughout the estimated tracks. This is one probable reason why the ANEES values for almost every estimate are higher than the optimal ANEES value.

Further, the landmark and frequency setups that were tested in the estimation were adapted to the problem statement. It covered all parts that were to be investigated. More testing could have been done between the different setups to show other interesting trends, but that would be beyond the scope of the problem statement.

7
Conclusions and future work

In this chapter, conclusions will be made concerning the problem statement. Interesting areas for further research will also be presented.

7.1 Conclusions

Based on the findings it is possible to draw conclusions for all the questions in the problem statement. The realistic landmark selection reflects how an actual implementation of the localisation system would perform. Compared to the ground truth, the localisation system estimated the path within an MSE of 0.4 - 0.63 m for the three tracks which arguably is a good estimate.

The MSE becomes lower with a higher image frequency, however, due to modelling errors, the estimate becomes overconfident. Two probable reasons behind this were found, both based on incorrect modelling of certainties. Either the INS and odometry integration factor uncertainty did not propagate fast enough at high frequencies, or the camera related uncertainty was set too low. A higher frequency did however yield a better result, which coincides with the intuitive solution. There was a clear difference between using a constant image frequency of 1 Hz and varying the time steps between images along the tracks, even if the same total amount of images were used. In short, this implies that a constant frequency is preferable. The localisation system showed promising results while it was denied new image data, as the esimate still was perceived as good. The uncertainty propagated in an intuitive manner during the periods of denied image data, and a decent estimation of the vehicle path was maintained aswell, which is a valuable attribute in a realistic implementation of the localisation system.

While testing how the localisation estimate was affected by having multiple landmarks visible in each image, the clear trend was that the estimate became better, with lower MSE values, and the ANEES values were fairly similar for these tests. The largest impact was seen during the urban track, as its MSE value improved more than the other tracks, while its ANEES value went from 17.87 to 13.38 as 2 and 4 landmarks were visible respectively. This implies that multiple landmarks in the images lead to a better path estimate without compromising estimate confidence.

Having a sparse landmark setup poses some significant issues. With only one visible landmark along the track, the localisation system presents a poor estimate, particularly if there are large distances between the track and the landmark. A few landmarks along the track improved the estimate significantly and is arguably necessary to maintain a good path estimate.

The localisation system shows promise and could prove beneficial for society as a localisation estimation method where no GNSS signals are available. However, no association issues were considered between landmarks and map data. In a real implementation, this would have to be evaluated as well. Due to the generated ground truth poses for the vehicle, which was of importance for the simulated cameras, these findings can contain slight misrepresentations of a true localisation system implementation.

7.2 Future work

There are several aspects of this thesis that could be interesting to develop further. To begin with, it would of course be more realistic if the real image data was used to create the projection constraints. This would require some image-based recognition algorithm, and would need to be very accurate so as not to impose misassociations in the localisation system. Currently, the localisation system relinearises often which makes the processing speed slow. This could be further improved which would be required for real applications. The localisation system parameters in terms of uncertainties need to be more precisely tuned. The current setup continuously proves to be overconfident, which can and should be improved.

Appendix

A
Theory

A.1 Odometry motion model

The odometry motion model is derived from Figure 3.7. The initial state of the local coordinate system is set as (X_0, Y_0, θ_0). With a measured angular velocity ω_k, time step Δt_k and velocity v_k it is possible to calculate the distance travelled for each k in the (X_0, Y_0, θ_0) coordinate system. In order to find the distance D_k, the velocity in direction X_k needs to be projected onto D_k, integrated with respect to the time step Δt_k, and then divided into its corresponding Δp_k^x and Δp_k^y components. The Δt_k, Δp_k^x and Δp_k^y components are defined as

$$\Delta t_k = t_{k+1} - t_k \tag{A.1a}$$

$$\Delta p_k^x = p_{k+1}^x - p_k^x \tag{A.1b}$$

$$\Delta p_k^y = p_{k+1}^y - p_k^y \tag{A.1c}$$

To project the velocity onto D_k, the angle δ needs to be found. From Figure 3.7, three triangles can be described by

$$\omega_k \Delta t_k + 2\varphi = \pi \tag{A.2a}$$

$$\omega_k \Delta t_k + \frac{\pi}{2} + \gamma = \pi \tag{A.2b}$$

$$\delta + \gamma + \pi - \varphi = \pi \tag{A.2c}$$

By replacing γ and φ in (A.2c) with φ from (A.2a) and γ from (A.2b), δ can be described as a function of $\omega_k \Delta t_k$ as

$$\delta = \frac{\omega_k \Delta t_k}{2} \tag{A.3}$$

The distance travelled D_k is then found by integrating the projected velocity.

$$V_D = V_k \cos\delta \tag{A.4a}$$

$$D_k = \int_{t_1}^{t_2} V_D \, dt \tag{A.4b}$$

$$D_k = \frac{2V_k}{\omega} \sin\frac{\omega_k \Delta t_k}{2} \tag{A.4c}$$

The Δx_k and Δy_k components can then be derived from (A.4c) together with the angle relative to the initial coordinate system (X_0, Y_0, θ_0) by projecting D_k on the (X_0, Y_0) axes.

$$\Delta p_k^x = \frac{2V_k}{\omega_k} \sin\left(\frac{\omega_k \Delta t_k}{2}\right) \cos\left(\frac{\omega_k \Delta t_k}{2} + \theta_k\right) \tag{A.5a}$$

$$\Delta p_k^y = \frac{2V_k}{\omega_k} \sin\left(\frac{\omega_k \Delta t_k}{2}\right) \sin\left(\frac{\omega_k \Delta t_k}{2} + \theta_k\right) \tag{A.5b}$$

The total translation in time step k in the (X_0, Y_0, θ_0) coordinate system can be calculated from (A.1b)–(A.1c) and (A.5a)–(A.5b), while the state transition of θ_{k+1} is purely additive between the k and $k+1$ time step, resulting in the full state transition expressions (A.6a)–(A.6c) for the odometry model.

$$p_{k+1}^x = p_k^x + \frac{2V_k}{\omega_k} \sin\left(\frac{\omega_k \Delta t_k}{2}\right) \cos\left(\frac{\omega_k \Delta t_k}{2} + \theta_k\right) \tag{A.6a}$$

$$p_{k+1}^y = p_k^y + \frac{2V_k}{\omega_k} \sin\left(\frac{\omega_k \Delta t_k}{2}\right) \sin\left(\frac{\omega_k \Delta t_k}{2} + \theta_k\right) \tag{A.6b}$$

$$\theta_{k+1} = \theta_k + \omega_k \Delta t_k \tag{A.6c}$$

The noise propagation w_k needs to be estimated for the model. By calculating the Jacobian of the measured data V_k and ω_k, it is possible to see how measurement noise would propagate in the time steps. In (A.7), the partial derivatives of the states concerning the measured data are calculated.

$$G = \begin{pmatrix} \frac{\partial p_{k+1}^x}{\partial V_k} & \frac{\partial p_{k+1}^x}{\partial \omega_k} \\ \frac{\partial p_{k+1}^y}{\partial V_k} & \frac{\partial p_{k+1}^y}{\partial \omega_k} \\ \frac{\partial \theta_{k+1}}{\partial V_k} & \frac{\partial \theta_{k+1}}{\partial \omega_k} \end{pmatrix} \tag{A.7}$$

Each elements in (A.7) can be found in (A.8a)–(A.8f).

