Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Trym Andreassen Synnevåg

Sensor fusion of phased array radio and inertial sensor measurements using factor-graph-based optimisation

Master's thesis in Industrial Cybernetics
Supervisor: Torleiv Håland Bryne
June 2023

NTNU
Norwegian University of
Science and Technology

Trym Andreassen Synnevåg

Sensor fusion of phased array radio and inertial sensor measurements using factor-graph-based optimisation

Master's thesis in Industrial Cybernetics
Supervisor: Torleiv Håland Bryne
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology
and Electrical Engineering
Department of Engineering Cybernetics

MASTER'S THESIS DESCRIPTION SHEET

Name: Trym Andreassen Synnevåg
Department: Engineering Cybernetics
Thesis title (Norwegian): Sensor fusjon av fase-basert radio og treghetsmålinger ved bruk av faktorgraf basert optimalisering.
Thesis title (English): Sensor fusion of phased array radio and inertial sensor measurements using factor-graph-based optimization.

Thesis Description: Global navigation satellite systems (GNSS) is the primary outdoor positioning sensor for robotic vehicles such as unmanned arial vehicles (UAVs), unmanned surface vehicles (USVs) and unmanned ground vehicles (UGVs). However, GNSS is susceptible to interference, natural and intentional, due to its low signal power. Alternative radio positioning sources have been investigated. Phased array radio systems (PARS) have been shown to be an option for aircraft navigation when PARS measurements have been filtered using inertial measurement units (IMUs) and nonlinear observers (NLO) or error state Kalman filters (EKF). Factor graph-based optimization (FGO) is taken over from ESKF as the filtering tool in simultaneously localization and mapping (SLAM). The focus of this theis is to investigate how FGO can be utilized in aided inertial navigation using both GNSS and PARS measurements for UAV navigation.

The following tasks should be considered:

1. Perform a short literature review on
    a. Necessary background theory on Bayesian estimation.
    b. GNSS, IMU, and PARS sensor and sensor errors in context of navigation
    c. traditional filtering methods for integrating IMU and PARS measurements.
    d. Factor graph-based optimization methods in context for sensor fusion of IMU and aiding sensors.
    e. on potential factor graph frameworks for sensor fusion for aided inertial navigation (INS)
2. Set up a robot operating systems framework (ROS) for data handling sensor data flow of GNSS, IMU and PARS data collected field data.
3. Implement one or more FGO-based filters for aided inertial navigation in C++ based on your literature review using a chosen framework. Justify your choice of FGO method and framework.
4. Fuse both GNSS and PARS measurement with IMU using FGO. Present and discuss your results using sensor data collected during field trials.
5. Present and implement a standard GNSS-aided INS using the ESKF for comparison.
6. Present and discuss your results. Discuss potential tradeoffs w.r.t FGO real-time filtering applicability versus estimation accuracy/performance.
7. Conclude on your results and suggest further work.

Start date: 2023-01-12
Due date: 2023-06-08
Thesis performed at: Department of Engineering Cybernetics, NTNU
Supervisor: Associate professor Torleiv H. Bryne,
Dept. of Eng. Cybernetics, NTNU

Preface

This master's thesis is a product of the course Engineering Cybernetics Master's Thesis (TTK4900), accounting for 30 credits in the 2-year Master's Degree Programme Industrial Cybernetics at the Norwegian University of Science and Technology. It is written at the Department of Engineering Cybernetics in the spring of 2023 and concludes my time as a student in Trondheim.

The outcome of this master's thesis is an accumulation of knowledge obtained over 6 years as a student. The chosen journey has been challenging, but in the end, I would not have it any other way. After 40 examinations, 1 bachelor's thesis and 1 master's thesis, I can proudly step into the next chapter of my life.

Acknowledgment

First of all, I would like to sincerely thank my supervisor Torleiv Håland Bryne. Throughout the spring semester, we have had weekly meetings, and his advice, guidance and motivation have been key factors for the outcome of this thesis. Torleiv assisted with theoretical aspects, relevant literature and continuous feedback until the end, being a great resource. I would also like to thank my good friend Jack Frain, my sister Ida Marie Synnevåg and Aleksander Nysted Elvebakk for assisting me in proofreading the thesis.

I also want to express gratitude towards all the amazing people I have had the pleasure of getting to know during my time as a student. To my great friends from the ISFiT Welfare board, Fuel Fighter Autonomous, Extra Leuthenhaven, The Originals in Australia, The Adams and Linde Allé in Copenhagen, Omega and Vivas. Thank you for 6 memorable years. It would never be the same without you.

Lastly, I would like to thank my family for their motivation and undivided support during my years as a student. I would never be where I am today without them.

Abstract

Adopting Unmanned Aerial Vehicles (UAVs) offers numerous advantages for commercial and military applications, with its high precision and cost-efficiency. The UAV laboratory at NTNU has conducted extensive research on the navigation of UAVs. The de facto method has been to employ Global Navigation Satellite System (GNSS)-aided Inertial Navigation System (INS) with the INS corrections implemented with a Kalman Filter (KF) based estimation algorithm, specifically the Multiplicative Error State Kalman Filter (MEKF). However, due to the vulnerabilities of GNSS, it is necessary with an alternative backup solution for safety-critical UAV operations. The UAV lab has found a Phased Array Radio System (PARS) a viable alternative, though there are challenges with multipath propagation. Moreover, the KF algorithm is sensitive to outliers, linearisation and non-gaussian sensor errors, and, therefore, it is interesting to explore alternative solutions potentially providing better robustness and higher precision in its estimates. A highly promising estimation formulation is Factor Graph Optimisation (FGO), a technique applied to problems such as Simultaneous Localisation and Mapping.

This thesis investigates to what degree an FGO implementation can improve the performance of UAV state estimation and increase robustness to noisy non-Gaussian measurement noise, such as PARS. This is explored by comparing the FGO algorithm against the MEKF. The MEKF is implemented in MATLAB using a toolbox from the UAV lab, while the FGO is realised in C++ using the Georgia Tech Smoothing and Mapping (GTSAM) framework. The algorithms are tested on a dataset provided by the UAV lab of a flight at Agdenes Airfield northwest of Trondheim. It provides Inertial Measurement Unit (IMU) measurements, accurate GNSS measurements and erroneous PARS data because of multipath propagation. Furthermore, a literature review was conducted on the two algorithms and their performance, focusing on GNSS and PARS measurements. This was consequently used to motivate FGO framework choice.

The FGO estimation scheme implemented in this thesis successfully estimates the position and attitude with GNSS-aided INS. Additionally, it achieves satisfactory estimates of the North and East position along with the attitude with PARS-aided INS. The FGO and MEKF produce highly similar position and attitude estimates. This implies that both find a similar optimal solution to their optimisation problem. However, both produce an inadequate estimation of the Down position with PARS-aided INS because of the high amount of multipath in elevation. The thesis finds that the traditional least square formulation of the FGO is a competitive alternative to the MEKF used by the UAV lab today. Furthermore, it has promising possibilities to improve robustness towards outliers, making it more flexible than the MEKF.

These thesis findings motivate further research in the field of UAV state estimation with FGO techniques. Specifically, the FGO potential to improve the robustness towards the noisy PARS measurements is advantageous to investigate and necessary to provide accurate estimates with PARS measurements.

Sammendrag

Å ta i bruk ubemannede luftfartøyer (UAV) gir en rekke fordeler for kommersielle og militære applikasjoner, med sin høye presisjon og kostnadseffektivitet. UAV-laboratoriet ved NTNU har drevet omfattende forskning på navigering av UAV. De facto-metoden har vært å bruke globalt navigasjons satellitt system (GNSS)-støttet treghetsnavigasjonssystem (INS) med INS-korreksjonene implementert med en Kalman filter (KF) basert estimeringsalgoritme, spesielt multiplikativ feiltilstand Kalmanfilter (MEKF). På grunn av sårbarhetene til GNSS er det imidlertid nødvendig med en alternativ backupløsning for sikkerhetskritiske UAV-operasjoner. UAV-laboratoriet har funnet et fasestyrt radiosystem (PARS) som et godt alternativ, selv om det er utfordringer med flerveisinterferens. Dessuten er KF-algoritmen følsom for avvikende målinger, linearisering og ikke-gaussiske sensorfeil, og derfor er det interessant å utforske alternative løsninger som potensielt gir bedre robusthet og høyere presisjon i estimatene. En svært lovende estimeringsformulering er factor graf optimalisering (FGO), en teknikk som brukes på problemer som samtidig lokalisering og kartlegging (SLAM).

Denne oppgaven undersøker i hvilken grad en FGO-implementering kan forbedre ytelsen til UAV-tilstandsestimat og øke robustheten mot støyende ikke-Gaussisk målestøy, slik som PARS. Dette utforskes ved å sammenligne FGO-algoritmen med MEKF. MEKF er implementert i MATLAB ved hjelp av en verktøykasse fra UAV-laben, mens FGO er realisert i C++ ved hjelp av Georgia Tech Smoothing and Mapping (GTSAM) rammeverket. Algoritmene er testet på et datasett levert av UAV-laben til en flyging på Agdenes flyplass nordvest for Trondheim. Den gir Treghetsmåleenhet (IMU)-målinger, nøyaktige GNSS-målinger og feilaktige PARS-data på grunn av flerveisinterferens. Videre ble det utført en litteraturgjennomgang på de to algoritmene og deres ytelse, med fokus på GNSS- og PARS-målinger. Dette ble følgelig brukt for å motivere FGO-rammevalg.

FGO-estimerings algoritmen implementert i denne oppgaven produserer velykket estimat av posisjonen og orientering med GNSS-støttet INS. I tillegg oppnår den tilfredsstillende estimater for nord- og østposisjonen sammen med orientering til PARS-støttet INS. FGO og MEKF produserer svært like poition- og orienteringsestimater. Dette innebærer at begge finner en lignende optimal løsning på optimaliseringsproblemet. Begge produserer imidlertid en utilstrekkelig estimering av ned-posisjonen med PARS-støttet INS på grunn av den høye mengden flerveisinterferens i høyden. Avhandlingen finner at den tradisjonelle minste kvadratiske formuleringen av FGO er et konkurransedyktig alternativ til MEKF som brukes av UAV-laboratoriet i dag. Videre har den lovende muligheter for å forbedre robustheten mot avikkende data, noe som gjør den mer fleksibel enn MEKF.

Disse funnene i oppgaven motiverer til videre forskning innen UAV-statestimering med FGO-teknikker. Spesifikt er FGO-potensialet for å forbedre robustheten mot de støyende PARS-målingene fordelaktig

å undersøke og nødvendig for å gi nøyaktige estimater med PARS-målinger.

Contents

Figures

Tables

Acronyms

AHRS Attitude and Heading Reference System. xvi, 48, 55, 58, 60, 64, 67, 90, 91, 93

ARS Attitude Rate Sensor. 23, 24

CDF Cumulative Distribution Function. 14, 15

ECEF Earth-Centred, Earth-Fixed. xv, 6, 7, 27, 52, 70, 74, 85

ECI Earth-Centred Inertial. xv, 6, 7, 26

EKF Extended Kalman filter. 31–33, 43, 44, 70, 73, 81, 82

ESKF Error State Kalman Filter. 31, 33, 43, 44, 70, 81

FG Factor Graph. xvii, 19, 20, 31, 36, 40, 41, 44, 48, 54, 55, 82

FGO Factor Graph Optimisation. vii, viii, xvi, xvii, 1–3, 31, 40, 41, 43, 44, 47, 48, 54–71, 73–75, 82, 89–93

GNSS Global Navigation Satellite System. vii, xv–xvii, 1, 3, 5, 27, 28, 32, 36, 39, 43–45, 47–49, 51, 52, 54–67, 69, 73, 74, 81–83, 89–93

GPS Global Positioning System. 1, 27, 45, 82

GTSAM Georgia Tech Smoothing and Mapping. vii, 44, 48, 54, 56, 57, 68, 70, 71, 73, 74

IMU Inertial Measurement Unit. vii, 5, 7, 23–25, 34, 36, 37, 39, 44, 45, 48, 52, 54, 55, 58, 68, 70, 73, 82, 83, 85

INS Inertial Navigation System. vii, xvii, 1–3, 5, 25, 26, 35, 43–45, 48, 57, 58, 61, 62, 65, 69, 71, 73, 81–83, 89, 91

KF Kalman Filter. vii, 2, 3, 31–33, 36, 40, 43–45, 73, 81, 83

LS Least Squares. 22, 23, 40, 68, 71, 73

LSO Least Squares Optimisation. 21, 36, 38, 40

MAE Mean Absolute Error. 21, 57, 58, 62, 65, 69

MAP Maximum A Posteriori. xv, 17, 31, 36, 39, 82

MATLAB MAtrix LABratory. 48, 52, 54

ME Mean Error. 21, 57, 58, 62, 65, 69

MEKF Multiplicative Error State Kalman Filter. vii, xvi, xvii, 1–3, 31, 33, 35, 36, 40, 41, 43, 45, 47, 48, 52, 55–67, 70, 71, 73, 74, 83, 89–93

MRP Modified Rodriguez Parameters. 33, 43, 55

NED North, East, Down. xv, 6–8, 26, 27, 45, 49, 52, 54, 70, 74, 85

NIS Normalised Innovation Squared. 36

NTNU Norwegian University of Science and Technology. vii, xvii, 2, 45, 48, 83, 87

PARS Phased Array Radio System. vii, viii, xv–xvii, 1–3, 5, 8, 28–30, 32, 43, 45, 47–49, 51, 52, 54, 56, 57, 61–69, 71, 73–75, 83, 85, 89, 91–93

PDF Probability Density Function. 14–16, 18

PPK Post-Processed Kinematics. xv–xvii, 27, 28, 48, 49, 51, 55, 58, 59, 61–67, 69, 89–93

RMSE Root Mean Square Error. 21, 45, 57, 58, 62, 65, 69

ROS Robot Operating System. 48, 54, 71, 73

RTK Real-Time Kinematics. 27, 28, 45

SLAM Simultaneous Localisation and Mapping. vii, 2, 44

STD Standard Deviation. 21, 57, 58, 62, 65, 69

UAV Unmanned Aerial Vehicle. vii, viii, xv, 1–3, 7, 8, 14, 27, 30, 48, 50, 57, 62, 70, 74, 83

UAV lab UAV laboratory. vii, xvii, 2, 3, 45, 47, 48, 55, 56, 73, 83, 87

Chapter 1

Introduction

This thesis considers state estimation of Unmanned Aerial Vehicles (UAVs) with the primary objective of enhancing the performance of autonomous navigation. This entails exploring the promising estimation scheme, Factor Graph Optimisation (FGO), to potentially increase accuracy and robustness compared to the current state-of-the-art algorithm, Multiplicative Error State Kalman Filter (MEKF). This chapter provides an introduction to the problem at hand, starting with a background on the field of state estimation and UAVs. This is followed by the motivation for writing this thesis and its main contributions. Lastly, the structure of this thesis is presented, providing an understanding of how the thesis is built to answer the problem to be addressed.

1.1 Background

Rapid technological advancements have led to robust and cost-effective solutions for various industries in the last few decades. One such advancement is the civilian UAV, also known as drones. These highly autonomous systems navigate their environment without human input or assistance, [1]. However, for a UAV to operate robustly and safely, it must determine its position and orientation relative to the environment and make informed decisions on its movements and destination. This process is commonly referred to as navigation while computing its location is communicated as state estimation. This is visualised in Figure 1.1, [2, p. 7, p. 311].

State estimation involves complicated mathematical modelling, which utilises measurements from various sensors. The most prominent sensor for localising seagoing, land and aerial vehicles is the Global Navigation Satellite System (GNSS), which is an umbrella term for satellite navigation systems, such as Global Positioning System (GPS). The GNSS provides worldwide positioning coverage calculated from a continuously transmitted signal on the UAV from at least four satellites. However, GNSS is susceptible to natural and intentional interference due to its low signal power [3]. Therefore, alternative local positioning systems, such as the Phased Array Radio System (PARS), are introduced as a substitute solution. GNSS and PARS provide measurements at a low frequency of 1-5 Hz, and it is advantageous to combine them with the high-frequency 50-200 Hz position system called Inertial Navigation System (INS) that provides accurate localisation between the GNSS or PARS measure-

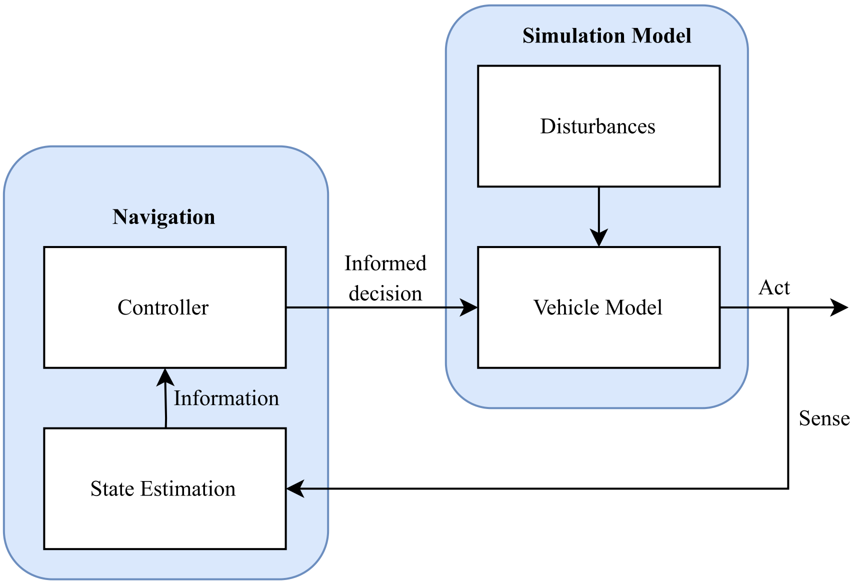

Figure 1.1: Block diagram of a classical control system. Split into the navigation system and simulation system. Inspired by [2, p. 7, p. 311].

ments. However, INS drift over time because of noisy and erroneous inertial sensor measurements, [2, Ch. 13, Ch. 14].

Various state estimation schemes exist, and the most influential estimation model is the Kalman Filter (KF), which dates back to the 1960s when it was applied by NASA [4]. Since then, many variants of the KF have risen and are highly competitive, such as the MEKF. Furthermore, in the last two decades, other state estimation methods have emerged through solving challenging problems, such as Simultaneous Localisation and Mapping (SLAM). One such estimation technique is FGO, [5, 6].

1.2 Motivation

UAVs offer numerous advantages, such as manoeuvring in areas dangerous for humans or where a human-crewed aircraft is impractical or unnecessary. Furthermore, they are cost- and time-effective and acquire high precision, which is valuable for commercial and military applications, [1]. Therefore, the UAV laboratory (UAV lab) at the Norwegian University of Science and Technology (NTNU) has been researching the development of UAV technologies for several years. Much research has gone into applying PARS as an alternative local localisation sensor, combined with the MEKF estimation method. The UAV lab aims to continue the research on UAV state estimation to increase the level of autonomy, which in turn will incentivise and simplify the process of producing civil and commercial applications.

1.3 Main Contributions

This thesis aims to build on the UAV lab research and investigate state-of-the-art estimation techniques by comparing the powerful MEKF against the promising FGO. The main focus will be on examining their accuracy with the GNSS-aided INS and the PARS-aided INS, in addition to their robustness when handling outliers in the measurement data. The problem to be addressed by this thesis is formulated as follows:

"***To what degree can Factor Graph Optimisation improve performance of UAV state estimation and increase robustness to noisy measurements, such as Phased Array Radio System?***"

This thesis is intended to increase the knowledge of using FGO as a state estimation technique on UAVs. Also, the outcomes contribute to the work by the UAV lab and the field of state estimation of UAVs in general.

1.4 Structure of Thesis

To answer the problem to be addressed, it is necessary to introduce several technical aspects. These are stated compactly in Chapters 2 to 4, while the methodology, results, discussion and finally, conclusion are presented in Chapters 5 to 7.

Chapter 2 provides all the necessary prerequisites for understanding this thesis, which involves mathematical, statistical and navigation-specific theory. Chapter 3 dives into the specific KF and FGO estimation techniques. Further, Chapter 4 gives insight into related work on KF and FGO estimation, focusing on GNSS and PARS systems.

In Chapter 5, the thesis methodology is stated with insight into the process and used resources. Additionally, the experimental setup with the associated dataset is explained in detail before the practical implementation of the estimation schemes is stated. The thesis results and discussion are provided in Chapter 6, and the problem to be addressed is finally concluded in Chapter 7.

Chapter 2

Prerequisites

This chapter presents relevant background prerequisites required to understand this thesis. This involves mathematical principles in Sections 2.1, 2.4 and 2.6, statistical concepts in Section 2.5, and navigational-specific theory in Sections 2.2, 2.3 and 2.7 to 2.10. The concepts introduced in this chapter are interconnected and built upon throughout the thesis, aiming to establish a cohesive flow with a clear progression in the topics.

Firstly, the principles of Euclidean space and the basics of coordinate systems are specified, which reference frames and pose representation extend upon. This will show to be fundamental in navigation and state estimation. Next, Lie theory is introduced as the pose can be defined as a Lie group, which in turn will manifest itself into statistical and optimisation theory. Lastly, numerous navigational-specific concepts such as Inertial Measurement Unit (IMU), INS, GNSS and PARS are explained.

2.1 Euclidean space and coordinates systems

Euclidean space is a mathematical concept used to describe the geometry of physical space and is one of many alternative methods to describe physical space. The Euclidean space is a finite n-dimensional space noted as $\mathbb{R}^n$. For example, the 3D Euclidean space, denoted $\mathbb{R}^3$, has three independent planes; length, width and height, [7, Ch. 2].

The Cartesian coordinate system is a tool used to describe Euclidean space where, in $\mathbb{R}^3$, any point in 3D space is defined by three coordinates (x, y, z). It is a set of orthogonal axes intersecting at one point in space called origo, denoted o. Alternative coordinate systems, such as the spherical system, describe any point in Euclidean space by the azimuth and elevation angles, ξ and α and a distance, ρ. It is possible to convert between the Cartesian and spherical coordinate systems using

$$x = \rho \cos(\xi) \cos(\alpha) \tag{2.1a}$$

$$y = \rho \sin(\xi) \cos(\alpha) \tag{2.1b}$$

$$z = -r \sin(\alpha) \tag{2.1c}$$

Both representations are depicted in Figure 2.1 along with a point $\boldsymbol{x}$ defined as $\boldsymbol{x} = (x, y, z)$ in Cartesian coordinates or $\boldsymbol{x} = (\rho, \xi, \alpha)$ in Spherical coordinates, [8]. Additionally, a Cartesian co-

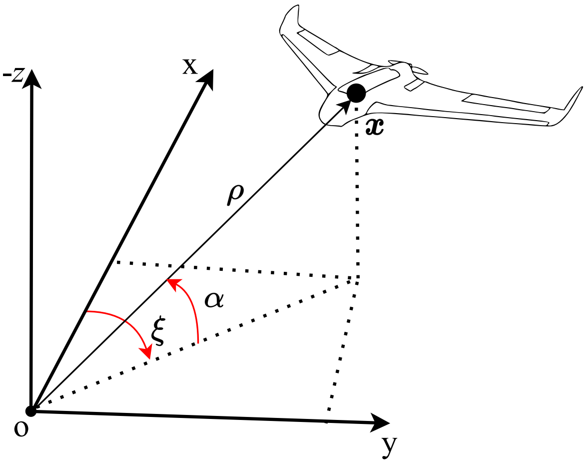

Figure 2.1: Cartesian and sperical coordinate system in a 3D Euclidean space. Describing a point $\boldsymbol{x}$ in Euclidean space with spherical coordinates.

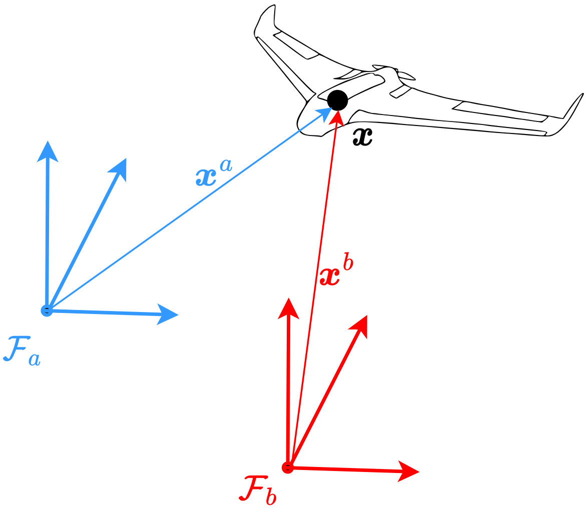

Figure 2.2: Point $\boldsymbol{x}$ relative to two different cartesian frames $\mathcal{F}_a$ and $\mathcal{F}_b$.

ordinate vector in $\mathbb{R}^3$ relative to a specific Cartesian frame, denoted $\mathcal{F}_a$, is noted as $\boldsymbol{x}^a$ and visualized in Figure 2.2. Describing a point in space with multiple Cartesian frames is possible and convenient, as in Figure 2.2. This thesis identifies vectors by bold lowercase and uppercase letters, e.g. $\boldsymbol{x}$ and $\boldsymbol{X}$, and matrices as bold uppercase letters. E.g. $\boldsymbol{R}$. For bold uppercase letters, it is specified throughout the thesis whether it is a vector or matrix. Their respective transposes are noted as $\boldsymbol{x}^T$ and $\boldsymbol{R}^T$, [8].

2.2 Reference Frames

Reference frames are specific Cartesian frames and are fundamental to describing an object's position and movement. Several standard reference frames exist, which are used depending on the application. For example, in Figure 2.3, two Earth-centred references are displayed. These frames describe an object relative to the Earth's centre. Earth-Centred Inertial (ECI), denoted {i}, is an inertial reference frame, i.e. its orientation does not change. The Earth-Centred, Earth-Fixed (ECEF) denoted {e}, reference frame has its origin in the Earth centre and rotates with the Earth relative to the ECI frame, [2, Ch. 2].

For local navigation, tangent planes near the Earth's surface are used, and one such frame is the North, East, Down (NED) denoted {n}, coordinate system. This frame is also depicted in Figure 2.3, where its x-axis points North, the y-axis points east and the z-axis down into the Earth's centre, [2, Ch. 2].

Any position on the Earth's surface can be described relative to the ECI or ECEF frame. The notation $\boldsymbol{p}_{en}^e$ is a vector which describes the position of the NED frame origin relative to the ECEF frame stated in the ECEF frame, [2, Ch. 2]. Additionally, any point on and above the Earth's surface can also be described by the geographic coordinate system, similar to a spherical coordinate system. This

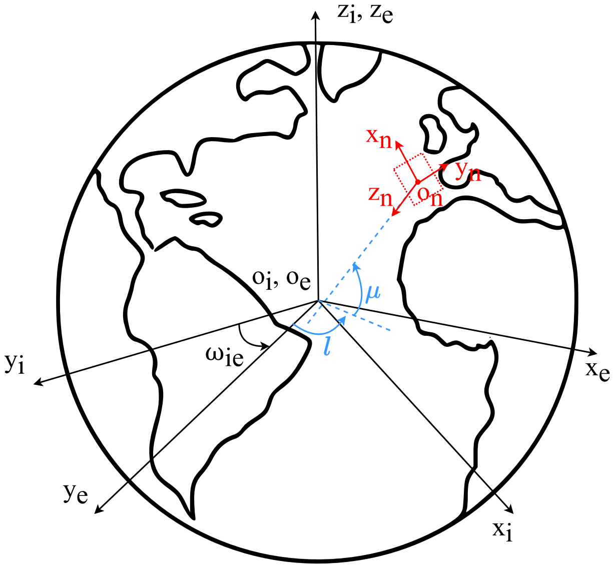

Figure 2.3: Three reference frames; ECI, ECEF, NED and the geographic coordinate system. The location of the NED frame is described by the geographic coordinates latitude, μ, and longitude, l. The ECEF describe relative to the ECI with the angular velocity ω_{ie}.

coordinate system can describe a position with two angles, latitude, μ, longitude, l and distance, which are relative to the axes of a local navigation frame. However, since the Earth is an ellipse, this coordinate system provides an ellipsoidal position. In addition, the orientation of the axis of the local navigation frame varies with the location on Earth. This is illustrated in Figure 2.3, precisely the blue elements. More details regarding the geographic coordinate system are found in [9, Ch. 2].

Objects also have a reference frame called the body-fixed reference frame, denoted {b}. This is a moving coordinate frame fixed to a specific point on the object. For a fixed-wing UAV, depicted in Figure 2.4, a body frame's x-axis points in the forward direction, and the z-axis points down perpendicular to the plane. The y-axis completes the right-hand frame and points to the right. Other reference frames are also defined for simplicity, such as a frame for an IMU, denoted {m}, and a frame for a radio antenna, denoted {r}, depicted in Figure 2.4 and Figure 2.5, respectively, [2, Ch. 2].

These definitions simplify describing a UAV's position, $\boldsymbol{p}$, relative to specific locations. For example, the drone can be described relative to the radio antenna, $\boldsymbol{p}^r_{rb}$, with $\boldsymbol{p}$ decomposed in $\{r\}$, or relative to the NED frame, $\boldsymbol{p}^n_{nb}$, with begin $\boldsymbol{p}$ decomposed in $\{n\}$.

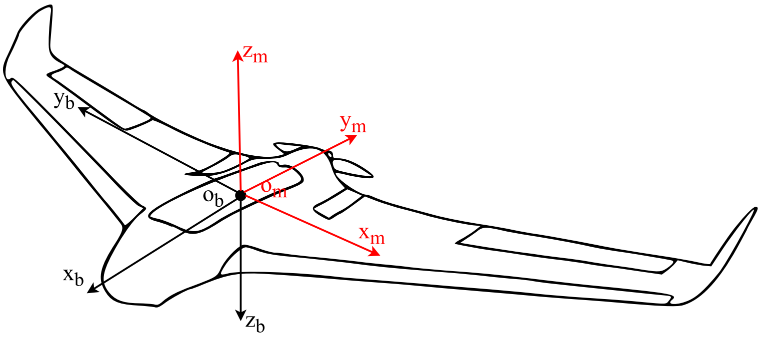

Figure 2.4: Body-fixed reference frame on a fixed wing UAV along with a measurement reference frame.

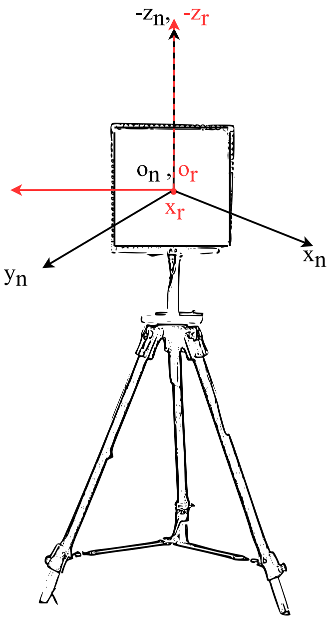

Figure 2.5: PARS reference frame along with a NED reference frame.

2.3 Representation of Pose

The pose of an object, such as a UAV, is a term used to describe both the position, often described in Cartesian coordinates, and the attitude or orientation, represented by either Euler angles, rotation matrix or unit quaternions, [6].

The pose of the UAV describes the body frame, $\mathcal{F}_b$, relative to another frame, such as the NED frame, $\mathcal{F}_n$. The position is then defined by the vector $\boldsymbol{p}^n_{nb} \in \mathbb{R}^3$, and the orientation is described by the rotation matrix $\boldsymbol{R}_{nb} \in SO(3)$, which belongs to the special orthogonal group $SO(3)$ describing the group of 3D rotation matrices defined according to

$$SO(3) := \{\boldsymbol{R} \in \mathbb{R}^{3x3} : \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}, \det(\boldsymbol{R}) = 1\}, \tag{2.2}$$

as in e.g. [2, Ch. 2],[8].

Comprising this information, the pose is fully described according to the homogeneous transformation matrix

$$\boldsymbol{T}_{nb} = \begin{bmatrix} \boldsymbol{R}_{nb} & \boldsymbol{p}^n_{nb} \\ \boldsymbol{0} & 1 \end{bmatrix} \in SE(3). \tag{2.3}$$

The transformation belongs to the Special Euclidean Group $SE(3)$ which describes the group of rigid motion in 3D defined as

$$SE(3) := \{(\boldsymbol{R}, \boldsymbol{p}) : \boldsymbol{R} \in SO(3), \boldsymbol{p} \in \mathbb{R}^3\}. \tag{2.4}$$

Given two matrices belonging to $SE(3)$, $\boldsymbol{T}_1 = (\boldsymbol{R}_1, \boldsymbol{p}_1), \boldsymbol{T}_2 = (\boldsymbol{R}_2, \boldsymbol{p}_2) \in SE(3)$, the group operation is defined

$$T_1 \cdot T_2 = (R_1R_2, p_1 + R_1p_2), \tag{2.5}$$

according to [2, Ch. 2],[8].

2.3.1 Alternative representations of attitude

In (2.3), the attitude is represented as a rotation matrix. Alternatively, the attitude can be described by Euler angles or unit quaternions, [2, Ch. 2]. These three representations have their pros and cons and are therefore applied interchangeably for different purposes.

Euler angels

Euler angles are a three-dimensional parameterisation $\boldsymbol{\Theta} = [\phi, \theta, \psi]$, where ϕ is the roll angle, θ is the pitch angle, and ψ is the yaw angle. These are angles about the x-, y- and z-axis of the reference frame, respectively. Euler angles are rotated sequentially where one common sequence is $z-y-x$, which rotates first about the z-axis, followed by the y-axis and then lastly about the x-axis. The Euler angles, using the $z-y-x$ sequence, relate to the rotation matrices

$$\begin{aligned} \boldsymbol{R}(\boldsymbol{\Theta}) &= \boldsymbol{R}_z(\psi)\boldsymbol{R}_y(\theta)\boldsymbol{R}_x(\phi) \\ &= \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \end{aligned} \tag{2.6}$$

as given in [2, Ch.2].

Euler angles are useful as they provide a minimal representation of the orientation with the same number of variables as there are degrees of freedom. Additionally, they are intuitive and easy to visualise. However, they suffer from singularities such as Gimbal lock, where an infinite number of Euler angles provide valid attitude representations, [2, Ch. 2].

Unit Quaternion

The unit quaternion is a four-dimensional vector which consists of a scalar component, $\eta \in \mathbb{R}$, and a vector component, $\boldsymbol{\epsilon} \in \mathbb{R}^3$ defined as

$$\boldsymbol{q} := \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix}. \tag{2.7}$$

A unit quaternion is modelled such that its norm is the unit norm, i.e

$$||\boldsymbol{q}|| = \sqrt{\eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2} = 1, \tag{2.8}$$

which yields a degree of freedom equal to three. Multiplication with unit quaternion is called the Hamiltonian quaternion product denoted, ⊗, such that

$$\boldsymbol{q}_3 = \boldsymbol{q}_1 \otimes \boldsymbol{q}_2 = \begin{bmatrix} \eta_{q_1}\eta_{q_2} - \boldsymbol{\epsilon}_{q_1}^T \boldsymbol{\epsilon}_{q_2} \\ \eta_{q_1}\boldsymbol{\epsilon}_{q_2} + \eta_{q_2}\boldsymbol{\epsilon}_{q1} + S(\boldsymbol{\epsilon}_{q_1})\boldsymbol{\epsilon}_{q_2} \end{bmatrix} \tag{2.9}$$

A unit quaternion can also be defined according to

$$\boldsymbol{q} := \begin{bmatrix} \cos(\frac{\alpha}{2}) \\ \boldsymbol{n}\sin(\frac{\alpha}{2}) \end{bmatrix}, \tag{2.10}$$

where α is the angle about the unitvector $\boldsymbol{\eta}$ as is often known as the angle-axis representation. The quaternion is difficult to intuitively interpret compared to Euler angles, but they are efficient for numerical calculation and avoiding singularities, [2, Ch. 2].

2.4 Lie Theory

The pose is part of Lie groups which deem it necessary to introduce Lie theory. Specifically, the $SO(3)$ and $SE(3)$ are part of the Matrix Lie Groups, [10], and since the pose lies on the manifold, it complicates perturbations, representing uncertainty and performing differentiation of said pose. To illustrate, adding increments to a rotation matrix belonging to $SO(3)$ does not necessarily yield a rotation which still belongs to $SO(3)$, as indicated in

$$\boldsymbol{R} \in SO(3), \boldsymbol{\delta R} \in SO(3), \qquad \boldsymbol{R} + \boldsymbol{\delta R} \notin SO(3) \tag{2.11}$$

Lie Theory provides the framework for handling variables on the manifold correctly, [8].

2.4.1 Lie Groups

Lie groups satisfy the properties of both a mathematical group and a smooth manifold. A group, noted $(\mathcal{G}, \circ)$, is a set $\mathcal{G}$ with a composition operator that satisfies the axioms

$$\begin{array}{rcl} \text{Closure under} \quad \circ & : & \mathcal{X} \circ \mathcal{Y} \in \mathcal{G} \\ \text{Identify} \quad \mathcal{E} & : & \mathcal{E} \circ \mathcal{X} = \mathcal{X} \circ \mathcal{E} = \mathcal{X} \\ \text{Inverse} \quad \mathcal{X}^{-1} & : & \mathcal{X} \circ \mathcal{X} = \mathcal{X}^{-1} \circ \mathcal{X}^{-1} = \mathcal{E} \\ \text{Associativity} & : & (\mathcal{X} \circ \mathcal{Y}) \circ \mathcal{Z} = \mathcal{X} \circ (\mathcal{Y} \circ \mathcal{Z}) \end{array}$$

for elements $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \in \mathcal{G}$.

The pose is said to be on the manifold, which communicates its existence on a topological structure. The manifold is a geometric object which locally looks like Euclidean space but may have a complex structure overall. For example, this complex structure could be a sphere which looks like a plane locally, visualised in Figure 2.6, [8]. Therefore, local operations on the manifold look like operations in the vector space. This is called the tangent space, and each local point on the manifold has a corresponding tangent space, where traditional linear algebra and calculus apply, [8].