$$\frac{\partial p^x_{k+1}}{\partial V_k} = \frac{2}{\omega_k} \sin\left(\frac{\omega_k \Delta t_k}{2}\right) \cos\left(\frac{\omega_k \Delta t_k}{2} + \theta_k\right) \tag{A.8a}$$

$$\begin{aligned}\frac{\partial p^x_{k+1}}{\partial \omega_k} = \frac{-V_k}{\omega_k^2}\Bigg(\Delta t_k \omega_k \sin\left(\frac{\Delta t_k \omega_k}{2}\right) \sin\left(\frac{\Delta t_k \omega_k}{2} + \theta_k\right) + \Bigg(2 \sin\left(\frac{\Delta t_k \omega_k}{2}\right) \\ - \Delta t_k \omega_k \cos\left(\frac{\Delta t_k \omega_k}{2}\right)\Bigg) \cos\left(\frac{\Delta t_k \omega_k}{2} + \theta_k\right)\Bigg)\end{aligned} \tag{A.8b}$$

$$\frac{\partial p^y_{k+1}}{\partial V_k} = \frac{2}{\omega_k} \sin\left(\frac{\omega_k \Delta t_k}{2}\right) \sin\left(\frac{\omega_k \Delta t_k}{2} + \theta_k\right) \tag{A.8c}$$

$$\begin{aligned}\frac{\partial p^y_{k+1}}{\partial \omega_k} = \frac{-V_k}{\omega_k^2}\Bigg(\Bigg(2 \sin\left(\frac{\Delta t_k \omega_k}{2}\right) - \Delta t_k \omega_k \cos\left(\frac{\Delta t_k \omega_k}{2}\right)\Bigg) \sin\left(\frac{\Delta t_k \omega_k}{2} + \theta_k\right) \\ - \Delta t_k \omega_k \sin\left(\frac{\Delta t_k \omega_k}{2}\right) \cos\left(\frac{\Delta t_k \omega_k}{2} + \theta_k\right)\Bigg)\end{aligned} \tag{A.8d}$$

$$\frac{\partial \theta_{k+1}}{\partial V_k} = 0 \tag{A.8e}$$

$$\frac{\partial \theta_{k+1}}{\partial \omega_k} = \Delta t_k \tag{A.8f}$$

The F matrix is the Jacobian of the states in time step $k+1$ with the partial derivatives with respect to the states in time step k as

$$F = \begin{pmatrix} \frac{\partial p^x_{k+1}}{\partial p^x_k} & \frac{\partial p^x_{k+1}}{\partial p^y_k} & \frac{\partial p^x_{k+1}}{\partial \theta_k} \\ \frac{\partial p^y_{k+1}}{\partial p^x_k} & \frac{\partial p^y_{k+1}}{\partial p^y_k} & \frac{\partial p^y_{k+1}}{\partial \theta_k} \\ \frac{\partial \theta_{k+1}}{\partial p^x_k} & \frac{\partial \theta_{k+1}}{\partial p^y_k} & \frac{\partial \theta_{k+1}}{\partial \theta_k} \end{pmatrix} \tag{A.9}$$

The elements in (A.9) can be found in (A.10a)–(A.10i).

$$\frac{\partial p^x_{k+1}}{\partial p^x_k} = 1 \tag{A.10a}$$

$$\frac{\partial p^x_{k+1}}{\partial p^y_k} = 0 \tag{A.10b}$$

$$\frac{\partial p^x_{k+1}}{\partial \theta_k} = -\frac{2V_k \sin\left(\frac{\omega_k \Delta t_k}{2}\right) \sin\left(\theta_k + \frac{\omega_k \Delta t_k}{2}\right)}{\omega_k} \tag{A.10c}$$

$$\frac{\partial p_{k+1}^{y}}{\partial p_{k}^{x}} = 0 \tag{A.10d}$$

$$\frac{\partial p_{k+1}^{y}}{\partial p_{k}^{y}} = 1 \tag{A.10e}$$

$$\frac{\partial p_{k+1}^{y}}{\partial \theta_{k}} = \frac{2V_k \sin\left(\frac{\omega_k \Delta t_k}{2}\right) \cos\left(\theta_k + \frac{\omega_k \Delta t_k}{2}\right)}{\omega_k} \tag{A.10f}$$

$$\frac{\partial \theta_{k+1}}{\partial p_{k}^{x}} = 0 \tag{A.10g}$$

$$\frac{\partial \theta_{k+1}}{\partial p_{k}^{y}} = 0 \tag{A.10h}$$

$$\frac{\partial \theta_{k+1}}{\partial \theta_{k}} = 1 \tag{A.10i}$$

B

Results

Here the results from the highway and urban tracks are presented. The different setups are modestly described, as they are identical to the settings presented for the country roads track in Chapter 5.

B.1 Ground truth result

In Table B.1 the MSE values between the ground truth and RTK measurements are shown.

Table B.1: *MSE between the ground truth and RTK data.*

	MSE [m]
Highway track	3.9836e-5
Urban track	0.0013308

B.2 Image frequency

The MSE and ANEES for the highway and urban track with an image frequency setting of 0.1 Hz, 0.5 Hz, 1 Hz and 2 Hz can be found in Table B.2 and B.3.

Table B.2: *MSE of the estimated path at different frequencies.*

	0.1 Hz	0.5 Hz	1 Hz	2 Hz
MSE Highway track [m]	7.7215	1.0622	0.69463	0.63192
MSE Urban track [m]	6.7864	1.9565	1.4292	1.3612

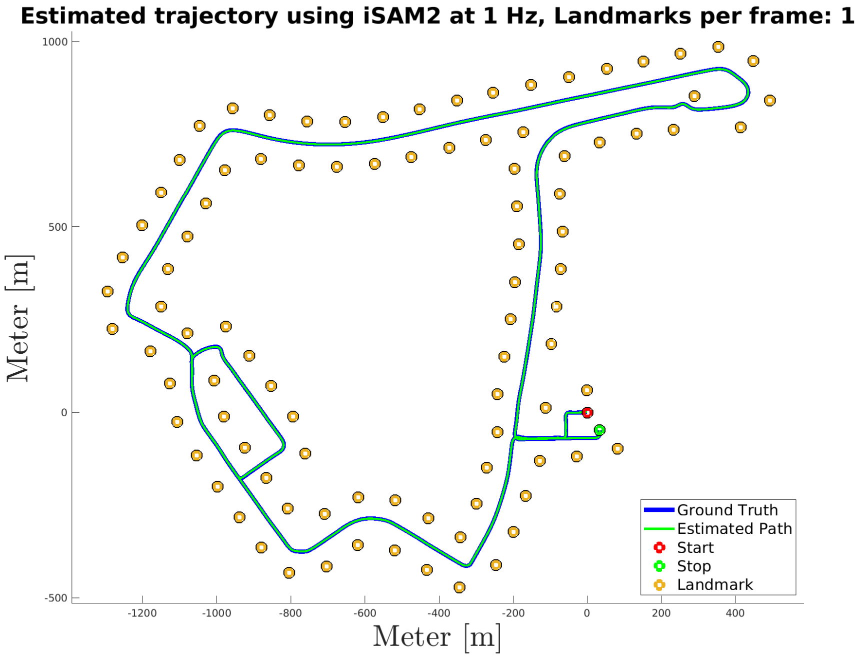

Figure B.1: *The estimated path for the highway track compared to ground truth with 1 Hz image frequency.*

Table B.3: *ANEES of the estimated path at different frequencies.*

	0.1 Hz	0.5 Hz	1 Hz	2 Hz
ANEES Highway track [-]	5.8523	9.9593	14.2338	22.4062
ANEES Urban track [-]	2.7062	16.9872	28.6026	44.3066

B.2.1 Highway track

The landmark setup for the highway track can be found in Figure B.1.