There is a specific point on the manifold called the identity, $\mathcal{E}$, illustrated in Figure 2.7. This element captures the algebraic structure and commutation of the group, also similar to origo in a Cartesian frame; it serves as a reference point for group transformations. Therefore, the identity tangent space operations are unique and defined as Lie Algebra, denoted $\mathfrak{m}$. Furthermore, the tangent space at the identity for $SO(3)$ and $SE(3)$ is denoted $\mathfrak{so}(3)$ and $\mathfrak{se}(3)$, respectively. The $\mathfrak{so}(3)$ space also consists

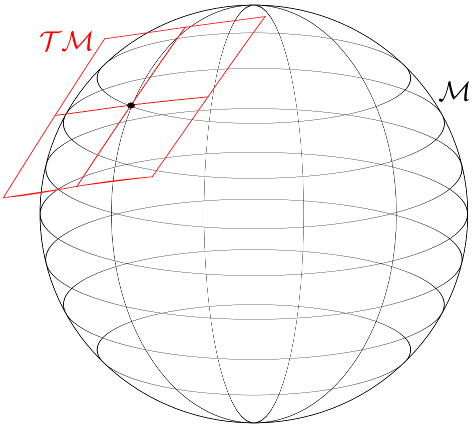

Figure 2.6: Illustration of a spherical manifold globally, $\mathcal{M}$, and a tangent plane, $\mathcal{TM}$, locally. Inspired from [8].

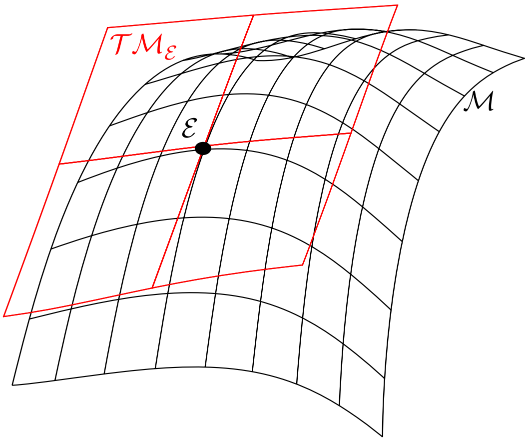

Figure 2.7: Illustration of a manifold, $\mathcal{M}$, with the identity point, $\mathcal{E}$ and the identity tangent plane, $\mathcal{TM}_{\mathcal{E}}$. Inspired from [8].

of 3×3 skew-symmetric matrices and any skew-symmetric matrix is given by a vector in $\mathbb{R}^3$ with the $(\bullet)^\wedge$ operator yielding,

$$\boldsymbol{\omega}^\wedge = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} = \boldsymbol{S}(\boldsymbol{\omega}) \in \mathfrak{so}(3) \tag{2.13}$$

and with the $(\bullet)^\vee$ operation

$$\boldsymbol{S}^\vee(\boldsymbol{\omega}) = \boldsymbol{\omega} \in \mathbb{R}^3 \tag{2.14}$$

for the opposite operation, [8, 11]. Conversion between Lie algebra and vector space on the tangent plane is illustrated in Figure 2.9.

2.4.2 Exponential and Logarithmic mapping

To operate on the manifold according to the axioms, it is necessary to be able to convert between the tangent space and the manifold. The capitalised exponential operator maps an element from the vector space to the manifold according to

$$\begin{aligned} &\mathrm{Exp}: \mathbb{R}^m \rightarrow \mathcal{M}, \\ &\mathrm{Exp}(\tau^{\mathcal{E}}) = \delta^{\mathcal{E}}. \end{aligned} \tag{2.15}$$

For the special orthogonal group the exponential map maps $\mathfrak{so}(3) \rightarrow SO(3)$, [8, 11]. Conversely, the capitalised logarithmic operator maps an element from the manifold to the vector space,

$$\begin{aligned} &\mathrm{Log}: \mathcal{M} \rightarrow \mathbb{R}^m \\ &\mathrm{Log}(\delta^{\mathcal{E}}) = \tau^{\mathcal{E}} \end{aligned} \tag{2.16}$$

according to [8]. This is visualised in Figure 2.8. A complete overview of the mappings related to Lie groups is provided in Figure 2.9. Furthermore, mapping from the Lie algebra space and the manifold is defined according to

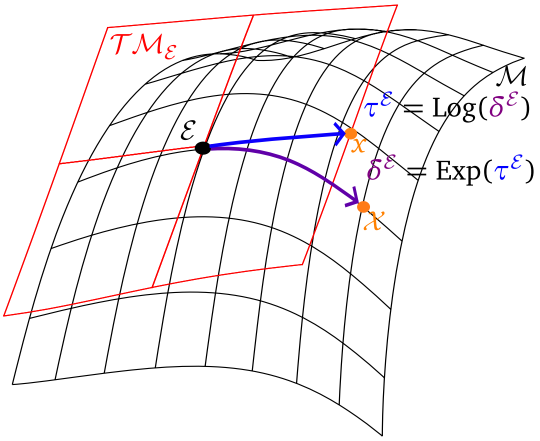

Figure 2.8: Illustration of a manifold, $\mathcal{M}$, a tangent plane, $\mathcal{TM}_\mathcal{E}$ and their relation through Logarithmic and Exponential mapping. Inspired from [8].

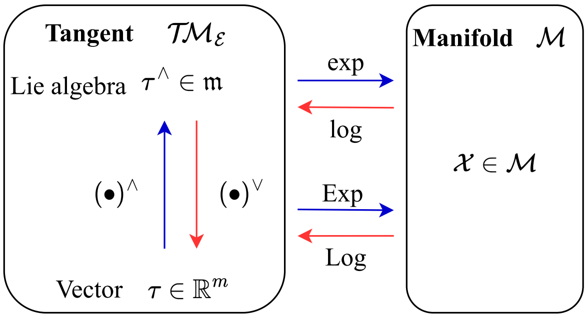

Figure 2.9: Mappings between the tangent plane, $\mathcal{TM}_\mathcal{E}$, and the manifold, $\mathcal{M}$, including the two representations on the tangent plane, lie algebra $\tau^\wedge \in \mathfrak{m}$ and vector space $\tau^\wedge \in \mathbb{R}^m$. Inspired from [10].

$$\begin{aligned} &\exp : \mathfrak{m} \rightarrow \mathcal{M}, \\ &\exp(\tau^{\mathcal{E},\wedge}) = \delta^\mathcal{E}. \end{aligned} \tag{2.17}$$

while mapping from the manifold to the Lie algebra space is defined as

$$\begin{aligned} &\log : \mathcal{M} \rightarrow \mathfrak{m} \\ &\log(\delta^\mathcal{E}) = \tau^{\mathcal{E},\wedge} \end{aligned} \tag{2.18}$$

2.4.3 Addition and subtraction on Manifold

To clarify the group transformations when using the group operator, $\circ$, addition and subtraction are introduced. Based on the elements $\mathcal{X}, \mathcal{Y}, \delta^\mathcal{X}, \delta^\mathcal{E} \in \mathcal{M}$ defined in Figure 2.10 and $\tau^\mathcal{X} = Log(\delta^\mathcal{X}), \tau^\mathcal{E} = \delta^\mathcal{E} \in \mathbb{R}^m$, the following group transformations hold

$$\begin{aligned} \mathcal{Y} &= \mathcal{X} \circ \delta^\mathcal{X} = \mathcal{X} \circ \text{Exp}\left(\tau^\mathcal{X}\right) \\ &= \delta^\mathcal{E} \circ \mathcal{X} = \text{Exp}\left(\tau^\mathcal{E}\right) \circ \mathcal{X} \end{aligned} \tag{2.19}$$

$$\begin{aligned} \tau^\mathcal{X} &= \text{Log}\left(\mathcal{X}^{-1} \circ \mathcal{Y}\right) \\ \tau^\mathcal{E} &= \text{Log}\left(\mathcal{Y} \circ \mathcal{X}^{-1}\right) \end{aligned} \tag{2.20}$$

This is clarified according to [8] with the addition, $\oplus$, and subtraction,$\ominus$, notation

$$\begin{aligned} \mathcal{Y} &= \mathcal{X} \oplus \tau^\mathcal{X} = \mathcal{X} \circ \text{Exp}\left(\tau^\mathcal{X}\right) \\ \tau^\mathcal{E} &= \mathcal{Y} \ominus \mathcal{X} = \text{Log}\left(\mathcal{Y} \circ \mathcal{X}^{-1}\right) \end{aligned} \tag{2.21}$$

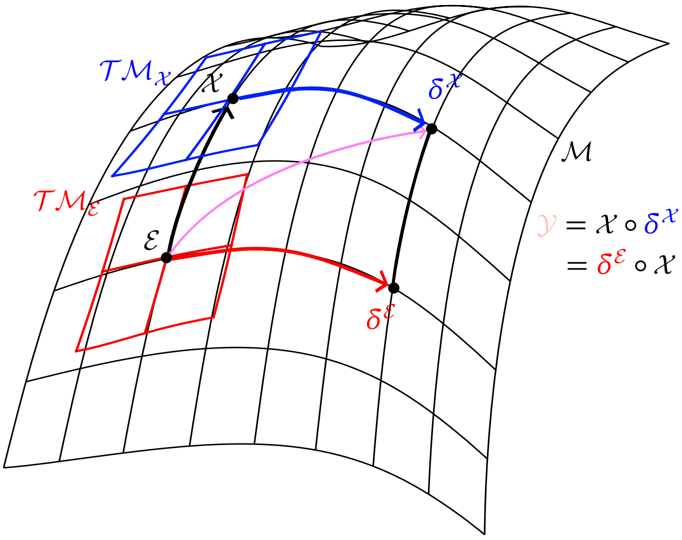

Figure 2.10: Illustrations of the operator, $\circ$, on the Manifold, $\mathcal{M}$, with three points $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{E} \in \mathcal{M}$ and the pertubations $\delta^\mathcal{E}$, $\delta^\mathcal{X} \in \mathcal{M}$. Inspired from [8].

These are also referred to as right plus and right minus. There are also left plus and left minus but are deemed unnecessary for this thesis.

2.4.4 Derivatives

Taking the derivative of a function f : $\mathcal{M} \rightarrow \mathcal{N}$ at a point $\mathcal{X} \in \mathcal{M}$ yield the jacobian $\boldsymbol{J}_\mathcal{X}$ which is computed as

$$J_\mathcal{X} = \frac{\partial f(\mathcal{X})}{\partial \mathcal{X}} = \lim_{\tau^\mathcal{X} \to 0} \frac{f(\mathcal{X} \oplus \tau^\mathcal{X}) \ominus f(\mathcal{X})}{\tau^\mathcal{X}} = \frac{\mathrm{Log}(f(\mathcal{X}^{-1} \circ f(\mathcal{X} \circ \mathrm{Exp}(\tau^\mathcal{X}))))}{\tau^\mathcal{X}} \tag{2.22}$$

where $\tau^\mathcal{X}$ is a infinitesimal small perturbation, according to [8]. This is the right derivative using the right plus operation. For small perturbations, the first-order Taylor expansion holds

$$f(\mathcal{X} \oplus \tau^\mathcal{X}) = f(\mathcal{X}) \oplus J_\mathcal{X} \tau^\mathcal{X} \tag{2.23}$$

2.5 Statistical prerequisites

Statistical principles provide the necessary tools to quantify uncertainty and variability in measurements and models. This is crucial in state estimation, as it is mainly working with inference utilising noisy data. This section introduces random variables and their probability distributions with a focus on the traditional Gaussian distribution. Further, the Bayesian statistical framework is introduced and used to formalise probabilistic estimation. Lastly, probabilistic graphical models, stochastic processes and relevant statistical error metrics are presented.

2.5.1 Random variables and probability distributions

A random variable does not have a set value. Instead, it is a mathematical object which describes all possible outcomes of a random phenomenon along with their probabilities. It is defined by a probability distribution function, where a random variable X is defined by its distribution $p(x)$,

$$X \sim p(x), \tag{2.24}$$

according to [12].

There are two types of random variables; continuous and discrete. A random phenomenon, such as rolling a dice, is a discrete random variable, while a UAV's location is a continuous random variable. A dice has six possible outcomes, while the position of a UAV can take on an infinite number of values. This is also called the outcome space of a random variable,[12].

The probability of a continuous random variable taking on one specific outcome equals zero since the outcome space is infinitely large. Therefore, instead of checking the probability of a specific value in the outcome space, it is more interesting looking at an interval. For modelling this probability, two important concepts are introduced; the Cumulative Distribution Function (CDF) and the Probability Density Function (PDF). The CDF of a random variable X is denoted $P(x)$ and defined as

$$\Pr\{X < x\} = P(x), \tag{2.25a}$$
$$\Pr\{X > x\} = 1 - P(x), \tag{2.25b}$$
$$\Pr\{x_1 < X < x_2\} = P(x_2) - P(x_1), \tag{2.25c}$$

where (2.25a) entail the probability for X to be less or equal to a specific outcome x. While (2.25c) describe the probability for X to lie within an interval of two specific outcome values x_1 and x_2, [12].

Furthermore, by taking the derivative of the CDF, the PDF is obtained,

$$p(x) = \frac{\partial P(x)}{\partial x}. \tag{2.26}$$

This yield the probability density describing the rate of change of the CDF at any given x. Consequently, the CDF can be described as an integral of the PDF or the sum of the PDF at an interval x, defined

$$\Pr\{X < x\} = \int_{-\infty}^{x} p(x)dx, \tag{2.27a}$$
$$\Pr\{X > x\} = \int_{x}^{\infty} p(x)dx, \tag{2.27b}$$
$$\Pr\{x_1 < X < x_2\} = \int_{x_1}^{x_2} p(x)dx. \tag{2.27c}$$

Lastly, according to [12], taking the integral over the entire PDF the probability adds up equal to 1 i.e.

$$\int_{-\infty}^{\infty} p(x)dx = 1. \tag{2.28}$$

Communicating the probability distribution of a continuous random variable refers to its PDF, as it entails the variable's probability in the entire outcome space. When talking about probability, it entails the CDF or integral of the PDF.

2.5.2 Conditionality, marginalisation, independence and Bayes rule

Two random variables with probability $X \sim p(x)$ and $Y \sim p(y)$, have a **conditional probability** given by

$$p(x|y) = \frac{p(x \cap y)}{p(y)}. \tag{2.29}$$

These are **independent** if the joint probability is given by

$$p(x \cap y) = p(x, y) = p(x)p(y), \tag{2.30}$$

while their **marginal probability** is defined as

$$p(x) := \sum_{n} p(x|y_n)p(y_n), \tag{2.31}$$

also referred to as the total probability theorem, [12].

The **Bayes' rule** is given as

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}, \tag{2.32}$$

formulating the process of switching conditionality.

2.5.3 The Univariate Gaussian Probability Distribution

The Gaussian distribution is widely used because of its many valuable properties. It is also known as the Normal distribution and is characterised by a bell-shaped curve. Its PDF, $p(x)$, describes the likelihood of observing a continuous random variable, x, using a mean value, $\bar{x}$ and a variance, σ^2. From [12], the probability distribution is mathematically described by

$$p(x) = \mathcal{N}\left(x; \bar{x}, \sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\bar{x})^2}{2\sigma^2}\right). \tag{2.33}$$

2.5.4 The Multivariate Gaussian Probability Distribution

The multivariate Gaussian probability distribution takes on the probability distribution of multiple continuous random variables comprised in the vector $\boldsymbol{X}$. The probability is described by

$$\boldsymbol{X} \sim \mathcal{N}(\bar{\boldsymbol{x}}, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \bar{\boldsymbol{x}})\Sigma^{-1}(\boldsymbol{x} - \bar{\boldsymbol{x}})\right). \tag{2.34}$$

The mean of the random variable, $\bar{\boldsymbol{x}}$ is now a vector describing the expectation of each variable and the variance is exchanged with a covariance matrix, $\boldsymbol{\Sigma}$, defining both the variance for each variable on the diagonal and also the covariance between the variables. From [12], expanding the expectation and covariance in two dimensions results in

$$\underbrace{\begin{bmatrix} X \\ Y \end{bmatrix}}_{\boldsymbol{X}} \sim \mathcal{N}\left(\underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\boldsymbol{x}}; \underbrace{\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}}_{\bar{\boldsymbol{x}}}, \underbrace{\begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 \end{bmatrix}}_{\boldsymbol{\Sigma}} \right). \tag{2.35}$$

2.5.5 On manifold probability distributions

A random variable on the manifold, $\mathcal{X} \in \mathcal{M}$ can be described as a perturbation

$$\mathcal{X} = \bar{\mathcal{X}} \oplus \tau \tag{2.36}$$

where $\tau \in \mathbb{R}$ is a random variable in the tangent space $\mathcal{TM}$. The PDF of τ

$$\tau \sim \mathcal{N}(0, \sigma^2) \tag{2.37}$$

can be defined to induce a PDF on $\mathcal{X}$

$$\mathcal{X} \sim \mathcal{N}(\bar{\mathcal{X}}, \sigma^2) \tag{2.38}$$

according to [8].

2.5.6 Bayesian probability framework and inference

Because of the probabilistic nature of sensor data, it is necessary to build a statistical framework that takes uncertainty into consideration. The Bayesian perspective is one such framework, which views probability as a degree of belief and confidence in an event. The belief of the event is subjective and not based on trial and error.

For instance, it is desirable to obtain a subjective probability distribution on a random variable X. This can be obtained by using knowledge and experience about the event. Therefore, it is desirable to model the distribution using PDFs of prior information on the event, $p(x)$, and a PDF, $p(z|x)$, called the likelihood model. The prior distribution describes a prior belief of the event generally based on available information. At the same time, the likelihood models the relationship between an outcome or measurement, z and the event, x. It provides information on how likely an outcome is based on the underlying belief of the event. Applying the Bayes formula (2.32) yield a posterior distribution,

$$p(x|z) \propto p(z|x)p(x), \tag{2.39}$$

which defines the expected distribution of the event given some prior information and likelihood of measurements. This description of an event is a conditional density, also called probabilistic inference. Prior information and how they give rise to uncertain measurements are inferred to a posterior conditional density, [12]. This understanding of inferring knowledge and obtaining probability distributions of random variables is the fundamental framework in many state estimation techniques.

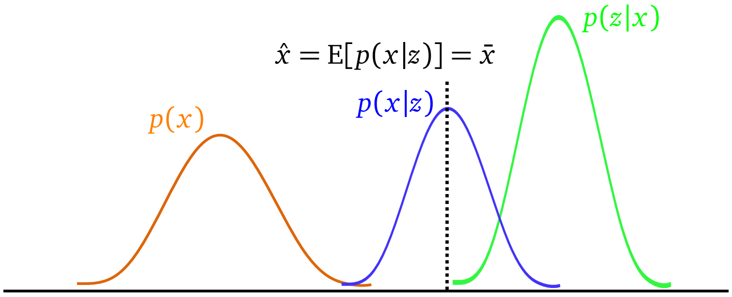

Figure 2.11: Visualisation of the computation of the MAP of Gaussian distributed likelihood and prior probabilities.

2.5.7 Probabilistic estimation

Estimation is the specific process of inferring knowledge about an unknown quantity $\boldsymbol{x}$ from data $\boldsymbol{z}$ which are related to $\boldsymbol{x}$. The relationship between $\boldsymbol{z}$ and $\boldsymbol{x}$ takes the form of the likelihood model, $p(\boldsymbol{z}|\boldsymbol{x})$. Additionally, in the Bayesian approach, each event is modelled with a priori of the event, $p(\boldsymbol{x})$. This yield the posteriori $p(\boldsymbol{x}|\boldsymbol{z}) \propto p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x})$. The estimation task is to extract the $\boldsymbol{x}$ that maximises the probability distribution of the posteriori. This is known as Maximum A Posteriori (MAP) and formulated as

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{z}) = \arg\max_{\boldsymbol{x}} p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x}) \tag{2.40}$$

If the random variable is Gaussian distributed, the MAP estimation is equivalent to finding the expected value of the a posteriori, $\bar{\boldsymbol{x}}$, [12], visualised in Figure 2.11.