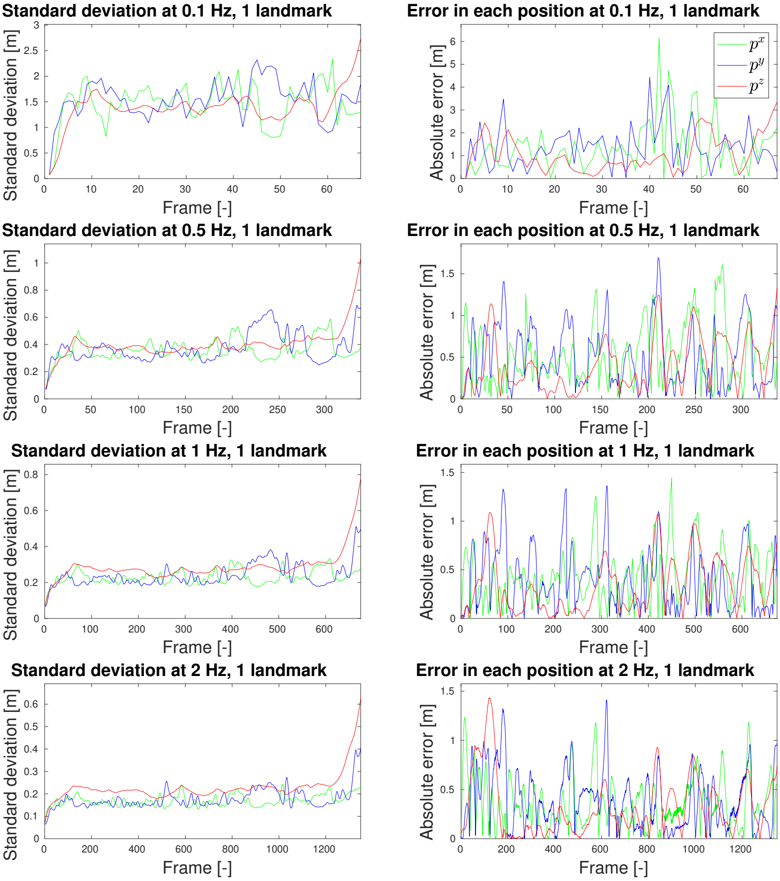

Figure B.2: *Absolute errors and the standard deviations of the estimates at 0.1, 0.5, 1, and 2 Hz image frequency during the country roads track.*

In Figure B.2 the error and standard deviation at each captured image can be found. During the data collection 67, 337, 675, and 1349 virtual images were captured at 0.1, 0.5, 1, and 2 Hz respectively.

B.2.2 Urban track

The landmarks used during the frequency testing for the urban track can be found in Figure B.3.

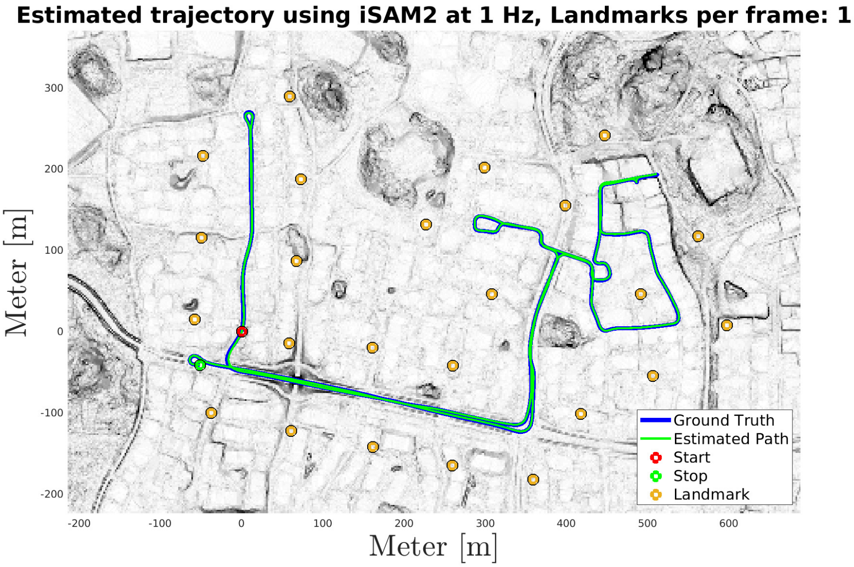

Figure B.3: The estimated path for the urban track compared to ground truth with 1 Hz image frequency.

The error between the estimate and ground truth, as well as the approximated standard deviation in each image data point along the country roads track, can be found in Figure B.4. During the data collection 57, 286, 572, and 1143 virtual images were captured at 0.1, 0.5, 1, and 2 Hz respectively.

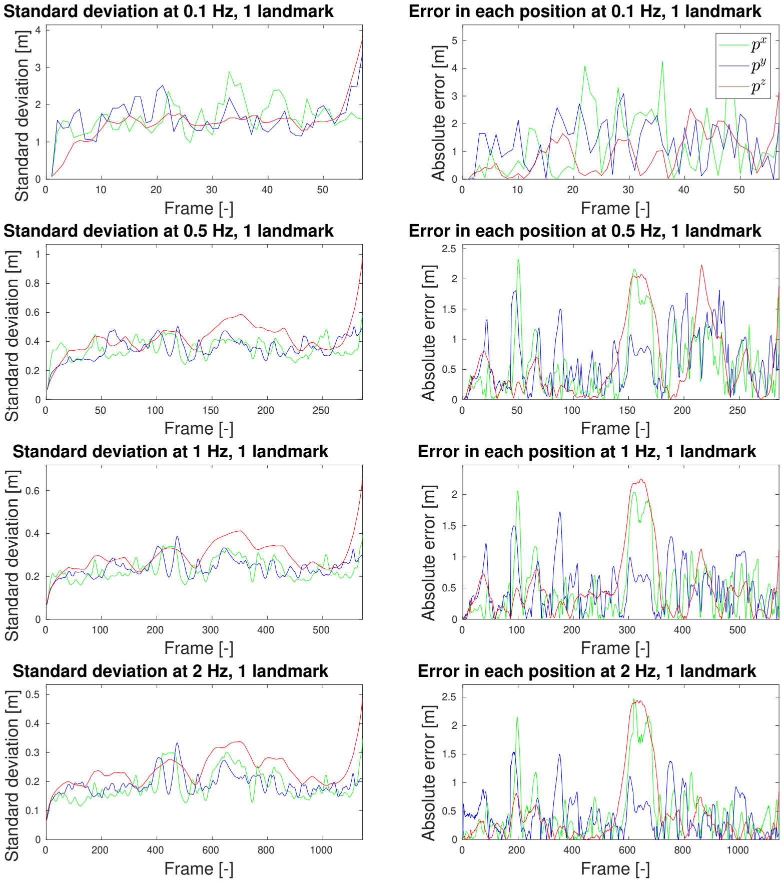

Figure B.4: Error and the standard deviations of the estimates at 0.1, 0.5, 1, and 2 Hz image frequency during the country roads track.

B.3 Varied image frequency

Two image frequency settings were tested. Firstly where an image was captured in an interval of 0.5 - 10 seconds after the previous image, and secondly where new images were denied for a set period of time, 6 times along the tracks.

B.3.1 Frequency interval 0.1 - 2 Hz

In total 675 and 572 virtual images were captured for the highway track and urban track respectively, which is the same amount of images as a constant image frequency of 1 Hz would generate. The generated MSE and ANEES values for these iterations can be found in Table B.4 - B.5.

Table B.4: *MSE and ANEES of the estimated highway track at varied frequencies.*

	MSE [m]	ANEES [-]
Test 1	0.8266	15.0623
Test 2	0.87566	15.1269
Test 3	0.93802	16.8289
Test 4	0.88803	16.3797
Test 5	0.85592	14.0831

Table B.5: *MSE and ANEES of the estimated urban track at varied frequencies.*

	MSE [m]	ANEES [-]
Test 1	1.8809	27.0327
Test 2	1.2991	21.2511
Test 3	1.1868	19.6703
Test 4	1.6288	26.1947
Test 5	1.3321	20.6331

B.3.2 Denied images

The results in terms of MSE and ANEES while denying image data to the estimate during parts of the trajectory can be found in Table B.6 - B.7.

Table B.6: *MSE and ANEES of the estimated highway track at varied frequencies.*

	Duration of denied image data	MSE [m]	ANEES [-]
Test 1	10	0.76659	12.7941
Test 2	20	1.0278	13.4491
Test 3	40	2.171	12.8501

Table B.7: *MSE and ANEES of the estimated urban track at varied frequencies.*

	Duration of denied image data	MSE [m]	ANEES [-]
Test 1	10	1.3741	23.7388
Test 2	20	1.6043	22.4938
Test 3	40	2.9575	24.8455

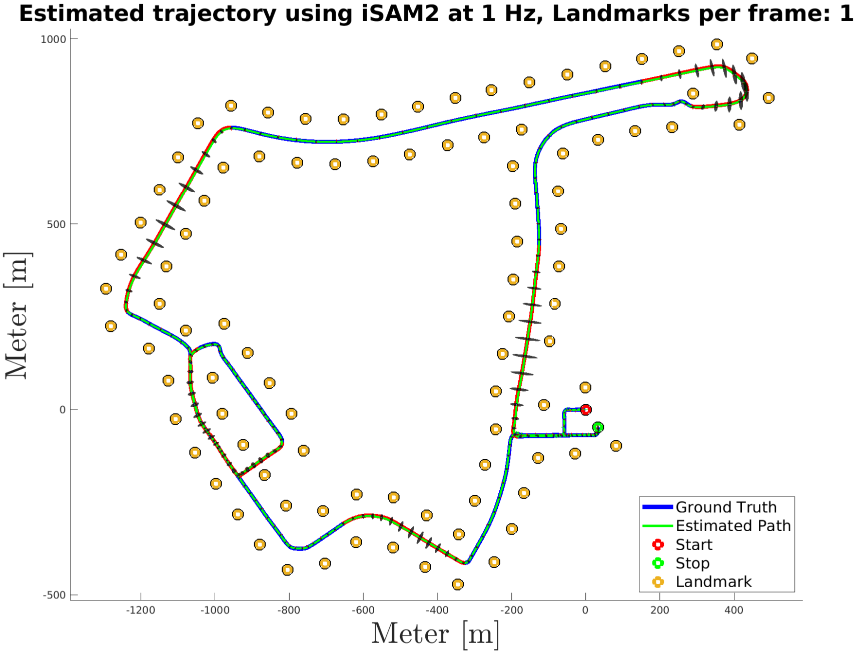

Figure B.5: *Estimated path for the highway track with uncertainties and 40 seconds of no camera measurements at 6 instances.*

Highway track with denied images

For the highway track the estimated path, when not taking images for 40 seconds at 6 instances, can be seen in Figure B.5.

Furthermore, the standard deviations and absolute errors for the three different time intervals can be seen in Figure B.6.

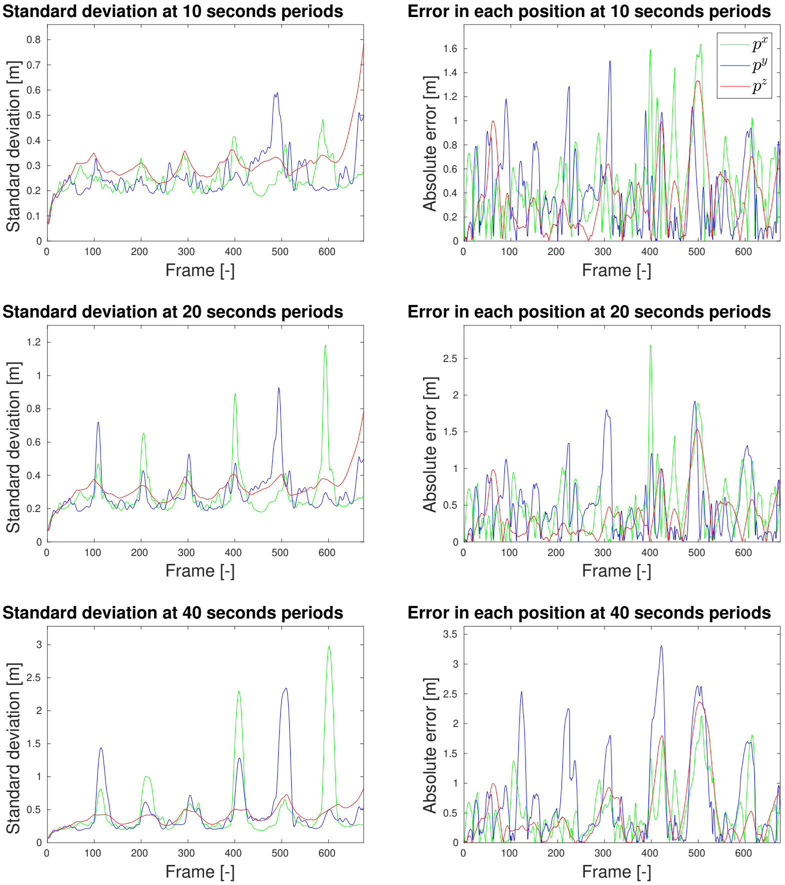

Figure B.6: Absolute errors and the standard deviations of the estimates at 1 Hz image frequency for the highway track, with 6 instances of 10, 20, and 40 seconds denied image data periods.

Urban track with denied images

The estimation of the urban track, with 40 seconds of no camera measurements, can be seen in Figure B.7.

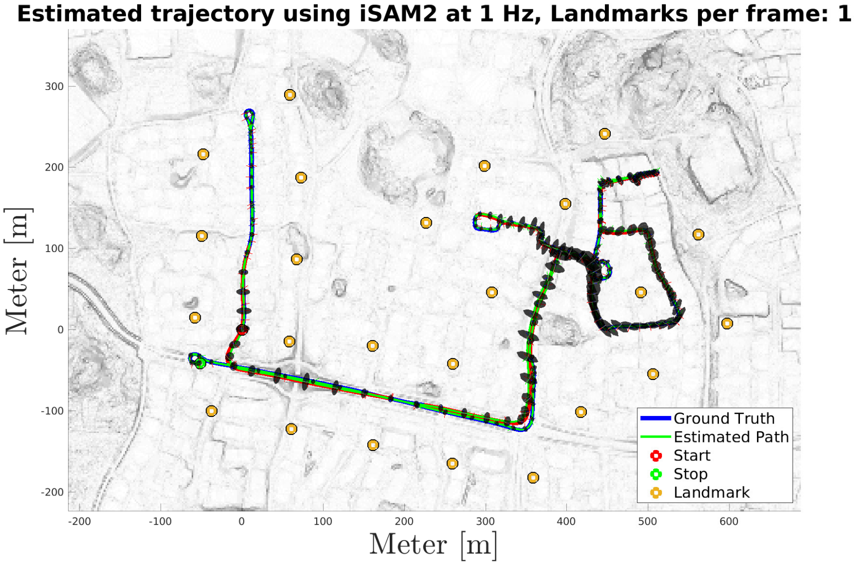

Figure B.7: *Estimated path for the urban track with uncertainties and 40 seconds of no camera measurements at 6 instances.*

In Figure B.8 the absolute errors and standard deviations are presented where 10, 20, and 40 seconds periods of image data is denied.