Filtering, fixed-lag smoothing and smoothing

Filtering and smoothing are two different approaches to estimation mentioned in [6]. Filtering is the process of estimating the current random variable or state defined as

$$\hat{\boldsymbol{x}}_k = \arg\max_{\boldsymbol{x}_k} p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \arg\max_{\boldsymbol{x}_k} p(\boldsymbol{z}_k|\boldsymbol{x}_{1:k})p(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1},\boldsymbol{z}_{1:k}) \tag{2.41}$$

where $p(\boldsymbol{x}_k|\boldsymbol{x}_{1:k-1},\boldsymbol{z}_{1:k})$ is called the markov or process model, [6]. Smoothing, on the other hand, is the process of estimating the states of the entire time horizon at once stated as

$$\hat{\boldsymbol{x}}_{1:k} = \arg\max_{\boldsymbol{x}_{1:k}} p(\boldsymbol{x}_{1:k}|\boldsymbol{z}_{1:k}) = \arg\max_{\boldsymbol{x}_{1:k}} p(\boldsymbol{z}_{1:k}|\boldsymbol{x}_{1:k})p(\boldsymbol{x}_{1:k}|\boldsymbol{x}_{1:k-1},\boldsymbol{z}_{1:k}) \tag{2.42}$$

When smoothing over a set of the time horizon it is communicated as fixed-lag smoothing, where the lag defines how far back in time random variables are estimated, [6, 12].

When filtering the Markov assumption

$$\begin{aligned} p(x_k|x_1,....,x_{k-2},x_{k-1},x_k,z_1,...,z_{k-2},z_{k-1}) = p(x_k|x_{k-1}) \\ p(z_k|x_1,....,x_{k-2},x_{k-1},x_k,z_1,...,z_{k-2},z_{k-1}) = p(z_k|x_k) \end{aligned} \tag{2.43}$$

is assumed to hold. This assumption states that the process model and the measurement model depend only on the last state or measurement. The assumption is also visualised in Figure 2.12, [12].

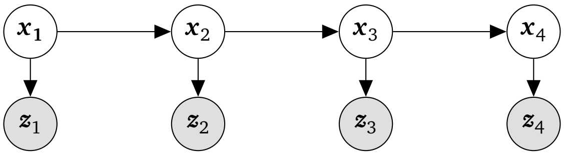

Figure 2.12: Example of a bayesian network with four states and four measurements as random variables.

2.5.8 Probabilistic graphical models

When the number of states and corresponding measurements increase, the estimation problem becomes complex. Probabilistic graphical models are helpful in representing relationships between states and measurements compactly. These models use structures, such as nodes and edges, to represent the probabilistic models, which in turn provide beneficial statistical properties that will simplify inference on large-scale systems. Two such graphical models are the Bayes network and factor graphs, [12, Ch. 12].

Bayes network

The Bayesian network or Bayes net consists of nodes representing known or unknown random variables and edges representing their relationships. This network is a directed acyclic graph which is a directed graph with no cycles, meaning that there is no way to start at any vertex in the graph and traverse a sequence of edges that eventually loops back to the starting vertex. This works naturally for models with a precise sequence and dependencies, [12, Ch. 12].

An example is depicted in Figure 2.12, with four states and one measurement for each state. The probability density of each node is constituted of all incoming edges. The random variable x_1 has no input edges, and therefore its probability distribution is not conditioned on other nodes yielding the PDF $p(x_1)$. The other nodes' distribution depend on adjacent nodes understood by the directional arrows. This yield the PDF for each node as $x_2 : p(x_2|x_1), x_3 : p(x_3|x_2), x_3 : p(x_4|x_3), z_1 : p(z_1|x_1), z_2 : p(z_2|x_2), z_3 : p(z_3|x_3), z_4 : p(z_4|x_4)$, [12, Ch. 12].

In the Bayes graphical structure, the joint probability density is equal to the product of each probability distribution related to each random variable or node. This is described as

$$p(\boldsymbol{\Theta}) = \prod_j p(\theta_j|\pi_j) \tag{2.44}$$

where $\boldsymbol{\Theta}$ is vector of all nodes, θ_j is node j and π_j is edge j. For the example in Figure 2.12, the resulting marginal density is given by

$$\begin{aligned} p(x_1, x_2, x_3, x_4, z_1, z_2, z_3, z_4) =& p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3)\times \\ & p(z_1|x_1)p(x_2|z_2)p(z_3|x_3)p(z_4|x_4) \end{aligned} \tag{2.45}$$

The joint probability can be used to marginalise out the probability of one state making it easier to infer knowledge on specific states, [12, Ch. 12].

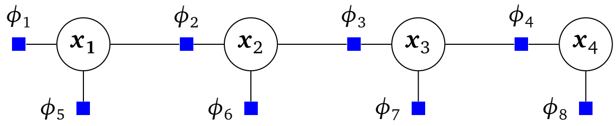

Figure 2.13: Example of a factor graph with non-directed edges between four nodes and eight factors.

Factor graph

A Factor Graph (FG) is a special kind of bipartite graph which again is a type of graph where its nodes can be divided into two sets. These are variables and factors, and the graph is constructed such that all edges connect a vertex in one set to a vertex in the other set, and no edges connect vertices within the same set. Bipartite graphs are often used to model situations where two distinct types of objects exist, and edges represent some relationship or interaction between them. The variable node represents random variables which are unknown and is to be estimated. In contrast, factor nodes represent either probabilistic relationships between the variable nodes or probabilistic conditions on the variable node. Figure 2.13 illustrates an FG with four variables and eight factors, [12, Ch. 12].

The joint probability density, also called the global function, is computed specifically for the random variables of interest

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \prod_j \phi_j(\boldsymbol{x}_j), \tag{2.46}$$

where $\boldsymbol{\Phi}$ is the global function, $\phi_k(\boldsymbol{x}_k)$ is a factor and $\boldsymbol{x}_k$ is the random variables adjacent to the factor, [12, Ch. 12]. Applying it to the example in Figure 2.13 the resulting global function is given by

$$\begin{aligned}\boldsymbol{\Phi}(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4) =& \phi_1(\boldsymbol{x}_1)\phi_2(\boldsymbol{x}_1,\boldsymbol{x}_2)\phi_3(\boldsymbol{x}_2,\boldsymbol{x}_3)\phi(\boldsymbol{x}_3,\boldsymbol{x}_4)\times && (2.47)\\ & \phi_5(\boldsymbol{x}_1)\phi_6(\boldsymbol{x}_2)\phi_7(\boldsymbol{x}_3)\phi_8(\boldsymbol{x}_4) && (2.48)\end{aligned}$$

It should be noted that the FG, as opposed to the Bayes net, operates with a global function denoted $\boldsymbol{\Phi}$ instead of the typical joint density $p(\boldsymbol{\Theta})$. This is because the FG can take on any complex model where each factor is any factored function dependent on states in its adjacent set. Instead of the factors only being probability distributions, it can take on any form of function. This will show to be helpful in modelling inference problems and let the factors be defined according to

$$\begin{aligned}\phi_1(\boldsymbol{x}_1) &= f_1(\boldsymbol{x}_1)\\ \phi_2(\boldsymbol{x}_1,\boldsymbol{x}_2) &= f_2(\boldsymbol{x}_2|\boldsymbol{x}_1)\\ \phi_3(\boldsymbol{x}_2,\boldsymbol{x}_3) &= f_3(\boldsymbol{x}_3|\boldsymbol{x}_2)\\ \phi_4(\boldsymbol{x}_3,\boldsymbol{x}_4) &= f_4(\boldsymbol{x}_4|\boldsymbol{x}_3)\\ \phi_5(\boldsymbol{x}_1) &= f_5(\boldsymbol{z}_1|\boldsymbol{x}_1) = l(\boldsymbol{x}_1;\boldsymbol{z}_1)\\ \phi_6(\boldsymbol{x}_2) &= f_6(\boldsymbol{z}_2|\boldsymbol{x}_2) = l(\boldsymbol{x}_2;\boldsymbol{z}_2)\\ \phi_7(\boldsymbol{x}_3) &= f_7(\boldsymbol{z}_3|\boldsymbol{x}_3) = l(\boldsymbol{x}_3;\boldsymbol{z}_3)\\ \phi_8(\boldsymbol{x}_4) &= f_8(\boldsymbol{z}_4|\boldsymbol{x}_4) = l(\boldsymbol{x}_4;\boldsymbol{z}_4)\end{aligned} \tag{2.49}$$

where the measurement models become a likelihood function where z_i are known, [12, Ch. 12].

Since the FG only has the random variables in the global function, it simplifies marginalisation compared to the Bayes net. This makes inference more efficient and more fitted for large-scale systems with numerous measurements. Therefore the actual value of the FG becomes present when finding marginal densities of older states and not only the last state, [12, Ch. 12].

2.5.9 Stochastic processes

Stochastic processes are models that describe a function of time with a typical behaviour of random fluctuation. These are useful when modelling noisy sensors since they exhibit random fluctuations in their data. Two such models are the Wiener process and white Gaussian noise. The Wiener process is known as the grandfather of stochastic processes according to [12], and to define the Wiener process, first a stochastic process, $x(t)$, is formulated according to

$$x(nT) = \sum_{i=1}^{n} x_i \tag{2.50}$$

where x_i are all independent and identically distributed random variables with expectation $\bar{x}_i = 0$ and variance $\sigma_i^2 = T$.

The Wiener process, $w(t)$, can the be defined as

$$w(t) = \lim_{T \to 0} x(t) \tag{2.51}$$

The central limit theory ensure that the Wiener process is Gaussian. The expectation equals zero because the sum of expectation of a Gaussians equals zero, while the variance is computed as

$$\mathrm{E}\left[w(t)^2\right] = \mathrm{Var}\left[\sum_{i=1}^{2}\right] = nT = \frac{t}{T}T = t \tag{2.52}$$

The computation finds that the variance increase as time increases. This has given the Wiener process the name random walk as it slowly increases and drifts over time, [12].

The white Gaussian noise is s sequence of independent and identically distributed Gaussian random variables in discrete time for the discrete case. In continuous-time white noise is defined as the derivative of the Wiener process

$$w(t) = \lim_{\Delta \to 0} \frac{w(t+\Delta) - w(t)}{\Delta}. \tag{2.53}$$

2.5.10 Statistical error

To measure the accuracy of estimates there are various methods to assert the error. Each method measures specific attributes.

Mean error

The Mean Error (ME), given as

$$ME = \frac{1}{N}\sum_{i=1}^{N}(\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i) \tag{2.54}$$

compute the average difference between the estimated value, $\hat{\boldsymbol{x}}_i$, and the true value, $\boldsymbol{x}_i$, over all the estimated states. If the ME is too high or too low, the estimates have a consequent bias. However, this does not inform on the magnitude or variability of the error, [13].

Mean absolute error

Mean Absolute Error (MAE), defined as

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i| \tag{2.55}$$

inform on the average magnitude of the error between the estimated and the true states, [13].

Standard deviation

The Standard Deviation (STD), formulated as

$$STD = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\hat{\boldsymbol{x}}_i - \bar{\hat{\boldsymbol{x}}})^2} \tag{2.56}$$

inform on the variability of the estimations by subtracting the ME, $\bar{\hat{\boldsymbol{x}}}$, from the estimates, [14].

Root mean square error

Root Mean Square Error (RMSE), given by

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i)^2} \tag{2.57}$$

can be compared to MAE and STD as it reflects both the average magnitude of error and the spread of those errors. Furthermore, it determine whether the estimates contain large and infrequent errors. The larger the difference between RMSE and MAE the more inconsistent the error size, [15].

2.6 Least Squares Optimisation

Least Squares Optimisation (LSO) is a powerful technique and, in simple terms, the task is to reduce errors between predicted data and true data [6, 8]. An optimisation problem can be formulated according to

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \mathrm{f}(\boldsymbol{x}) \tag{2.58}$$

where the optimal variable, $\boldsymbol{x}^*$, is found by minimising the objective function, $\mathrm{f}(\boldsymbol{x})$, with respect to $\boldsymbol{x}$. In the Least Squares (LS) formulation, the objective function is composed of the norm of squared residuals, $\boldsymbol{r}(\boldsymbol{x})$ defined by

$$\mathrm{f}(\boldsymbol{x}) = \boldsymbol{r}^T(\boldsymbol{x})\boldsymbol{r}(\boldsymbol{x}) = ||\boldsymbol{r}(\boldsymbol{x})||^2 \tag{2.59}$$

according to [8].

This section first introduces the uncomplicated linear LS problem before building on that with the more complex nonlinear problem.

2.6.1 Linear least squares

In linear LS problems, the residuals are linear and can be defined according to

$$\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \tag{2.60}$$

which results in the least square optimisation formulation

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||^2 \tag{2.61}$$

The $\boldsymbol{x}$ that minimise the squared error between $\boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{b}$ is the optimal point, $\boldsymbol{x}^*$, [8]. Because this is an unconstrained linear problem, it simplifies the process of finding the minimum. The objective function's minimum exists where the objective function's gradient equals zero. This can be computed with

$$\begin{aligned} \mathrm{f}(\boldsymbol{x}) &= ||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||^2 && (2.62a)\\ \nabla f(\boldsymbol{x}) &= \nabla||\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}||^2 && (2.62b)\\ &= \nabla 2\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}) = 0 && (2.62c)\\ \boldsymbol{x}^* &= (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b} && (2.62d) \end{aligned}$$

where $\boldsymbol{x}^*$ is the solution.

2.6.2 Nonlinear least squares

In the nonlinear LS problem, the residuals are no longer linear; therefore, the (2.62d) can not be used directly. One method to circumvent this problem is linearising the nonlinear residuals before applying (2.62d). The (2.62d) will, in this case, act as a step, which will move towards a set of $\boldsymbol{x}$ that will minimise the objective function, [8]. The nonlinear residuals, $r(\boldsymbol{\mathcal{X}})$ where $\mathcal{X} \in \mathcal{M}$, are linearised according to

$$\begin{aligned} f(\boldsymbol{\mathcal{X}}) &= ||r(\boldsymbol{\mathcal{X}})|| = \sum_i ||r_i(\boldsymbol{\mathcal{X}}_i)|| \\ r_i(\boldsymbol{\mathcal{X}}_i) &= r_i(\boldsymbol{\mathcal{X}}_i \oplus \boldsymbol{\tau}_i) \approx r_i(\boldsymbol{\mathcal{X}}_i) + \boldsymbol{J}_{\boldsymbol{\mathcal{X}}_i}\boldsymbol{\tau}_i \end{aligned} \tag{2.63}$$

The residual at timestep i is linearised about the last estimate, alternatively the initial guess, using a Tayler expansion. The jacobian, $\boldsymbol{J}_{\boldsymbol{\mathcal{X}}}$, is computed according to (2.22), and $\boldsymbol{\tau} \in \mathbb{R}^m$ is a concatenation of tangent vectors, [8].

The linearised residual is reformulated to a linear LS problem as

$$\begin{aligned} f(\mathcal{X}) = f(\hat{\mathcal{X}} \oplus \boldsymbol{\tau}) &\approx \sum_i ||r(\hat{\mathcal{X}}_i) - \boldsymbol{J}_{\mathcal{X}_i}\boldsymbol{\tau}_i||^2 \\ &= \sum_i ||\boldsymbol{J}_{\mathcal{X}_i}\boldsymbol{\tau}_i - (-r(\hat{\mathcal{X}}_i))||^2 \\ &= \sum_i ||\boldsymbol{A}_i\boldsymbol{\tau}_i - \boldsymbol{b}_i||^2 \\ &= ||\boldsymbol{A}\boldsymbol{\tau} - \boldsymbol{b}||^2 \end{aligned} \tag{2.64}$$

where $\boldsymbol{\tau} = \mathcal{X} \ominus \hat{\mathcal{X}}$ is the state update vector. In order to reach the optimal point, it is necessary to re-linearise and step towards it recursively, as each step is only a simplification and needs iterative refinement, [8].

The Gauss-Newton and Levenberg-Marquardt algorithms are two techniques that can solve nonlinear LS optimisation problems, [6]. However, they differ in their formulation of the step direction. The Gauss-Newton apply (2.62d), but this method can have challenges when the inverse is singular and impossible to compute. Levenberg-Marquardt then provides an alternative method which is not as accurate but is more than good enough, [8].

2.7 Inertial Measurement Unit

An IMU is a computer that continuously measures and records specific force and angular rates exerted on it. The IMU mainly uses a three-axis accelerometer and a three-axis gyroscope, also called an Attitude Rate Sensor (ARS), to measure the forces. It can also provide additional sensors, such as a one-axis barometer, [9, Ch. 4].

An accelerometer is a sensor measuring acceleration on the device. There are different accelerometers, but they build on the simple principle of Newton's second law, $\sum F = ma = kx$. A simple accelerometer can measure an object's displacement when under accelerating force. Based on this displacement, one can compute the force, F, on the object and, consequently, the acceleration, a, when the mass, m, and spring constant, k, are known, [9, Ch. 4].

A gyroscope is a sensor which measures angular rate, ω. The traditional definition of a gyroscope is the spinning-mass gyroscope based on the conservation of angular momentum, which state; the angular momentum of a body relative to the internal space will remain unchanged unless acted upon by a torque. Placing an instrument inside a body free to rotate will remain aligned with the internal space as the body rotates. Then it is possible to measure the instrument's attitude, [9, Ch. 4].

The accelerometer measures three-axis specific force: the non-gravitational forces per unit mass. The measured force given in the IMU frame is defined as

$$\boldsymbol{f}_{imu}^{m} = \boldsymbol{f}_{nb}^{m} \tag{2.65a}$$

$$\boldsymbol{f}_{nb}^{m} = \frac{1}{m}(\boldsymbol{f}_{nb,tot}^{m} - \boldsymbol{f}_{nb,g}^{m}) \tag{2.65b}$$

$$\boldsymbol{f}_{nb}^{m} = \frac{1}{m}(m\boldsymbol{a}_{nb}^{m} - m\boldsymbol{g}_{nb}^{m}) \tag{2.65c}$$

$$\boldsymbol{f}_{nb}^{m} = \boldsymbol{a}_{nb}^{m} - \boldsymbol{g}_{nb}^{m} \tag{2.65d}$$

where the specific force equals the acting acceleration neglecting the gravitational forces. The reference frame, $\mathcal{F}_m$, is the IMU measurement frame describing its installation, depicted in Figure 2.4.

The angular rate measurement equations for a three-axis ARS are given by

$$\boldsymbol{\omega}_{imu}^{m} = \boldsymbol{\omega}_{nb}^{m} \tag{2.66}$$

according to [2, Ch. 14].

2.7.1 Sensory inaccuracies

The IMU produce sensor measurement at typically 50-200 Hz, and these readings have noise. Additionally, installation misalignment and other calibration errors propagate into the measurements. Therefore, both the accelerometer and ARS measurement model have errors, resulting in the more accurate model given by

$$\begin{aligned} \boldsymbol{f}_{imu}^{m} &= \boldsymbol{a}_{nb}^{m} - \boldsymbol{g}_{nb}^{m} + \text{error} \\ \boldsymbol{\omega}_{imu}^{m} &= \boldsymbol{\omega}_{nb}^{m} + \text{error} \end{aligned} \tag{2.67}$$

according to [9, Ch. 4].