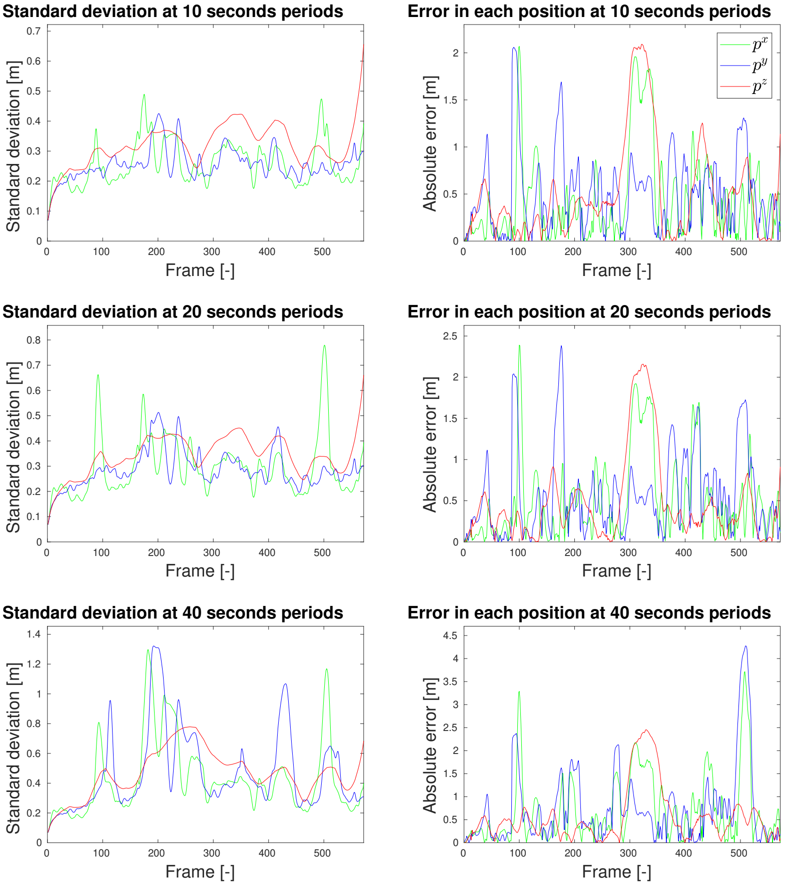

Figure B.8: *Absolute errors and the standard deviations of the estimates at 1 Hz image frequency for the urban track, with 6 instances of 10, 20, and 40 seconds denied image data periods.*

B.4 Amount of visible landmarks per image

The values for MSE and ANEES for the highway and urban tracks when two or four landmarks were visible in each image can be seen in Table B.8 and B.9.

Table B.8: *MSE of the estimated path with multiple visible landmarks.*

Visible in each image	2 Landmarks	4 Landmarks
MSE Highway track [m]	0.52473	0.40376
MSE Urban track [m]	0.52037	0.30561

Table B.9: *ANEES of the estimated path multiple visible landmarks.*

Visible in each image	2 Landmarks	4 Landmarks
ANEES Highway track [-]	11.2073	11.6676
ANEES Urban track [-]	17.8713	13.3808

B.4.1 Highway track

For the highway track, the results can be seen in Figure B.9 when using 2 or 4 landmarks.

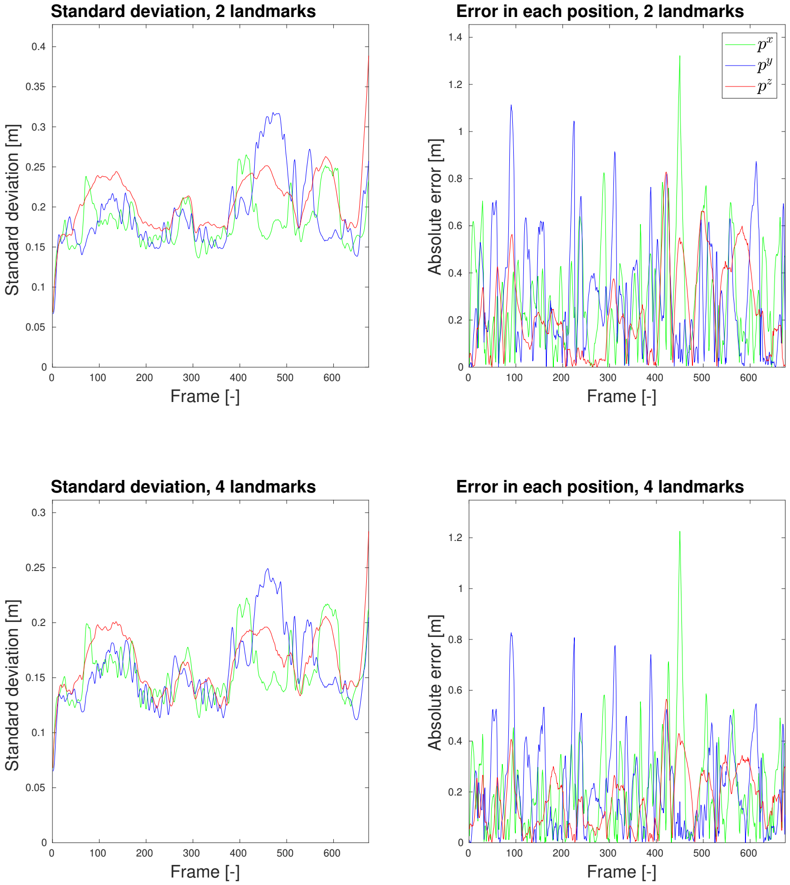

Figure B.9: *Absolute errors and the standard deviations of the estimates at 1 Hz during the highway track with 2 and 4 landmarks visible in each image.*

B.4.2 Urban track

In Figure B.10 the result for the urban track can be found while using 2 or 4 landmarks in each image.

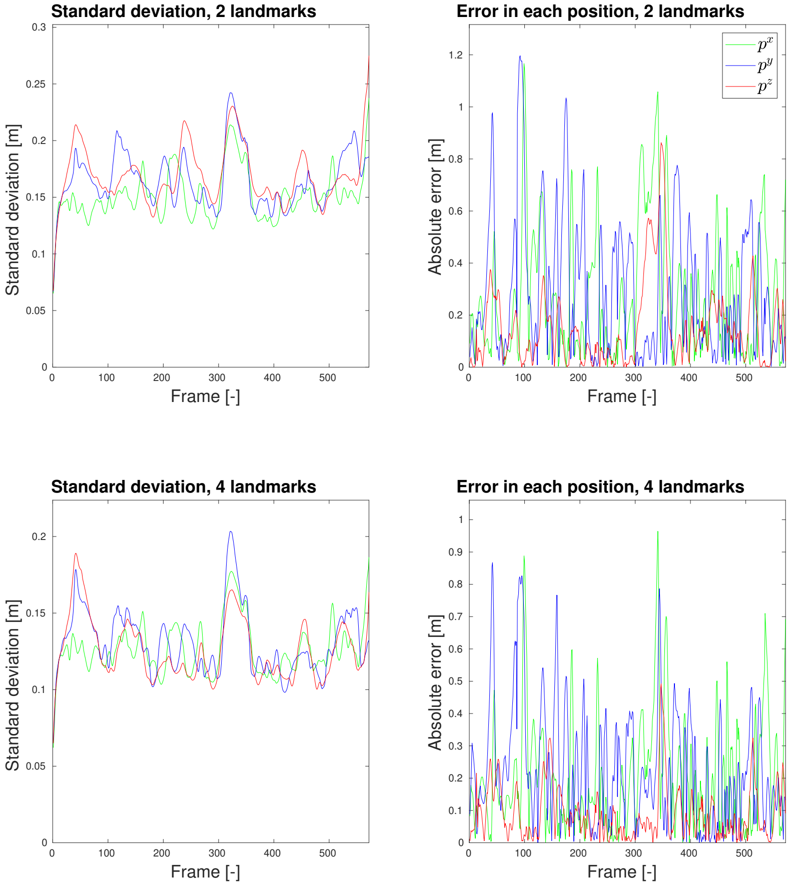

Figure B.10: *Absolute errors and the standard deviations of the estimates at 1 Hz during the urban track with 2 and 4 landmarks visible in each image.*

B.5 Sparse landmark selection

For these tests, a landmark selection as described in 4.5.2 was used.

B.5.1 Highway track

For this trajectory, a distant landmark was placed 5 km east, 1 km south, and 100 meters above the vehicle's starting position. In Figure B.11 the result where only one distant visible landmark per image can be found, and where two complementary landmarks are added along the track.

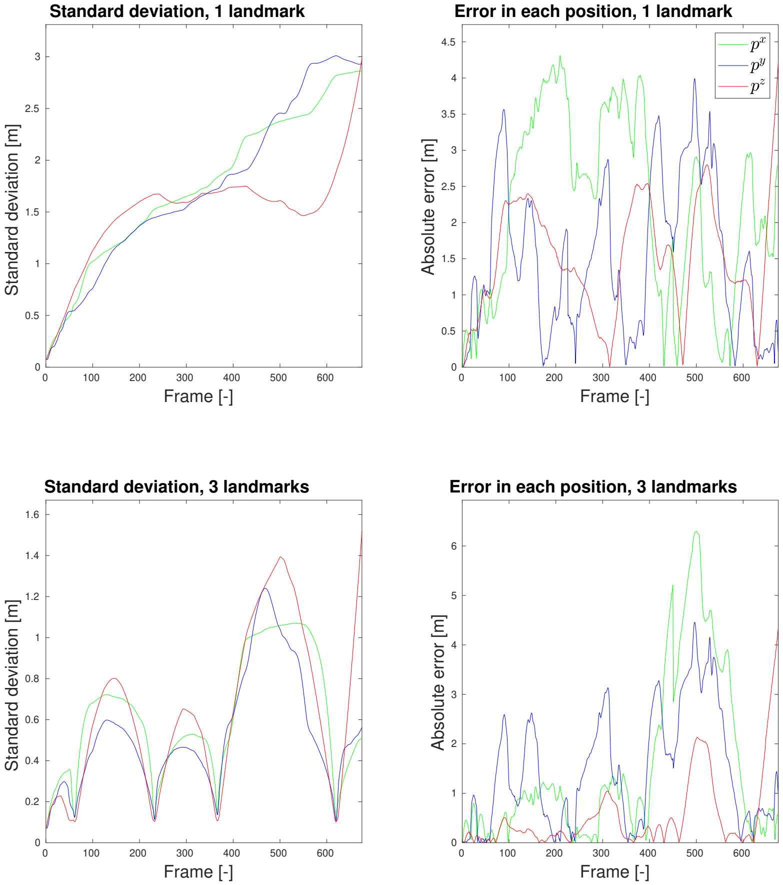

Figure B.11: *Absolute errors and the standard deviations of the estimates at 1 Hz during the highway track with a distant landmark and 2 complementary landmarks.*

The estimated path with the complementary landmarks can be seen in Figure B.12. The values of MSE and ANEES for the two tests can be seen in Table B.10.

Table B.10: *Values of MSE and ANEES for the two landmark setups.*

	MSE [m]	ANEES [-]
Distant landmark	12.8197	7.3256
3 landmarks	10.0988	18.4555

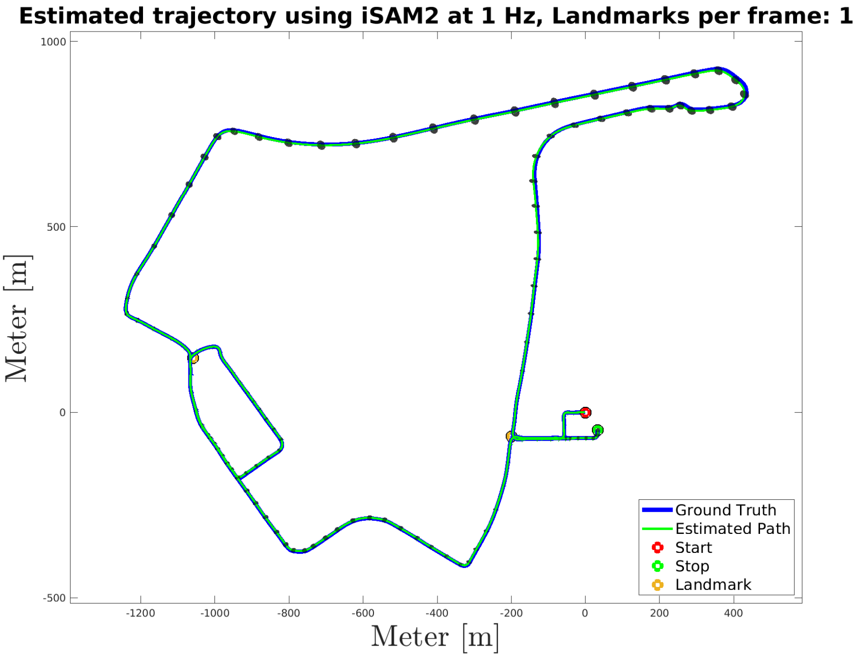

Figure B.12: Estimated path with uncertainties using a distant landmark and two complementary landmarks.

B.5.2 Urban track

For this trajectory, a distant landmark was placed 200 m west, 5 km north, and 100 meters above the vehicle's starting position. In Figure B.13 the result where only one distant visible landmark per image can be found, and where two complementary landmarks are added along the track. The estimated path can be seen in B.14. The values of MSE and ANEES for the two tests can be seen in Table B.11.