The most prominent errors in the Inertial Measurement Unit (IMU) are bias, random noise, scale factor and cross-coupling. The bias is a constant error in the accelerometer and gyroscope and is often the most dominant error in the IMU. They are denoted $\boldsymbol{b}_{ars}^{b} = [b_{ars,x}^{b}, b_{ars,y}^{b}, b_{ars,z}^{b}]$ and $\boldsymbol{b}_{acc}^{b} = [b_{acc,x}^{b}, b_{acc,y}^{b}, b_{acc,z}^{b}]$ for the gyro and accelerometer, respectively, [9, Ch. 4].

Random noise is assumed to be zero-mean Gaussian white noise. The noise can originate from electrical noise in the IMU, mechanical instability in the accelerometer or resonance in the gyro. The random noise is denoted $\boldsymbol{w}_{ars}^{b} = [w_{ars,x}^{b}, w_{ars,y}^{b}, w_{ars,z}^{b}]$ and $\boldsymbol{w}_{acc}^{b} = [w_{acc,x}^{b}, w_{acc,y}^{b}, w_{acc,z}^{b}]$ for the gyro and accelerometer, respectively. The white noise can be reduced by applying a lowpass filter directly on the data. When integrating random noise, it becomes a random walk according to Section 2.5.9, and therefore the noise is also sometimes defined as a random walk, [9, Ch. 4].

Scale factor errors originate when applying unit conversion by the IMU. Cross-coupling errors are inaccuracies because of the misalignment of the sensitivity axes of the inertial sensors relative to the actual IMU frame and the uncertainty of the IMU frame relative to the body frame. These errors are minor and, therefore, often not included in the measurement model, [9, Ch. 4].

Adjusting the measurement models to include bias and random noise yields the new models

$$
\begin{aligned}
\boldsymbol{f}^m_{imu} &= \boldsymbol{a}^m_{nb} - \boldsymbol{g}^m_{nb} + \boldsymbol{b}^b_{acc} + \boldsymbol{w}^b_{acc} \\
\dot{\boldsymbol{b}}^b_{acc} &= \boldsymbol{w}^b_{acc} \\
\boldsymbol{\omega}^m_{imu} &= \boldsymbol{\omega}^m_{nb} + \boldsymbol{b}^b_{ars} + \boldsymbol{w}^b_{ars} \\
\dot{\boldsymbol{b}}^b_{ars} &= \boldsymbol{w}^b_{ars}
\end{aligned}
\tag{2.68}
$$

where the noise is zero-mean Gaussian white noise and the biases are modelled as a wiener process.

2.7.2 Incremental velocity and angular outputs

To obtain incremental outputs of the velocity, $\boldsymbol{v}^m_{imu}$, and the attitude, $\boldsymbol{\alpha}^m_{imu}$, the measurements can be integrated over each time step

$$
\begin{aligned}
\boldsymbol{v}^m_{imu}(t) &= \int_{t-Ts}^{t} \boldsymbol{f}^m_{imu}(t)dt = \Delta \boldsymbol{v}^m_{imu} \\
\boldsymbol{\alpha}^m_{imu}(t) &= \int_{t-Ts}^{t} \boldsymbol{\omega}^m_{imu}(t)dt = \Delta \boldsymbol{\alpha}^m_{imu}
\end{aligned}
\tag{2.69}
$$

These incremental outputs are called delta angle, $\Delta\boldsymbol{\alpha}^m_{imu}$, and delta velocity, $\Delta\boldsymbol{v}^m_{imu}$. However, delta velocity is not truly incremental velocity due to integrating specific force and not pure acceleration. Furthermore, by assuming the zero-order hold the difference between each step is approximately zero, i.e. $\boldsymbol{f}^m_{imu}(t) - \boldsymbol{f}^m_{imu}(t - Ts) \approx \boldsymbol{0}$ and $\boldsymbol{\omega}^m_{imu}(t) - \boldsymbol{\omega}^m_{imu}(t - Ts) \approx \boldsymbol{0}$, which in turn leads to

$$
\begin{aligned}
\boldsymbol{v}^m_{imu}(t) &= \boldsymbol{f}^m_{imu}(t)Ts = \Delta \boldsymbol{v}^m_{imu} \\
\boldsymbol{\alpha}^m_{imu}(t) &= \boldsymbol{\omega}^m_{imu} Ts = \Delta \boldsymbol{\alpha}^m_{imu}
\end{aligned}
\tag{2.70}
$$

according to [9, Ch. 4, p. 149]. It is important to note that because the measurements contain constant bias and noise, integrating them results in drift over time as the inaccuracies accumulate.

2.8 Inertial Navigation System

An INS consists of an IMU and a navigation computer. The computer calculates gravitational acceleration and manipulates IMU output data to obtain predictions of the position, velocity and attitude of the body to which the IMU is installed. The equations used for computing the pose and velocity are called the strapdown navigation equations, where accelerometer and angular rate measurements are used as input. The process of the INS is depicted in Figure 2.14. Firstly, the algorithm is initialised with necessary preliminary information about the system, followed by a continuous recursive propagation of the position, velocity and attitude using the IMU and a gravitational model. The strapdown equations depend on the reference frame used. For non-inertial reference frames, the rotation of the frame must be taken into account, [9, Ch. 5].

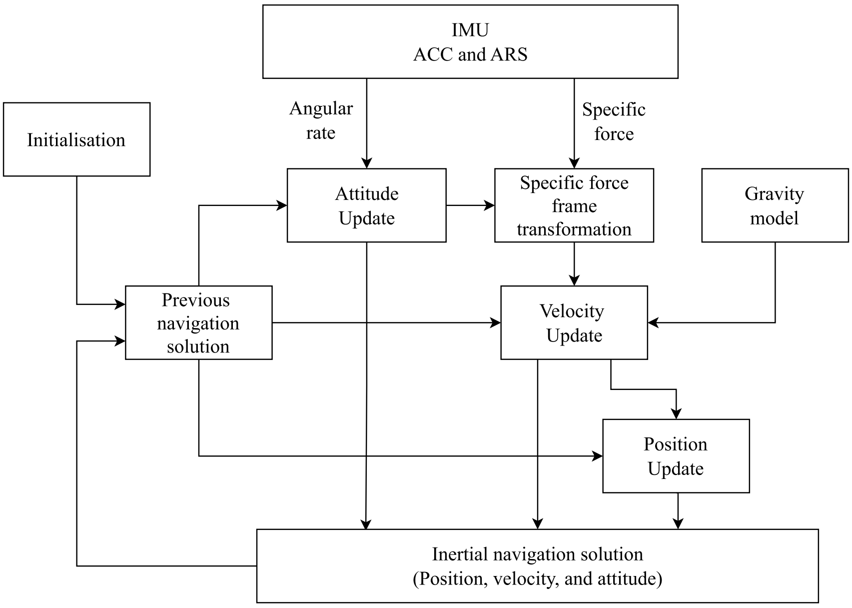

Figure 2.14: INS flow diagram for computing position, velocity and orientation with angular rate and specific force measurements as input. Inspired from Groves [9, Ch. 5, p. 167].

2.8.1 Inertial Strapdown Equations

The most straightforward formulation of the strapdown equations is formulated by assuming the navigation frame to be inertial. This can be the ECI frame or inertial NED frame. The NED frame is assumed inertial for simplification, traditionally done for low-speed applications that cover a smaller local area, such as boats. ECI and inertial NED have equivalent representations and it is only necessary to use a transformation to convert between them. Nevertheless, it is important to note that the inertial assumption for NED is a less accurate description than non-inertial reference frames, [2, 9].

The complete dynamical model of the strapdown equations is given as

$$\begin{aligned}
\dot{\boldsymbol{p}}^n_{nb} &= \boldsymbol{v}^n_{nb} \\
\dot{\boldsymbol{v}}^n_{nb} &= \boldsymbol{R}_{nm}(\boldsymbol{f}^m_{imu} - \boldsymbol{b}^b_{acc} - \boldsymbol{w}^b_{acc}) + \boldsymbol{g}^n_{nb}(\boldsymbol{p}^n_{nb}) \\
\dot{\boldsymbol{b}}^b_{acc} &= \boldsymbol{w}^b_{acc} \\
\dot{\boldsymbol{q}}_{nb} &= \frac{1}{2}\boldsymbol{q}_{nb} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}^m_{imu} - \boldsymbol{b}^b_{ars} - \boldsymbol{w}^b_{ars} \end{bmatrix} \\
\dot{\boldsymbol{b}}^b_{ars} &= \boldsymbol{w}^b_{ars}
\end{aligned} \tag{2.71}$$

for the state $\boldsymbol{x}^n_{nb} = [\boldsymbol{p}^n_{nb}, \boldsymbol{v}^n_{nb}, \boldsymbol{b}^b_{acc}, \boldsymbol{q}_{nb}, \boldsymbol{b}^b_{ars}] \in \mathbb{R}^3$, where $\boldsymbol{f}^m_{imu}$ and $\boldsymbol{\omega}^m_{imu}$ are modelled according to (2.68). The position and velocity are computed using specific force, while the attitude is calculated using the angular rates from the gyroscope. To avoid singularities and complex numerical calculations unit quaternions are used, [2, 9].

2.8.2 Inertial strapdown Equations on the manifold

Modelling the strapdown equations on the manifold yield

$$\begin{aligned}
\dot{\boldsymbol{p}}^n_{nb} &= \boldsymbol{v}^n_{nb} \\
\dot{\boldsymbol{v}}^n_{nb} &= \boldsymbol{R}_{nm}(\boldsymbol{f}^m_{imu} - \boldsymbol{b}^b_{acc} - \boldsymbol{w}^b_{acc}) + \boldsymbol{g}^n_{nb}(\boldsymbol{p}^n_{nb}) \\
\dot{\boldsymbol{b}}^b_{acc} &= \boldsymbol{w}^b_{acc} \\
\dot{\boldsymbol{R}}_{nb} &= \boldsymbol{R}_{nb}\boldsymbol{\omega}^{\wedge,m}_{imu} \\
\dot{\boldsymbol{b}}^b_{ars} &= \boldsymbol{w}^b_{ars}
\end{aligned} \tag{2.72}$$

where the states $\boldsymbol{\mathcal{X}}^n_{nb} = [\boldsymbol{p}^n_{nb}, \boldsymbol{v}^n_{nb}, \boldsymbol{b}^b_{acc}, \boldsymbol{R}_{nb}, \boldsymbol{b}^b_{ars}] \in \mathcal{M}$, are defined on the manifold. Rotation matrices are applied instead of quaternions to model the attitude and to propagate the states on the manifold, [2, 9, 11].

2.8.3 Non-inertial strapdown equations

When modelling the equations with non-inertial frames such as ECEF and non-inertial NED, it is necessary to consider the Earth's rotation. This is done by simply adding the rotation into the calculations of the velocity and the attitude as

$$\begin{aligned}
\dot{\boldsymbol{p}}^e_{eb} &= \boldsymbol{v}^e_{eb} \\
\dot{\boldsymbol{v}}^e_{eb} &= \boldsymbol{R}_{em}\boldsymbol{f}^m_{eb} + \boldsymbol{g}^e_b(\boldsymbol{p}^e_{eb}) \underbrace{-2S(\boldsymbol{\omega}^e_{ie})\boldsymbol{v}^e_{eb}}_{\text{Earth Rotation}} \\
\dot{\boldsymbol{q}}_{eb} &= \frac{1}{2}\boldsymbol{q}_{eb} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}^m_{eb} \end{bmatrix} \underbrace{-\frac{1}{2}\begin{bmatrix} 0 \\ \boldsymbol{\omega}^e_{ie} \end{bmatrix} \otimes \boldsymbol{q}_{eb}}_{\text{Earth Rotation}}
\end{aligned} \tag{2.73}$$

For larger areas, this is more accurate, [2, 9].

2.9 Global Navigation Satellite System

A GNSS can be used to obtain the absolute position of a UAV. This position is obtained by an internal GNSS antenna on the UAV, combining signals from at least four satellites and calculating the position. GNSS is an umbrella term that includes systems such as GPS and GLONASS. The GNSS provide position measurement at typically 1-5 Hz with an accuracy down to 2-10 meters, formulated as

$$\boldsymbol{y}_{gnss} = \boldsymbol{p}^e_{eb,gnss} = \boldsymbol{p}^e_{eb} + \boldsymbol{\epsilon}_{\text{gnss}}, \quad \boldsymbol{p}^e_{eb} \in \mathbb{R}^3, \quad \boldsymbol{\epsilon}_{gnss} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_{gnss}) \tag{2.74}$$

according to [2, Ch. 13].

2.9.1 Real-time and Post-Processed Kinematics

The GNSS measurements have noise and a few meters of accuracy. To obtain higher accuracy, two specific methods can be applied; Real-Time Kinematics (RTK) and Post-Processed Kinematics (PPK).

They incorporate errors induced by the environment by introducing base stations with a known position of high accuracy to estimate the potential errors. Based on these estimations, the GNSS measurements can be corrected to obtain a more accurate position and velocity. This increases the accuracy to less than 0.5m, [16, 17].

RTK provide real-time correction during flight to the GNSS data and is used to ensure accurate autonomous navigation in real-time. On the other hand, PPK is a method to post-process the GNSS data with the correction data from the base stations. This yields a more precise position than the RTK; however, this method is not real-time. Therefore, it is used in completed experiments to compute a more accurate position and velocity for analysis reasons, [16, 17].

2.9.2 Availability

A challenge with GNSS is that the satellite signals have a low signal-to-noise ratio which is problematic in areas with poor satellite reception. Additionally, this makes it prone to intentional attacks known as jamming and spoofing, [3]. Lastly, the data only arrives every few seconds, meaning that high-speed systems can only partially base their positioning information on the GNSS data, [17].

2.10 Phased Array Radio Systems

PARS is an alternative technique to GNSS for measuring position. Applying PARS makes it possible to obtain absolute positioning estimates, in addition to high-rate data transfer for communication. PARS sends out and receives radio signals, and with its beamforming properties, these signals can be directed towards the receiving radio. In turn, this yields excellent range and the ability to estimate the position of the receiving radio. However, this solution is local, as it requires antennas that are inside range for communication, [17].

The PARS comprises multiple individual antennas combined with a feed network consisting of amplifiers and phase shifters, visualised in Figure 2.15. A PARS perform beam-forming by using the phase-shifter to adjust the phase of each antenna element. This, in turn, directs the main beam towards one specific direction, depicted in Figure 2.16. Most of the energy is sent in one direction while some scatter. Using a two-dimensional PARS, the direction of the beam is defined by two angles called azimuth, ξ, and elevation angle, α, [18].

The computation of the azimuth and elevation angle, together with the range between the transmitter and receiver, yields the aircraft's spherical position relative to the antenna frame given as measurements,

$$\boldsymbol{y}_{pars} = \begin{bmatrix} y_\rho \\ y_\xi \\ y_\alpha \end{bmatrix} = \begin{bmatrix} \rho + \epsilon_\rho \\ \xi + \epsilon_\xi \\ \alpha + \epsilon_\alpha \end{bmatrix}, \qquad \boldsymbol{\epsilon}_{pars} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_s) \tag{2.75}$$

affected by noise, $\boldsymbol{\epsilon}_{pars}$. There are several algorithms for computing $[\rho, \xi, \alpha]$ based on the direction of arrival problem, [17]. When the range, azimuth angle and elevation angle are computed it is

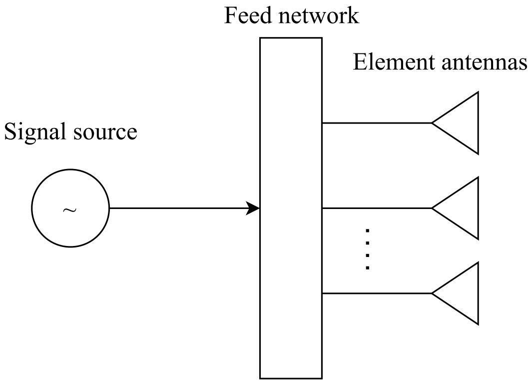

Figure 2.15: Architecture of a simple array antenna. Including a signal source, a feed network and variable number of element antennas. Inspired from [19, p. 1115]

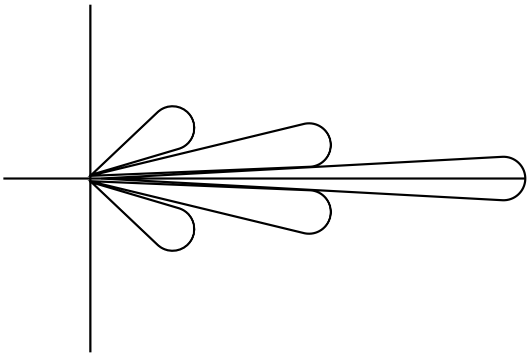

Figure 2.16: A visualisation of a signal transmission and its pattern synthesis. The beamformed directed signal transmit weaker signals in adjacent directions. Inspired from [19, p. 1115]

possible to calculate the Cartesian coordinates from the spherical position applying (2.1),

$$\boldsymbol{p}^r_{rb,pars} = \begin{bmatrix} p^r_{rb,x} \\ p^r_{rb,y} \\ p^r_{rb,z} \end{bmatrix} = y_\rho \begin{bmatrix} \cos(y_\xi)\cos(y_\alpha) \\ \sin(y_\xi)\cos(y_\alpha) \\ -\sin(y_\alpha) \end{bmatrix} \tag{2.76}$$

The nonlinear mapping result in a bias in the position coordinates which is corrected for with

$$\boldsymbol{p}^r_{rb,pars} = y_\rho \begin{bmatrix} (b_\xi b_\alpha)^{-1}\cos(y_\xi)\cos(y_\alpha) \\ (b_\xi b_\alpha)^{-1}\sin(y_\xi)\cos(y_\alpha) \\ -b_\alpha^{-1}\sin(y_{alpha}) \end{bmatrix}, \quad b_\xi = e^{-\frac{\sigma_\xi^2}{2}}, \quad b_\alpha = e^{-\frac{\sigma_\alpha^2}{2}} \tag{2.77}$$

according to [17]. It is also necessary to propagate uncertainty and cross terms through the covariance matrix when converting from spherical to Cartesian coordinates. The covariance matrix for PARS measurements, $\boldsymbol{R}_s$, is a diagonal matrix

$$\boldsymbol{R}_s = \begin{bmatrix} \sigma_\rho^2 & 0 & 0 \\ 0 & \sigma_\xi^2 & 0 \\ 0 & 0 & \sigma_\alpha^2 \end{bmatrix} \tag{2.78}$$

The measurement jacobian, $\boldsymbol{M}$, relative to $\boldsymbol{\epsilon}_{pars} = [\epsilon_\rho, \epsilon_\xi, \epsilon_\alpha]$ is

$$\boldsymbol{M} = \frac{\partial \boldsymbol{p}^e_{rb,pars}}{\partial \boldsymbol{\epsilon}} = \begin{bmatrix} \cos(\xi)\cos(\alpha) & -\rho\cos(\alpha)\sin(\xi) & -\rho\cos(\xi)\sin(\alpha) \\ \cos(\alpha)\sin(\xi) & \rho\cos(\xi)\cos(\alpha) & -\rho\sin(\xi)\sin(\alpha) \\ \sin(\alpha) & 0 & -\rho\cos(\alpha) \end{bmatrix} \tag{2.79}$$

The resulting covariance, $\boldsymbol{R}^r_{pars}$ is calculated by performing a similarity transform with the jacobian, $\boldsymbol{M}$,

$$\boldsymbol{R}^r_{pars} = \boldsymbol{M}\boldsymbol{R}_s\boldsymbol{M}^T \tag{2.80}$$

according to [17].

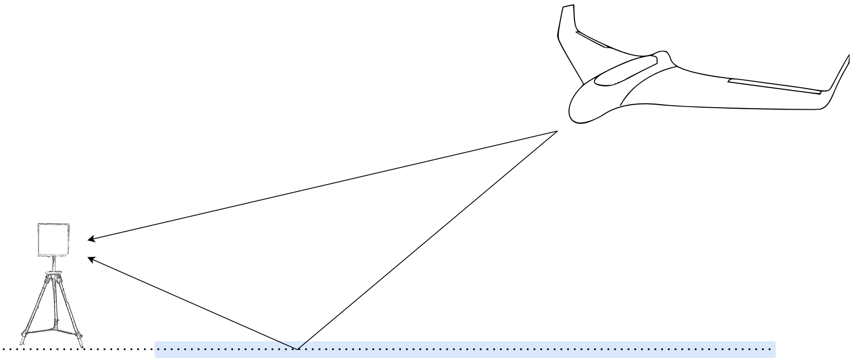

Figure 2.17: An illustration of the multipath propagation problem with PARS. The UAV fly across water and because of pattern synthesis, the signal is sent in different directions. These are reflected and arrive at the ground antenna.

2.10.1 Multipath Propagation and outlier rejection

Multipath propagation occurs when one signal from a PARS antenna reaches the receiving antenna from multiple angles, illustrated in Figure 2.17. This is a normal phenomenon when working with radio systems, and for UAV flights over the ocean, this is typical as signals are heavily reflected. When a signal is reflected and received at a later time from a different angle, it is subject to a phase shift. Therefore, the resulting position computation relative to the radio is incorrect for that signal. These multipath propagations are a challenging problem with PARS, and outlier rejection is applied to mitigate their impact. These use various techniques to determine the validity of a radio measurement, [17].

2.10.2 Loosely and tightly coupled systems

Loosely and tightly coupled systems refer to different levels of integration and interdependence between components of the PARS. Specifically, the PARS position measurements are input to the estimation algorithms in a loosely coupled system. On the other hand, a tightly coupled system integrates the PARS pseudo measurements range, azimuth and elevation, relating to the position The inverse of (2.76) is calculated as

$$\rho = \|\boldsymbol{p}^r_{rb,pars}\|_2 \tag{2.81}$$

$$\xi = \text{atan2}(p^r_{rb,y}, p^r_{rb,x}) \tag{2.82}$$

$$\alpha = \text{atan2}\left(p^r_{rb,z}, \sqrt{\left(p^r_{rb,x}\right)^2 + \left(p^r_{rb,y}\right)^2}\right) \tag{2.83}$$

to best capture the correlations between the measurements and to minimise bringing error into the system, e.g. avoid including error from the nonlinear mapping in (2.77). Therefore, the tightly coupled system generally produces more precise estimates than the loosely coupled system, [9, Ch. 14].

Chapter 3

State estimation schemes

Several methods exist for state estimation, and the choice of algorithm depends on the problem at hand. Two methods are KF and FGO, which differ in their underlying probabilistic modelling structure. Both take on the Bayesian probabilistic framework, but KF utilises the Bayes net, while FGO utilises the structure from FG models. Additionally, the KF technique is solely a filtering method, while factor graphs have the opportunity to do, filtering, fixed-lag smoothing or smoothing.

3.1 Kalman Filtering

Kalman Filtering has been a leading estimation methodology for many years and is still frequently applied today. The basic concept of Kalman Filtering was presented in the 1960s, and 1985 McGee and Schmidt published an article stating that Kalman Filtering is a fundamental analytic tool for solving some estimation problems. Today the field of KF is heavily researched, and various advanced formulations are produced to overcome the shortcomings of the filtering technique. Four formulations are introduced in this section, with each formulating building on the last; the traditional KF, Extended Kalman filter (EKF), Error State Kalman Filter (ESKF) and the MEKF. The traditional KF and the EKF are the simplest versions. The traditional KF assumes both gaussianity and linearity, which is rarely applicable. The EKF attempts to address more complicated nonlinear systems; however, this is often inaccurate. Therefore, more sophisticated algorithms are commonly used instead, such as the ESKF and the MEKF. To understand the Kalman Filtering concepts, KF and EKF are first formulated before ESKF and MEKF are presented in more detail, [4, 20]. Lastly, outlier rejection with the MEKF formulation is stated.

Kalman filtering finds the optimal state, $\boldsymbol{x}^*$, by solving the MAP estimator defined as

$$\hat{\boldsymbol{x}}_k = \arg\max_{\boldsymbol{x}_k} p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) = \arg\max_{\boldsymbol{x}_k} p(\boldsymbol{z}_k|\boldsymbol{x}_k)p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \tag{3.1}$$

which reformulate (2.41) by applying the Markov assumption in (2.43). Generally, the KF algorithms work by first predicting using the Markov model and then updating the predictions with the likelihood model. The Markov model is the strapdown equation defined in Section 2.8, while the likelihood model is either GNSS or PARS measurement model in Section 2.9 and Section 2.10, respectively. This process is completed iteratively to estimate the current state.

3.1.1 Traditional Kalman Filtering

As mentioned, the traditional KF assumes both the Markov and likelihood models to be Gaussian and linear. These assumptions yield the most straightforward formulation of the Kalman Filter and make it possible to obtain a closed-form solution to the filtering problem. The discrete Markov and measurement models are described statistically as

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) &= \mathcal{N}(\boldsymbol{x}_k; \boldsymbol{A_d}\boldsymbol{x}_k + \boldsymbol{B_d}\boldsymbol{u}_k, \boldsymbol{Q_d}) \\ p(\boldsymbol{z}_k|\boldsymbol{x}_k) &= \mathcal{N}(\boldsymbol{z}_k; \boldsymbol{C_d}\boldsymbol{x}_k, \boldsymbol{R_d}) \\ p(\boldsymbol{x}_0) &= \mathcal{N}(\boldsymbol{x}_0; \hat{\boldsymbol{x}}_0, \boldsymbol{P}_0) \end{aligned} \tag{3.2}$$

and with equivalent mathematical representations

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{A_d}\boldsymbol{x}_k + \boldsymbol{B_d}\boldsymbol{u}_k + \boldsymbol{w}_k, & \boldsymbol{w}_k &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q_d}) \\ \boldsymbol{z}_k &= \boldsymbol{C_d}\boldsymbol{x}_k + \boldsymbol{\epsilon}_k, & \boldsymbol{\epsilon}_k &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R_d}) \\ \boldsymbol{x}_0 &= \mathcal{N}(\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0) \end{aligned} \tag{3.3}$$

The KF algorithm is described by the prediction step and correction step, given in (3.4) and (3.5), respectively.

Prediction Step

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1}^{-} &= \boldsymbol{A}_{d,k}\hat{\boldsymbol{x}}_k + \boldsymbol{B}_{d,k}\boldsymbol{u}_k \\ \boldsymbol{P}_{k+1}^{-} &= \boldsymbol{A}_{d,k}\boldsymbol{P}_k\boldsymbol{A}_{d,k}^T + \boldsymbol{Q}_{d,k} \end{aligned} \tag{3.4}$$

Update Step

$$\begin{aligned} \boldsymbol{K}_k &= \boldsymbol{P}_k^{-}\boldsymbol{C}_{d,k}^T(\boldsymbol{C}_{d,k}\boldsymbol{P}_k^{-}\boldsymbol{C}_{d,k}^T + \boldsymbol{R}_{d,k})^{-1} \\ \hat{\boldsymbol{y}}_k &= \boldsymbol{C}_{d,k}\hat{\boldsymbol{x}}_k^{-} \\ \hat{\boldsymbol{x}}_k^{+} &= \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k) \\ \boldsymbol{P}_k^{+} &= (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_{d,k})\boldsymbol{P}_k^{-}(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_{d,k})^T + \boldsymbol{K}[k]\boldsymbol{R}_{d,k}\boldsymbol{K}_k^T \end{aligned} \tag{3.5}$$

3.1.2 Extended Kalman Filter

The EKF is an alternative when the Markov or likelihood models are nonlinear. It solves the problem by linearising the nonlinearities about the most recent estimates before calculating the same equations as for the traditional KF. The nonlinear Markov process and likelihood model are defined as

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \mathrm{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{w}_k \\ \boldsymbol{z}_{k+1} &= \mathrm{h}(\boldsymbol{x}_{k+1}) + \boldsymbol{\epsilon}_k \end{aligned} \tag{3.6}$$

and linearised according to

$$
\begin{aligned}
\boldsymbol{A}_k(\hat{\boldsymbol{x}}_k) &= \frac{\partial}{\partial \boldsymbol{x}_k}\mathrm{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)|_{\boldsymbol{x}_k=\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k=\boldsymbol{u}_k} \\
\boldsymbol{H}_k(\hat{\boldsymbol{x}}_k) &= \frac{\partial}{\partial \boldsymbol{x}_k}\mathrm{h}(\boldsymbol{x}_k)|_{\boldsymbol{x}_k=\hat{\boldsymbol{x}}_k}
\end{aligned}
\tag{3.7}
$$

3.1.3 Multiplicative and Error State Kalman Filter

The ESKF differs from the KF and the EKF by instead of estimating the states directly, they are estimated indirectly through their error terms. Therefore, three new states are defined; the true state, $\boldsymbol{x}^n_{nb}$, the nominal state, $\boldsymbol{x}^n_{nb,nom}$, and the error state, $\boldsymbol{\delta x}$, stated in (3.8). These states are related by (3.9), where $\oplus$ are addition for all states except the attitude, which is the Hamiltonian product $\otimes$ defined in (2.9).

$$
\boldsymbol{x}^n_{nb} = \begin{bmatrix} \boldsymbol{p}^n_{nb} \\ \boldsymbol{v}^n_{nb} \\ \boldsymbol{q}_{nb} \\ \boldsymbol{b}^b_{acc} \\ \boldsymbol{b}^b_{ars} \end{bmatrix}, \boldsymbol{x}^n_{nb,nom} = \begin{bmatrix} \boldsymbol{p}^n_{nb,nom} \\ \boldsymbol{v}^n_{nb,nom} \\ \boldsymbol{q}_{nb,nom} \\ \boldsymbol{b}^b_{acc,nom} \\ \boldsymbol{b}^b_{ars,nom} \end{bmatrix}, \boldsymbol{\delta x} = \begin{bmatrix} \boldsymbol{\delta p} \\ \boldsymbol{\delta v} \\ \boldsymbol{\delta a} \\ \boldsymbol{\delta b}_{acc} \\ \boldsymbol{\delta b}_{ars} \end{bmatrix}
\tag{3.8}
$$

$$
\boldsymbol{x}^n_{nb} = \boldsymbol{x}^n_{nb,nom} \oplus \boldsymbol{\delta x} = \begin{bmatrix} \boldsymbol{p}^n_{nb,nom} + \delta \boldsymbol{p} \\ \boldsymbol{v}^n_{nb,nom} + \delta \boldsymbol{v} \\ \boldsymbol{q}_{nb,nom} \otimes \boldsymbol{\delta q}(\boldsymbol{\delta a}) \\ \boldsymbol{b}^b_{acc,nom} + \delta \boldsymbol{b}_{acc} \\ \boldsymbol{b}^b_{ars,nom} + \delta \boldsymbol{b}_{ars} \end{bmatrix}
\tag{3.9}
$$

MEKF is a specific version of the ESKF to avoid various limitations. In the MEKF, the attitude is parameterised by a four-dimensional unit quaternion in the true and nominal state, $\boldsymbol{q}$, While the attitude error is uniquely defined by a three-parameter minimal representation, $\boldsymbol{\delta a}$. Various representations of $\boldsymbol{\delta a}$ exist. This thesis takes on the Modified Rodriguez Parameters (MRP). The relations between $\boldsymbol{\delta q}$, $\boldsymbol{\delta a}$ and the MRP representation are given by

$$
\boldsymbol{\delta q} = \begin{bmatrix} \delta\eta \\ \boldsymbol{\delta\epsilon} \end{bmatrix}
\tag{3.10a}
$$

$$
\boldsymbol{\delta a} = 4\frac{\boldsymbol{\delta\epsilon}}{1+\delta\eta}
\tag{3.10b}
$$

$$
\boldsymbol{\delta q} = \frac{1}{16+\boldsymbol{\delta a}^T\boldsymbol{\delta a}}\begin{bmatrix} 16-\boldsymbol{\delta a}^T\boldsymbol{\delta a} \\ 8\boldsymbol{\delta a} \end{bmatrix}
\tag{3.10c}
$$

The MEKF estimate the error state, $\boldsymbol{\delta x}$. The true states are then found by the injection process, which is computing (3.9). Additionally, the MEKF is split up into the prediction and update steps similar to traditional KF but with a slight twist. The nominal state is the output from the strapdown equations, where the noise is excluded, i.e. assuming the noise equals zero, as in (3.12). The error state is then computed in the update step by minimising the error between the nominal state and the new measurement. The flow of the MEKF is depicted in Figure 3.1.

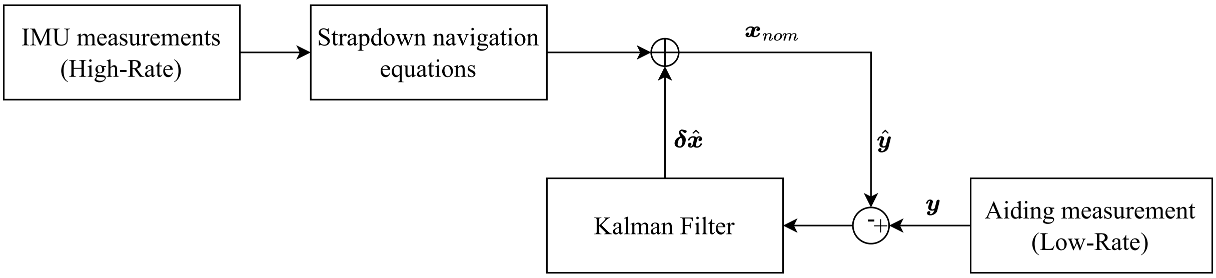

Figure 3.1: Simple flow diagram of the ESKF algorithm. Inspired by [2, p. 477].

Since the error state is to be estimated, it is necessary to compute their kinematics. These are non-linear equations linearised by first-order approximation presented by Sola [5]. The linear dynamics are defined in (3.11). The function S is the skew matrix conversion stated in (2.13), and $\boldsymbol{I}_3$ is the 3x3 identity matrix. Also. the vector $\boldsymbol{w}^b$ is the process noise vector for the error state system model, and $\boldsymbol{\epsilon}^b$ is the noise vector for the measurement model.

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}^n_{nb})\delta\boldsymbol{x} + \boldsymbol{E}(\boldsymbol{x}^n_{nb})\boldsymbol{w}^b \tag{3.11a}$$

$$\boldsymbol{z}^n_{nb} = \boldsymbol{C}\boldsymbol{x}^n_{nb} + \boldsymbol{\epsilon}^b \tag{3.11b}$$

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_3 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{R}_{nb}(\boldsymbol{q}_{nb})\mathrm{S}(\boldsymbol{f}^b_{imu} - \boldsymbol{b}^b_{acc}) & -\boldsymbol{R}_{nb}(\boldsymbol{q}_{nb}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\mathrm{S}(\boldsymbol{\omega}^b_{imu} - \boldsymbol{b}^b_{ars}) & \boldsymbol{0} & -\boldsymbol{I}_3 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\frac{1}{T_{acc}}\boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\frac{1}{T_{ars}}\boldsymbol{I}_3 \end{bmatrix} \tag{3.11c}$$

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{R}_{nb}(\boldsymbol{q}_{nb}) & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I}_3 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_3 \end{bmatrix} \tag{3.11d}$$

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0}_{3x12} \end{bmatrix} \tag{3.11e}$$

Prediction step

The state and covariance are propagated using the strapdown navigation equations. These are initialised for the first iteration with $x(0)$ and $P(0)$, followed by recursive predictions using the estimated state. These are the nominal states in (3.12), where the noise is assumed to be zero. Also, the IMU measurements are assumed to have been transformed to $\mathcal{F}_b$ by using a rotation matrix R_{bm}. Finally, the covariance matrix is predicted according to (3.13).

$$\begin{aligned}
\dot{\boldsymbol{p}}^n_{nb,nom} &= \boldsymbol{v}^n_{nb,nom} \\
\dot{\boldsymbol{v}}^n_{nb,nom} &= \boldsymbol{R}_{nb}(\boldsymbol{f}^b_{imu} - \boldsymbol{b}^b_{acc,nom}) + \boldsymbol{g}^n_b(\boldsymbol{p}^n_{nb,nom}) \\
\dot{\boldsymbol{q}}_{nb,nom} &= \frac{1}{2}\boldsymbol{q}_{nb,nom} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}^b_{imu} - \boldsymbol{b}^b_{ars,nom} \end{bmatrix} \\
\dot{\boldsymbol{b}}^b_{acc,nom} &= 0 \\
\dot{\boldsymbol{b}}^b_{ars,nom} &= 0
\end{aligned} \tag{3.12}$$

$$\boldsymbol{P}^-_{k+1} = \boldsymbol{A}_{d,k}\boldsymbol{P}_k\boldsymbol{A}^T_{d,k} + \boldsymbol{E}_{d,k}\boldsymbol{Q}_{d,k}\boldsymbol{E}^T_{d,k} \tag{3.13}$$

Update step

Based on the predictions, the error state and covariance are estimated through correction according to

$$\begin{aligned}
\boldsymbol{K}_k &= \boldsymbol{P}^-_k\boldsymbol{C}^T_{d,k}(\boldsymbol{C}_{d,k}\boldsymbol{P}^-_k\boldsymbol{C}^T_{d,k} + \boldsymbol{R}_{d,k})^{-1} \\
\delta\hat{\boldsymbol{x}}_k &= \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{nom,k}) \\
\boldsymbol{P}^+_k &= (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_{d,k})\boldsymbol{P}^-_k(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_{d,k})^T + \boldsymbol{K}_k\boldsymbol{R}_{d,k}\boldsymbol{K}^T_k
\end{aligned} \tag{3.14}$$

Injection and reset step

Lastly, the true states are computed by injecting the error state into the INSs state

$$\boldsymbol{x}^n_{nb,k} = \boldsymbol{x}^n_{nb,nom,k} \oplus \boldsymbol{\delta x}_k \tag{3.15}$$

transforming the estimation error covariance

$$\boldsymbol{P}^+_k = \boldsymbol{G}\boldsymbol{P}^+_k\boldsymbol{G} \tag{3.16}$$

and then resetting the MEKF

$$\boldsymbol{\delta x} = \boldsymbol{0}. \tag{3.17}$$

3.1.4 Outlier rejection

A valuable step in estimation schemes is to pass the measurements through a validation gate. This is essential to avoid degradation, especially for noisy measurements or data affected by multipath propagation. Measurements that land outside a validation gate are defined as outliers and not processed in the estimation scheme. To determine whether a measurement is an inlier, a test is computed

$$(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k)^T\left(\boldsymbol{C}_d\boldsymbol{P}^-_k\boldsymbol{C}^T_d + \boldsymbol{R}_d\right)^{-1}(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k) \le \chi^2_{\alpha,\eta_y} \tag{3.18}$$

and if the test holds, the measurements are an inlier. If not, the measurement is discarded as an outlier. χ^2_{α,η_y} is the gate threshold chosen based on the confident interval $1-\alpha$ and the degrees of freedom, $\eta_y = 3$, which for the MEKF is three. This validation gate is also referred to as the Normalised Innovation Squared (NIS), [21, 22]

3.2 Factor Graph Optimisation on the Manifold

FG state estimation utilises the FG structure and its inference techniques to obtain estimates of the states. Again, the estimation problem is to compute The MAP formulated in (2.40), and helpfully, the global nonlinear function of the FG, $\phi(x)$, is equivalent to the posterior, which is maximised according to

$$\arg\max_{\mathcal{X}_{1:k}} \phi(\mathcal{X}_{1:k}) = \arg\max_{\mathcal{X}_{1:k}} p(\mathcal{X}_{1:k}|z_{1:k}) = \arg\max_{\mathcal{X}_{1:k}} p(z_{1:k}|\mathcal{X}_{1:k})p(\mathcal{X}_{1:k}) \tag{3.19}$$

Applying the structure of the FG the resulting a posteriori equals

$$\arg\max_{\mathcal{X}_{1:k}} \phi(\mathcal{X}_{1:k}) = \arg\max_{\mathcal{X}_{1:k}} \prod_k \phi_k(\mathcal{X}_k) = p(\mathcal{X}_1)l(\mathcal{X}_1; z_1) \prod_{k>1} p(\mathcal{X}_k|\mathcal{X}_{k-1})l(\mathcal{X}_k; z_k) \tag{3.20}$$

where $\mathcal{X} \in \mathcal{M}$.