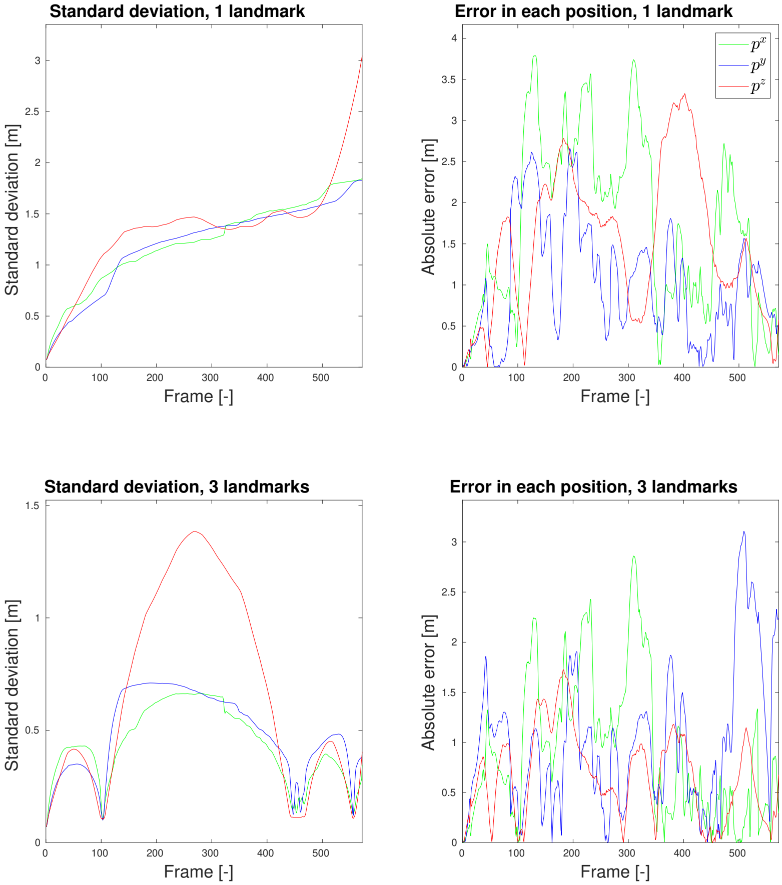

Figure B.13: *Absolute errors and the standard deviations of the estimates at 1 Hz during the urban track with a distant landmark and 2 complementary landmarks.*

Table B.11: *Values of MSE and ANEES for the two landmark setups.*

	MSE [m]	ANEES [-]
Distant landmark	8.7762	6.2901
3 landmarks	3.4817	15.4020

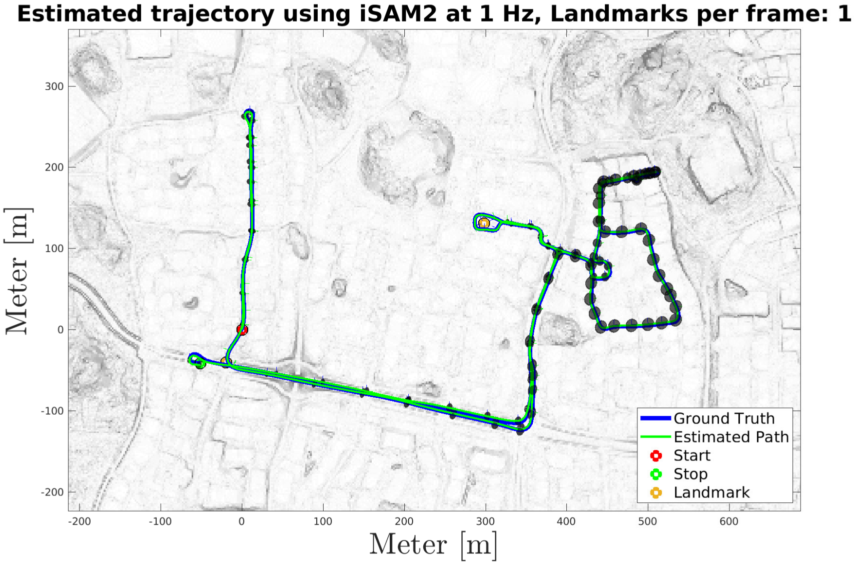

Figure B.14: *Estimated path with uncertainties using a distant landmark and two complementary landmarks.*

B.6 Realistic landmark selection

For this part, the landmarks were selected according to Section 4.5.2 from the real images. The values for MSE and ANEES can be found in Table B.12.

***Table B.12:** MSE and ANEES of the estimated country roads track at realistic landmark setups.*

	MSE [m]	ANEES [-]
Highway track	0.6254	9.8007
Urban track 3	0.40266	16.342

B.6.1 Highway track

In Figure B.15 the estimated path for the highway track can be seen with the chosen landmarks. The absolute errors and standard deviations can be seen in Figure B.16.

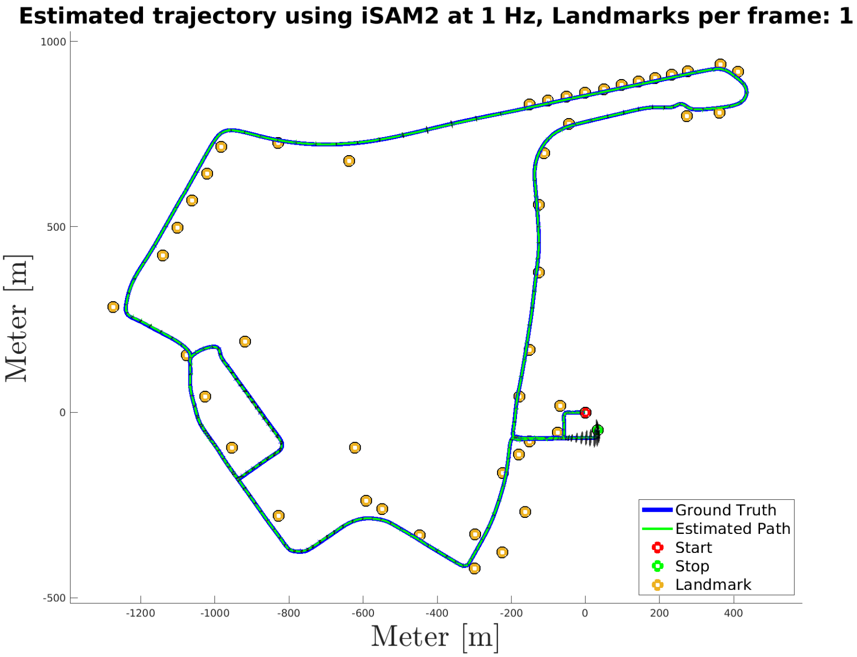

Figure B.15: *The estimated path with realistic landmarks for the highway track.*

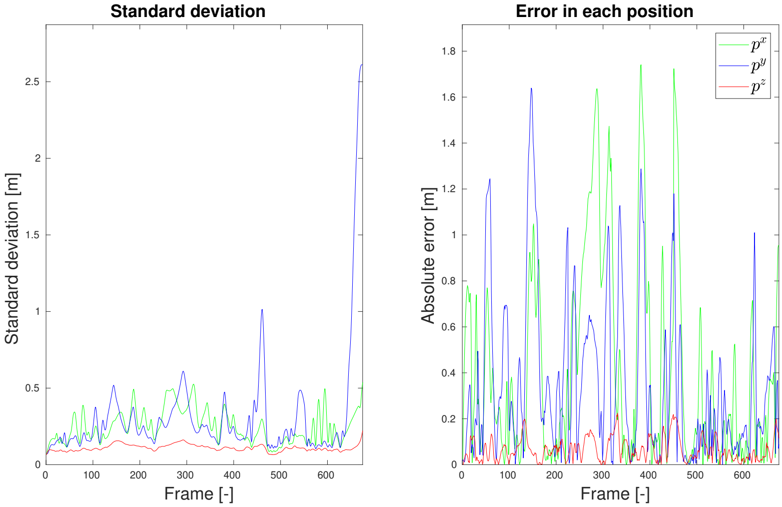

Figure B.16: *Absolute errors and the standard deviations of the estimates with realistic landmarks for the highway track.*

B.6.2 Urban track

For the urban track, the estimated path and its landmarks can be seen in Figure B.17 while the absolute errors and standard deviations can be seen in Figure B.18.