To solve the estimation problem, the likelihood model, $l(\mathcal{X}; z)$, and the system model, $p(\mathcal{X}_i|\mathcal{X}_{i-1})$, are formulated as factors. The system model is split into two factors; the IMU and the bias factor. The likelihood model depends on the type of measurement, for example, the GNSS factor. The high-rate IMU measurements are used to predict and create one preintegration IMU factor. This factor act as a constraint between the two low-rate GNSS measurements, visualised in Figure 3.3. An FG is depicted in Figure 3.2 with three states and seven factors. One GNSS factor for each state and one bias and IMU factor between the two states.

The MAP problem is solved as an LSO problem. The following equations are formulated for a smoothing problem, but it is trivial to turn it into a filtering or fixed-lag smoothing problem by marginalising. This section derives its content from Foster et al. [11]. Finally, to clarify any misunderstandings, the variable k in this section describe all nodes in the FG, i.e. not including the predictions. This differs from the MEKF formulation where k is used for both predictions and corrections. This is natural since the predictions are accumulated into one preintegration factor for the optimisation problem. The prediction step is described by Δt.

3.2.1 Preintegration modelling and IMU factor

The IMU factor, depicted in Figure 3.2, is similar to the prediction step in the KF. The factor comprises a so-called preintegration model, which builds on the nonlinear manifold strapdown equations in (2.72), which are compactly written as

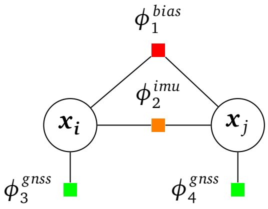

Figure 3.2: Example of a factor graph with two nodes and four factors. Including a bias constraint, a preintegration constraint and two GNSS factor constraints.

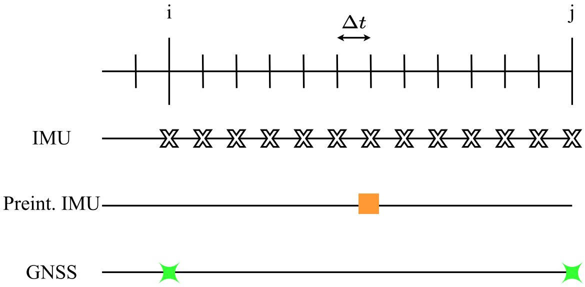

Figure 3.3: Visualisation of rate difference between IMU and GNSS and the accumulation of the preintegrated IMU factor. Inspired by Forster et al. [11].

$$\mathcal{X}_{i+\Delta t} \sim \mathcal{N}(\boldsymbol{f}(\mathcal{X}_i), \boldsymbol{Q}) \tag{3.21a}$$

$$\mathcal{X}_{i+\Delta t} = \mathrm{f}(\mathcal{X}_i) \oplus \boldsymbol{w}, \qquad \boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}) \in \mathcal{T}\mathcal{M}_{\mathcal{X}_i} \tag{3.21b}$$

The preintegration model predicts the states using the strapdown equations with IMU measurements as input for each timestamp measurement arrives. Then, when the correction measurement arrives, the preintegration model accumulates all these propagations into one IMU factor from timestep $t = i$ to timestep $t = j$, depicted in Figure 3.3. The preintegration is computed by applying Euler integration of the states excluding the biases as

$$\boldsymbol{R}_{i+\Delta t} = \boldsymbol{R}_i \mathrm{Exp}(\boldsymbol{\omega}_{imu,i} \Delta t) \tag{3.22a}$$

$$\boldsymbol{v}_{i+\Delta t} = \boldsymbol{v}_i + \boldsymbol{a}_{imu,i} \Delta t \tag{3.22b}$$

$$\boldsymbol{p}_{i+\Delta t} = \boldsymbol{p}_i + \boldsymbol{v}_i \Delta t + \frac{1}{2} \boldsymbol{a}_{imu,i} \Delta t^2 \tag{3.22c}$$

The integration is computed on the manifold, an essential part of the factor graph preintegration model. The predictions are accumulated by taking the product of attitude integrations and the sum for position and velocity integrations as

$$\boldsymbol{R}_j = \boldsymbol{R}_i \prod_{t=i}^{j-1} \mathrm{Exp}(\boldsymbol{\omega}_{imu,t} \Delta t) \tag{3.23a}$$

$$\boldsymbol{v}_j = \boldsymbol{v}_i + \sum_{t=i}^{j-1} \boldsymbol{R}_t \boldsymbol{a}_{imu,t} \Delta t \tag{3.23b}$$

$$\boldsymbol{p}_j = \boldsymbol{p}_i + \sum_{t=i}^{j-1} \boldsymbol{v}_t \Delta t + \sum_{t=i}^{j-1} \frac{1}{2} \boldsymbol{a}_{imu,t} \Delta t^2 \tag{3.23c}$$

Introducing $\Delta \boldsymbol{R}_{ij} = \boldsymbol{R}_i^T \boldsymbol{R}_j$, $\boldsymbol{v}_j = \boldsymbol{v}_i + \Delta \boldsymbol{v}_{ij}$ and $\boldsymbol{p}_j = \boldsymbol{p}_i + \Delta \boldsymbol{p}_{ij}$, isolating the noise and reformulating (3.23) yield

$$\Delta \boldsymbol{R}_{ij} = \boldsymbol{R}_i^T \boldsymbol{R}_j \mathrm{Exp}\left(\boldsymbol{\delta\phi}_{ij}\right) \tag{3.24a}$$

$$\Delta \boldsymbol{v}_{ij} = \boldsymbol{R}_i^T \left(\boldsymbol{v}_j - \boldsymbol{v}_i - \boldsymbol{g}\Delta t_{ij}\right) + \boldsymbol{\delta v}_{ij} \tag{3.24b}$$

$$\Delta \boldsymbol{p}_{ij} = \boldsymbol{R}_i^T \left(\boldsymbol{p}_j - \boldsymbol{p}_i - \boldsymbol{v}_i \Delta t_{ij} - \frac{1}{2}\boldsymbol{g}\Delta t_{ij}^2\right) + \boldsymbol{\delta p}_{ij} \tag{3.24c}$$

where the noise is defined according to

$$[\boldsymbol{\delta\phi}, \boldsymbol{\delta v}, \boldsymbol{\delta p}] \sim \mathcal{N}\left(\boldsymbol{0}_{9x1}, \Sigma_{preint,ij}\right) \tag{3.25}$$

(3.24) is a function of the to-be estimated state, but the advantage of this formulation is revealed when formulating the residuals for the LSO problem. Furthermore, the formulation assumes that the bias is given. The bias is incorporated using a bias update $\boldsymbol{b} = \bar{\boldsymbol{b}} + \boldsymbol{\delta b}$, where $\bar{\boldsymbol{b}}$ is the given constant bias, while $\boldsymbol{\delta b}$ is the change in bias. The bias is incorporated by updating (3.24) to

$$\Delta \boldsymbol{R}_{ij}\left(\boldsymbol{b}_{ars,i}\right) \simeq \Delta \boldsymbol{R}_{ij}\left(\bar{\boldsymbol{b}}_{ars,i}\right) \mathrm{Exp}\left(\frac{\delta \Delta \boldsymbol{R}_{ij}}{\delta \boldsymbol{b}_{ars}} \boldsymbol{\delta b}_{ars}\right) \tag{3.26a}$$

$$\Delta \boldsymbol{v}_{ij}\left(\boldsymbol{b}_{ars,i}, \boldsymbol{b}_{acc,i}\right) \simeq \Delta \boldsymbol{v}_{ij}\left(\bar{\boldsymbol{b}}_{ars,i}, \bar{\boldsymbol{b}}_{acc,i}\right) + \frac{\delta \Delta \boldsymbol{v}_{ij}}{\delta \boldsymbol{b}_{ars}} \boldsymbol{\delta b}_{ars,i} + \frac{\delta \Delta \boldsymbol{v}_{ij}}{\delta \boldsymbol{b}_{acc}} \boldsymbol{\delta b}_{acc,i} \tag{3.26b}$$

$$\Delta \boldsymbol{p}_{ij}\left(\boldsymbol{b}_{ars,i}, \boldsymbol{b}_{acc,i}\right) \simeq \Delta \boldsymbol{p}_{ij}\left(\bar{\boldsymbol{b}}_{ars,i}, \bar{\boldsymbol{b}}_{acc,i}\right) + \frac{\delta \Delta \boldsymbol{p}_{ij}}{\delta \boldsymbol{b}_{ars}} \boldsymbol{\delta b}_{ars,i} + \frac{\delta \Delta \boldsymbol{p}_{ij}}{\delta \boldsymbol{b}_{acc}} \boldsymbol{\delta b}_{acc,i} \tag{3.26c}$$

The Markov model can be formulated with the accumulated preintegration formulations, (3.26). Expanding on the Markov model from (3.20) using (3.21) together with (2.34) yields

$$\begin{aligned}
p(\boldsymbol{\mathcal{X}}_{1:k}) &= p(\boldsymbol{\mathcal{X}}_1) p(\boldsymbol{\mathcal{X}}_2 | \boldsymbol{\mathcal{X}}_1) p(\boldsymbol{\mathcal{X}}_k | \boldsymbol{\mathcal{X}}_{k-1}) && (3.27a)\\
&= \mathcal{N}(\boldsymbol{\mathcal{X}}_1, \boldsymbol{Q}) \mathcal{N}(f(\boldsymbol{\mathcal{X}}_1), \boldsymbol{Q}) \mathcal{N}(f(\boldsymbol{\mathcal{X}}_{k-1}), \boldsymbol{Q}) && (3.27b)\\
&= \mathcal{N}(\boldsymbol{\mathcal{X}}_1, \boldsymbol{Q}) \prod_{i=2}^{k-1} \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{Q}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{\mathcal{X}}_j - f(\boldsymbol{\mathcal{X}}_i))^T \boldsymbol{Q}^{-1} (\boldsymbol{\mathcal{X}}_j - f(\boldsymbol{\mathcal{X}}_i))\right) && (3.27c)\\
&= \mathcal{N}(\boldsymbol{\mathcal{X}}_1, \boldsymbol{Q}_1) \prod_{i=2}^{k-1} \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_{preint,ij}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \Delta \boldsymbol{\mathcal{X}}_{ij}^T \boldsymbol{\Sigma}_{preint,ij}^{-1} \Delta \boldsymbol{\mathcal{X}}_{ij}\right) && (3.27d)
\end{aligned}$$

where $\Delta \boldsymbol{\mathcal{X}}_{ij} = [\Delta \boldsymbol{p}_{ij}, \Delta \boldsymbol{v}_{ij}, \Delta \boldsymbol{R}_{ij}] = \boldsymbol{r}_x$ is the residual.

3.2.2 Bias factor

The bias factor accounts for the bias elements in the strapdown equations (2.72), visualised in Figure 3.2. The biases are slowly time-varying quantities and are therefore modelled as a Wiener process

$$\dot{\boldsymbol{b}}_{acc} = \boldsymbol{w}_{acc}, \qquad \boldsymbol{w}_{acc} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{acc}) \tag{3.28a}$$

$$\dot{\boldsymbol{b}}_{ars} = \boldsymbol{w}_{ars}, \qquad \boldsymbol{w}_{ars} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{ars}) \tag{3.28b}$$

Integrating the biases over the time interval $[t = i, t = j]$ yield

$$\boldsymbol{b}_{acc,j} = \boldsymbol{b}_{acc,i} + \boldsymbol{w}_{acc} \tag{3.29a}$$

$$\boldsymbol{b}_{ars,j} = \boldsymbol{b}_{ars,i} + \boldsymbol{w}_{ars} \tag{3.29b}$$

The resulting model for the bias factor is equivalent to the IMU factor given as

$$\begin{aligned} p\left(\boldsymbol{b}_{1:k}\right) \propto & \mathcal{N}\left(\boldsymbol{b}_1, \boldsymbol{\Sigma}_{bias}\right) \times \\ & \prod_{i=2}^{k-1} \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_{bias}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\boldsymbol{b}_j - \boldsymbol{b}_i\right)^T \boldsymbol{\Sigma}_{bias}^{-1}\left(\boldsymbol{b}_j - \boldsymbol{b}_i\right)\right) \end{aligned} \tag{3.30}$$

where $\boldsymbol{b} = [\boldsymbol{b}_{acc}, \boldsymbol{b}_{ars}]^T$, $\boldsymbol{\Sigma}_{bias} = \text{diag}[\boldsymbol{\Sigma}_{acc}, \boldsymbol{\Sigma}_{ars}]$ and the residual defined as $\boldsymbol{r}_b = \boldsymbol{b}_j - \boldsymbol{b}_i$.

3.2.3 GNSS factor

Each node has a GNSS factor inferring knowledge of the pose, as in Figure 3.2. The GNSS measurement is modelled as

$$\boldsymbol{z} \sim \mathcal{N}(h(\boldsymbol{\mathcal{X}}), \boldsymbol{R}) \tag{3.31a}$$

$$\boldsymbol{z} = h(\boldsymbol{\mathcal{X}}) + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{R}) \tag{3.31b}$$

Assuming Gaussian distribution, and since the noise is white, the measurements are independent. This result in the likelihood model, $p(\boldsymbol{z}_{1:k}|\boldsymbol{x}_{1:k})$, formulated as a product of multivariate Gaussian distributions

$$p(\boldsymbol{z}_{1:k}|\boldsymbol{\mathcal{X}}_{1:k}) = p(\boldsymbol{z}_1)p(\boldsymbol{z}_2)p(\boldsymbol{z}_k) \tag{3.32a}$$

$$= \mathcal{N}(h(\boldsymbol{\mathcal{X}}_1), \boldsymbol{R})\mathcal{N}(h(\boldsymbol{\mathcal{X}}_2), \boldsymbol{R})\mathcal{N}(h(\boldsymbol{\mathcal{X}}_k), \boldsymbol{R}) \tag{3.32b}$$

$$= \prod_{i=1}^{k} \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{R}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\boldsymbol{z}_i - h(\boldsymbol{\mathcal{X}}_i)\right)^T \boldsymbol{R}^{-1}\left(\boldsymbol{z}_i - h(\boldsymbol{\mathcal{X}}_i)\right)\right) \tag{3.32c}$$

with the residual $\boldsymbol{r}_z = \boldsymbol{z} - h(\boldsymbol{\mathcal{X}})$.

3.2.4 Nonlinear least square formulation

Combining (3.27d) , (3.30) and (3.32c) result in the MAP formulation

$$\begin{aligned} \boldsymbol{\mathcal{X}}_{1:k}^{*} = & \arg\max_{\boldsymbol{\mathcal{X}}_{1:k}} p(\boldsymbol{z}_{1:k}|\boldsymbol{\mathcal{X}}_{1:k}) p(\boldsymbol{\mathcal{X}}_{1:k}) = \arg\max_{\boldsymbol{\mathcal{X}}} \mathcal{N}(\boldsymbol{\mathcal{X}}_1, \boldsymbol{Q}) \mathcal{N}(\boldsymbol{b}_1, \boldsymbol{\Sigma}_{bias}) \times \\ & \prod_{i=2}^{k-1} \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_{preint,ij}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \Delta\boldsymbol{\mathcal{X}}_{ij}^T \boldsymbol{\Sigma}_{preint,ij}^{-1} \Delta\boldsymbol{\mathcal{X}}_{ij}\right) \times \\ & \prod_{i=2}^{k-1} \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_{bias}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\boldsymbol{b}_j - \boldsymbol{b}_i\right)^T \boldsymbol{\Sigma}_{bias}^{-1}\left(\boldsymbol{b}_j - \boldsymbol{b}_i\right)\right) \times \\ & \prod_{i=1}^{k} \frac{1}{(2\pi)^{\frac{n}{2}} |\boldsymbol{R}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\boldsymbol{z}_i - h(\boldsymbol{\mathcal{X}}_i)\right)^T \boldsymbol{R}^{-1}\left(\boldsymbol{z}_i - h(\boldsymbol{\mathcal{X}}_i)\right)\right) \end{aligned} \tag{3.33}$$

Taking the negative log of (3.33) it is possible to reformulate the problem to a LS minimisation problem described by

$$\begin{aligned}\boldsymbol{\mathcal{X}}^*_{1:k} =& \arg\min_{\boldsymbol{\mathcal{X}}_{1:k}} p(\boldsymbol{z}_{1:k}|\boldsymbol{\mathcal{X}}_{1:k})p(\boldsymbol{\mathcal{X}}_{1:k}) \propto \\ & \arg\min_{\boldsymbol{\mathcal{X}}_{1:k}}\Bigg[-\mathcal{N}(\boldsymbol{\mathcal{X}}_1,\boldsymbol{Q}) - \mathcal{N}(\boldsymbol{b}_1,\boldsymbol{\Sigma}_{bias}) - \\ & \sum_{i=2}^{k-1}\Big(\frac{1}{2}(\Delta\boldsymbol{\mathcal{X}}_{ij})^T\boldsymbol{\Sigma}^{-1}_{preint,ij}(\Delta\boldsymbol{\mathcal{X}}_{ij})\Big) - \\ & \sum_{i=2}^{k-1}\Big(\frac{1}{2}(\boldsymbol{b}_j-\boldsymbol{b}_i)^T\boldsymbol{\Sigma}^{-1}_{bias}(\boldsymbol{b}_j-\boldsymbol{b}_i)\Big) - \\ & \sum_{i=1}^{k}\Big(\frac{1}{2}(\boldsymbol{z}_i - h(\boldsymbol{\mathcal{X}}_i))^T\boldsymbol{R}^{-1}(\boldsymbol{z}_i - h(\boldsymbol{\mathcal{X}}_i))\Big)\Bigg]\end{aligned} \tag{3.34}$$

Reformulating and comprising the problem by introducing the *squared Mahalanobis norm*, $||.||^2_{\Sigma}$, [23, Ch. 1.4], the solution to the smoothing problem is proportional to,

$$\boldsymbol{\mathcal{X}}^*_{1:k} = \arg\min_{\boldsymbol{\mathcal{X}}_{1:k}}\left[\sum_{i=2}^{k-1}\left|\left|\Delta\boldsymbol{\mathcal{X}}_{ij}\right|\right|^2_{\Sigma_{preint,ij}} + \sum_{i=2}^{k-1}\left|\left|\boldsymbol{b}_j-\boldsymbol{b}_i\right|\right|^2_{\Sigma_{bias}} + \sum_{i=1}^{k}||\boldsymbol{z}_i - h(\boldsymbol{\mathcal{X}}_i)||^2_{\boldsymbol{R}}\right] \tag{3.35}$$

which is initialised with the prior information on the states and bias, $\mathcal{N}(\boldsymbol{\mathcal{X}}_1,\boldsymbol{Q})$,$\mathcal{N}(\boldsymbol{b}_1,\boldsymbol{\Sigma}_{bias})$. The optimal states in the least squares problem are found by computing $\boldsymbol{x}$ for each time step such that the three residuals, $\boldsymbol{r}_x,\boldsymbol{r}_b,\boldsymbol{r}_z$, are minimised. This is achieved by following the methodology in Section 2.6.2.