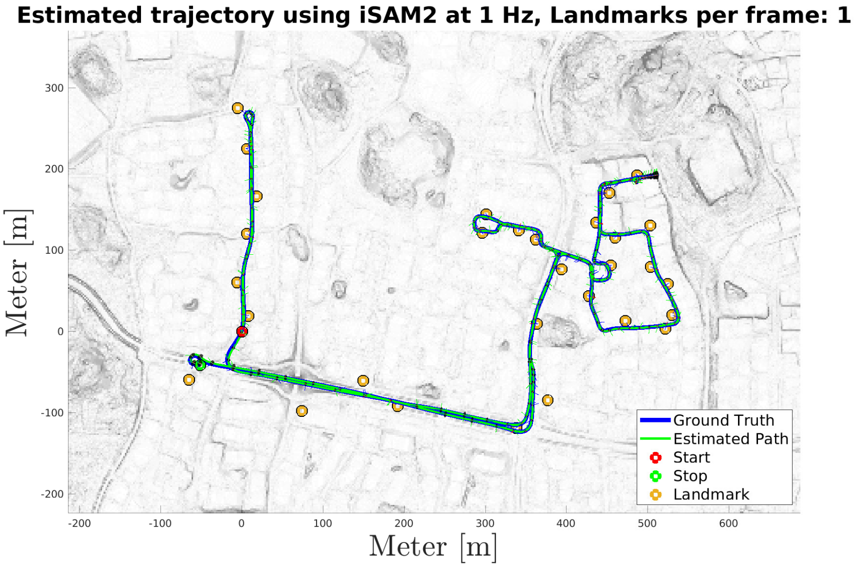

Figure B.17: *The estimated path with realistic landmarks for the urban track.*

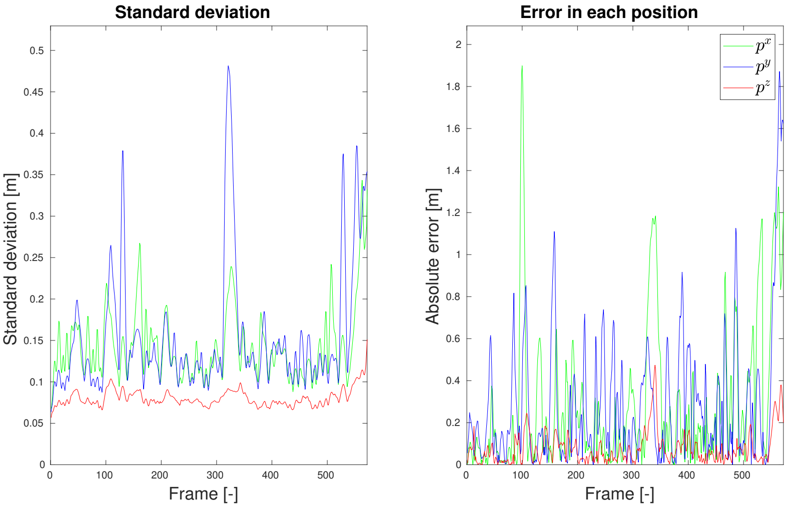

Figure B.18: *Absolute errors and the standard deviations of the estimates with realistic landmarks for the urban track.*

Bibliography

[1] J. Zhao, "A Review of Wearable IMU (Inertial-Measurement-Unit)-based Pose Estimation and Drift Reduction Technologies," *Journal of Physics: Conference Series*, vol. 1087, Sep 2018.

[2] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *Vision Algorithms: Theory and Practice*, (Berlin, Heidelberg), pp. 298–372, Springer Berlin Heidelberg, 2000.

[3] K. Wilson and N. Snavely, "Robust Global Translations with 1DSfM," *Lecture notes in Computer Science (LNCS)*, vol. 8691, pp. 61–75, 2014.

[4] S. Albrecht, "An Analysis of Visual Mono-SLAM," Master's thesis, Osnabrück University, 2009.

[5] Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proceedings Ninth IEEE International Conference on Computer Vision*, vol. 2, pp. 1403–1410, 2003.

[6] D. Yuen and B. MacDonald, "Vision-based localization algorithm based on landmark matching, triangulation, reconstruction, and comparison," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 217–226, 2005.

[7] M. Servieres, V. Renaudin, A. Dupuis, and N. Antigny, "Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking," *Journal of Sensors*, vol. 2021, p. 26, Jan 2021.

[8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[9] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," in *Eighteenth National Conference on Artificial Intelligence*, (USA), p. 593–598, American Association for Artificial Intelligence, 2002.

[10] A. I. Mourikis and S. I. Roumeliotis, "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3565–3572, 2007.

[11] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 225–234, 2007.

[12] H.-P. Chiu, M. Sizintsev, X. S. Zhou, P. Miller, S. Samarasekera, and R. Kumar, "Sub-meter vehicle navigation using efficient pre-mapped visual landmarks," *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 505–512, 2016.

[13] X. Qu, B. Soheilian, and N. Paparoditis, "Vehicle localization using monocamera and geo-referenced traffic signs," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 605–610, 2015.

[14] C. Beall and F. Dellaert, "Appearance-based localization across seasons in a metric map," in *6th Workshop on Planning, Perception and Navigation for Intelligent Vehicles*, 2014.

[15] C. Tang, O. Wang, and P. Tan, "GSLAM: Initialization-Robust Monocular Visual SLAM via Global Structure-from-Motion," in *2017 International Conference on 3D Vision (3DV)*, pp. 155–164, 2017.

[16] C. Hui and M. Shiwei, "Visual SLAM based on EKF filtering algorithm from omnidirectional camera," in *2013 IEEE 11th International Conference on Electronic Measurement Instruments*, vol. 2, pp. 660–663, 2013.

[17] A. Díaz, E. Caicedo, L. Paz, and P. Piniés, "A Real Time 6DOF Visual SLAM System Using a Monocular Camera," in *2012 Brazilian Robotics Symposium and Latin American Robotics Symposium*, pp. 45–50, 2012.

[18] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE International Conference on Robotics and Automation*, pp. 3281–3288, 2011.

[19] D. Knuth, "SfM," 2022. `https://gtsam.org/doxygen/a01594.html`, accessed: 14/02/2022.

[20] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing," *I. J. Robotic Res.*, vol. 25, pp. 1181–1203, 12 2006.

[21] S. Lange, N. Sünderhauf, and P. Protzel, "Incremental smoothing vs. filtering for sensor fusion on an indoor UAV," in *2013 IEEE International Conference on Robotics and Automation*, pp. 1773–1778, 2013.

[22] F. Daellert, "GTSAM," 2022. `https://gtsam.org/`, accessed: 16/02/2022.

[23] S. Särkkä, *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks, Cambridge University Press, 2013.

[24] N. Bergman, *Recursive Bayesian Estimation : Navigation and Tracking Applications*. PhD thesis, Linkoping University, 1999.

[25] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[26] M. Kaess and F. Dellaert, "Covariance recovery from a square root information matrix for data association," *Robotics and Autonomous Systems*, vol. 57, no. 12, pp. 1198–1210, 2009.

[27] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," *IEEE Transactions on Robotics*, vol. PP, 08 2016.

[28] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*. USA: CRC Press, Inc., 1st ed., 1994.

[29] R. Nylén and K. Rajala, "Development of an ICP-based Global Localization System," Master's thesis, Linköping University, 2021.

[30] Lantmateriet, "Kartor och karttjänster," 2018. `https://www.linkoping.se/bygga-bo-och-miljo/fastigheter-och-lantmateri/kartor-och-karttjanster`, accessed: 11/09/2022.

[31] E. Blasch, A. Rice, C. Yang, and I. Kadar, "Relative track metrics to determine model mismatch," in *2008 IEEE National Aerospace and Electronics Conference*, pp. 257–264, 2008.