The LSO problem in (3.35) is a smoothing problem. To perform fixed-lag smoothing or filtering only a subset of the states are optimised. Removing the uninteresting states directly leads to overconfidence in the system by trusting the estimations of the removed states, therefore, it is necessary to marginalise out the states.

3.3 Tuning

The tuning parameters are similar for the MEKF and FGO schemes, but the FG has one additional adjustable variable. These should be tuned using sensory data and expected or measured uncertainty to the extent that it is possible.

Prior

Both schemes have prior information on the states with a tuned confidence. The variances stated in Table 3.1 is gathered on the diagonal of the covariance matrix $\boldsymbol{P}_0$.

Prediction

The Markov model for the KF and the preintegration model in FGO has the same tuning variables stated in Table 3.2. These are joined on the diagonal of the matrix $\boldsymbol{Q}$.

Table 3.1: Prior tunable variables for the MEKF and FGO algorithms.

Variable	Definition	Unit
$\boldsymbol{\sigma}_{p_0}$	Uncertainty on prior position.	$[\mathrm{m}]$
$\boldsymbol{\sigma}_{v_0}$	Uncertainty on prior velocity.	$\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$
$\boldsymbol{\sigma}_{\Theta_0}$	Uncertainty on prior attitude.	$[\mathrm{rad}]$
$\boldsymbol{\sigma}_{b_0^{acc}}$	Uncertainty on prior accelerometer bias.	$\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$
$\boldsymbol{\sigma}_{b_0^{ars}}$	Uncertainty on prior gyroscope bias.	$\left[\frac{\mathrm{rad}}{\mathrm{s}}\right]$

Table 3.2: Tunable variable in prediction for the MEKF and FGO algorithms.

Variable	Definition	Unit
$\boldsymbol{\sigma}_{q^{ars}}$	Measurement uncertainty in the gyroscope.	$\left[\frac{\mathrm{rad}}{\mathrm{s}}\frac{1}{\mathrm{Hz}}\right]$
$\boldsymbol{\sigma}_{q_{bias}^{ars}}$	Uncertainty in the gyroscope bias.	$\left[\frac{\mathrm{rad}}{\mathrm{s}^{1.5}}\right]$
$\boldsymbol{\sigma}_{q^{acc}}$	Measurement uncertainty in the accelerometer.	$\left[\frac{\mathrm{m}}{\mathrm{s}^2}\frac{1}{\mathrm{Hz}^{0.5}}\right]$
$\boldsymbol{\sigma}_{q_{bias}^{acc}}$	Uncertainty in the accelerometer bias.	$\left[\frac{\mathrm{m}}{\mathrm{s}^{2.5}}\right]$
$\boldsymbol{\sigma}_{int}^{*}$	Uncertainty associated with the integration of the states.	$\left[\frac{\mathrm{m}}{\mathrm{s}\,\mathrm{s}^{0.5}}\right]$

* Only for FG.

Correction

Both schemes have a correction step with the tuning matrix $\boldsymbol{R}$, encompassing the uncertainty in the position correction measurements, $\boldsymbol{\epsilon}$, in meters.

Chapter 4

Related Work

This chapter delves into a literature review of relevant work accomplished in the field of KF and FGO state estimation with a focus on GNSS and PARS-aided INS. A complete summary of research articles reviewed in this thesis is listed in Appendix A.

4.1 Kalman Filtering with GNSS-aided INS

The traditional KF is extensively used today but is often not applicable because of its linear assumptions. However, it is one of the few estimation techniques considered solved, as it will always find the optimal solutions if its assumptions are satisfied, [24]. The EKF relax the linear requirements; however, the computed Kalman Gain is now only an approximation and therefore, the EKF is only suboptimal compared to the traditional KF. The filter's accuracy consequently depends on the extent of nonlinearity in the system or measurement model. For excessively nonlinear systems, the accuracy will decay, [24]. Even though the EKF is reported to be the leading filter because of its efficient computation compared to potentially more accurate filters such as unscented KF or particle filters, [25]. Several examples of sufficiently accurate KF and EKF for GNSS-aided INS integration exist [26–31].

The ESKF is a natural extension of the EKF to improve on the main limitation, linearisation. The system operates close to the origin of the linear error state kinematics, increasing accuracy and ensuring that parameter singularities, such as gimbal lock, are avoided. Furthermore, because of the small perturbations in the error-state system, second-order products are negligible, resulting in efficient and fast computations, [5, 32]. This makes the ESKF robust and several articles show successful and competitive integration of GNSS-aided INS using ESKF, [32–34].

MEKF specify the usage of a quaternion as the attitude representation, specifically a minimal representation of the error attitude, for example, the MRP representation. Comprehensive research has been completed on this topic. F. Landis Marley reported the importance of using quaternions instead of other representations, such as Euler anglesand rotation matrices, in 1982. Furthermore, Marley posted an article assessing this topic specifically in 2002. Both articles also conclude that using the four dimension quaternion yields problems because of normalisation issues and singularities in the covariance matrix. Therefore, a minimal three-dimensional representation is advantageous, [31, 35].

Research relating to Lie theory in sensor fusion has increased in recent years. As described in Section 2.4, the to-be estimated pose lies on the manifold; therefore, any perturbing should be computed accordingly. However, most KF schemes assume to act on Euclidean space. Some EKFs on the manifold has been developed and found to be successful, yielding better performance compared to state-of-the-art estimation techniques, [36, 37]. The ESKF manages to reduce the effect of this assumption because of the small perturbation in the error states, which ensures that excluding Lie Algebra has a minor impact. However, taking Lie Theory into account is expected to yield higher accuracy, according to Sola [5].

4.2 Factor Graph with GNSS-aided INS

FGO for state estimation has grown in popularity since its introduction in SLAM algorithms. Dellaert and Kaess demonstrated in 2006 that FG-SLAM yields better accuracy than EKF-SLAM, [6]. Since then, numerous articles have been published on the topic. In 2012 Indelman et al. presented one of the first formulations of an FG framework for integrating INS combined with other external measurements. They compared the FG framework to a conventional EKF and found the FG to be a highly competitive algorithm. Focusing on asynchronous measurements, delayed measurements and adaptive fixed-lag smoothing ensured that the FG framework was considerably more flexible than the EKF, [38]. This work was continued in 2013, looking, more specifically, into comparing various fixed-lag smoothers and their computation time. Displaying longer lags yielded considerably better results, and fixed-lag optimisation gave near-optimal estimation for a fraction of the computation time of complete batch optimisation, [39]. Forster et al. continued the work and formulated a on manifold preintegration model applicable to IMU factors. The IMU integration was combined with visual sensors resulting in a real-time applicable system. This eventually led to the Georgia Tech Smoothing and Mapping (GTSAM) framework, an open-source library released on GitHub in 2019. GTSAM is explained in greater detail in Section 5.1.1, [11].

Sola has also contributed to the estimation field, focusing on manifold applications. Sola et al. produced a comprehensive and detailed article on Lie theory for state estimation in 2021 [10]. This led to the WOLF framework published in 2022 [40]. This library is based on FG and is applicable for a vast number of applications within autonomous systems. On the other hand, the framework is newly released and requires improvements, especially in real-time performance.

Many articles look into combining INS and visual sensors for smoothing problems. Some look into integrating GNSS-aided INS, such as Chang et al. in 2019, which yielded better accuracy and robustness than an EKF. The study focused on outlier rejection, a problem for KF, [41]. Wen et al. compared an FGO framework to an EKF, concluding that FGO provides better accuracy because of more iterations during the optimisation and better time correlation between states, [42]. These studies and others, such as Das et al. and Zhao et al., analyse alternative objective functions to the traditional least-square formulation. Communicated as robust formulations, they provide better handling of outliers and large residuals by, for example, weighting the nodes by their depending on their distance from other estimations, [43, 44]. Additionally, Dellaert and Kaess [23] provide an overview of FGO for robot perception, where they state the importance of using robust loss functions, such as Huber and Turkey loss to accommodate outliers in the optimisation, [23].

4.3 PARS-aided navigation

The UAV lab at NTNU has completed a series of works on PARS-aided navigation. Albrektsen completed a PhD in 2018, including, among others, three articles looking into PARS-aided navigation systems released in the period 2017 and 2018. The first study in 2017 looked into using PARS instead of GNSS and found that 87.89 % of the PARS measurements are within 40m of the GPS-RTK measurements for that flight. A barometer is used for these specific measurements to compensate for the poor vertical measurements of the PARS. This greatly improves vertical estimation, [45]. Albrektsen et al. concluded that PARS is an alternative to GNSS with satisfactory accuracy when using a nonlinear observer for estimation, [46]. However, multipath propagation is a problem, especially in the elevation angle affecting the height measurements. Albrektsen et al. produced drift-free pose estimations with 26.3 RMSE after including a magnetometer, [46]. The study utilised a loosely coupled system and expected a tightly coupled observer to yield higher accuracy, [47].

The work on PARS-aided navigation was continued at the UAV lab, and in 2019 and 2020, Gryte et al. published two articles based on the work of Albrektsen. In addition, Gryte et al. utilised a MEKF for state estimation with GNSS, PARS and a barometer for height measurement. This resulted in submeter height RMSE and sub-twenty meter for North and East RMSE compared to the GNSS-aided INS system, [17, 22].

Mika Okuhara is working on the topic and released two articles in 2021 and 2022 on PARS. Okuhara et al. worked on methods for calibrating the PARS at the start of experiments. This was completed using a MEKF where the orientation between the radio and NED frame is computed. As a result, the orientation of the radio is successfully calculated, which is necessary to use the PARS for navigation, [48, 49]. The various articles are compactly listed in Table A.3 in Appendix A.

NTNU and the UAV lab have the most extensive research on the topic. However, others have also looked into the technology, such as Linkoping University, where Carl Rapp completed a master's thesis looking into various methods to include PARS by combining it with IMU, barometers using both loosely and tightly coupled KF, [50]. Rapp ended up removing the elevation angle measurements from the PARS because of the large amount of multipath propagation, focusing on the 2D position estimations. Rapp found that the IMU and barometer-aided algorithm performed best.

Chapter 5

Methodology, Experimental Setup and Implementation

This chapter provides a detailed account of the process for this thesis, along with an overview of the employed resources. The dataset and experimental setup used in this thesis are also demonstrated. Lastly, implementation details of the state estimation algorithms are provided. Information regarding the written code for this thesis is found in Appendix C.

5.1 Methodology

The work conducted in this master's thesis follows a methodology with seven primary milestones for a structured and systematic investigation of the problem to be addressed:

1. Meta-analysis
2. Theory review
3. Dataset analysis
4. Implementation of MEKF
5. Tuning the MEKF
6. Implementation of FGO
7. Tuning the FGO

The first part of the thesis went into a meta-analysis of existing literature relevant to state estimation, MEKF, FGO and PARS, stating key findings. This is followed by establishing a solid theoretical foundation upon which state estimation builds. Next, a dataset was provided, and some time went into an examination and exploration. Considerable amount of time was spent implementing and understanding the MEKF and FGO concepts. The MEKF was first implemented with GNSS sensor data after a long process of trial and error. The MEKF algorithm was already implemented in the navigation toolbox provided by the UAV lab; however, for a complete understanding of the concepts, the algorithm was implemented from scratch. Next, the MEKF was tuned based on tuning parameters in the navigation toolbox. Then the FGO was implemented with GNSS and tuned. The GNSS data was then switched to include PARS measurements.

5.1.1 Software

Various resources have been utilised during this thesis, precisely two software programs. MAtrix LABratory (MATLAB) is a coding program mainly known for procedure programming and its arrays, making it simple for doing vector and matrix operations, [51]. On the other hand, C++ is another coding program more renowned for its computational speed and objective coding, [52]. This project applied MATLAB to implement the MEKF algorithm, while C++ were used for the FG system.

To supplement the programming languages, various libraries are utilised. The UAV lab provided the navigation toolbox library containing already implemented algorithms and helpful functions for simplifying the implementation in MATLAB. The library GTSAM is used to implement the FG. GTSAM provides a framework for implementing various state estimation and smoothing schemes and can be used for a wide range of applications, [53]. This chose to apply GTSAM since it implements the formulated FGO in Section 3.2. Additionally, it has an extensive user network, great supporting material and is maturely developed.

For simplifying the data management and implementation in C++, the Robot Operating System (ROS) is used. This is a comprehensive framework consisting of a set of libraries. ROS is helpful in building robot applications and provides tools, for example, within communication between programs. However, ROS only runs on Linux operating systems, so the FG system is implemented on Ubuntu, [54, 55].

5.2 Dataset and Experimental setup

The UAV lab at NTNU uses Agdenes Airfield as their primary testing field for UAVs 90 km northwest of Trondheim. On October 8 2020, a field test was carried out at Agdenes. Multiple flights were completed with a Skywalker X8 UAV, and flight 1 is depicted in Figure 5.1, flying over the Trondheim fjord.

The Skywalker X8 UAV is fitted with various systems for autonomous flight. The autopilot system was Pixhawk, running ArduPlane flight control software. Pixhawk utilised a GNSS, a digital compass, a barometric sensor and an INS. In addition to the autopilot, the UAV had a Sensonor STIM 300 IMU, a Ublox F9P-ZED GNSS module and Radionor Communications CRE2 144-LW PARS, [56–58]. Lastly, two CRE189 ground unit antennas were deployed. The UAV PARS position is calculated on the ground station and is then transmitted to the UAV. The Skywalker X8 and a ground antenna are visualised in Figure 5.2.

A SenTiBoard was used to synchronise the timestamps of the IMU and the GNSS measurements. The PARS unit was used for both communication and PARS measurements. The STIM 300 IMU provides data at 50, 100 and 200 Hz, the GNSS unit delivers data at 5 Hz, and the PARS provides data each time the radio is ping-ed, around 5 Hz.

The IMU, GNSS and PARS data gathered from flight 1 are used in this thesis. Additionally, Attitude and Heading Reference System (AHRS) and GNSS-PPK data is used for attitude and positions

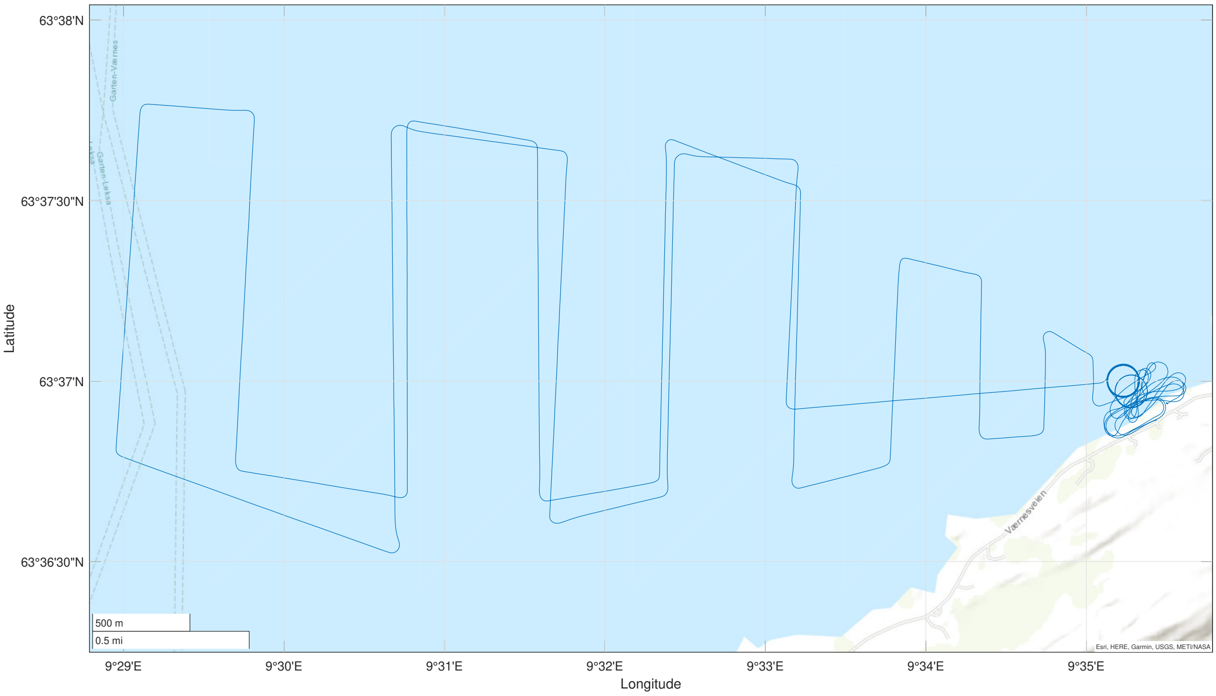

Figure 5.1: Map of the flight 1 trajectory completed on October 8 2020 at Agdenes airfield.

references, respectively. The GNSS data is visually compared against GNSS-PPK measurements in Figure 5.3 and Figure 5.4 in 2D and 3D, respectively. The GNSS is more or less equivalent to the GNSS-PPK data. Figure 5.5 and Figure 5.6 PARS measurements are visualised along with GNSS-PPK data in 2D and 3D, respectively. The PARS data varies considerably, and it is clear that there are many outliers in the data. Especially in the Down direction, the PARS struggles to give accurate measurements. Furthermore, when flying east or west, i.e. travelling away from or toward the PARS station, it struggles more than flying perpendicular to the PARS in a North or south direction. The dataset is plotted using the ground antenna location as the origo for the NED frame.

5.2.1 Dataset processing

In order to visualise the data and use the measurements in a loosely coupled system, they are converted into position vectors in the $\mathcal{F}_n$. The PARS measurements arrive relative to the radio frame depicted in Figure 2.5. $\boldsymbol{p}^r_{rb}$, (2.77) , is computed to obtain the PARS position vector. The position vector, $\boldsymbol{p}^r_{rb}$, is converted from the radio frame to the NED frame with

$$\boldsymbol{p}^n_{nb} = \boldsymbol{R}_{nr}\boldsymbol{p}^r_{rb} \tag{5.1}$$

The rotation matrix $\boldsymbol{R}_{nr}$ is the orientation of the radio relative to the NED frame. This orientation is found through calibration algorithms completed by Okuhara et al. [59], and the resulting rotations matrix for flight 1 is given in (B.2) in Appendix B. The PARS covariance matrix, $\boldsymbol{R}_s$, is transformed to Cartesian coordinates with (2.80) and converted relative to the NED frame through a similarity

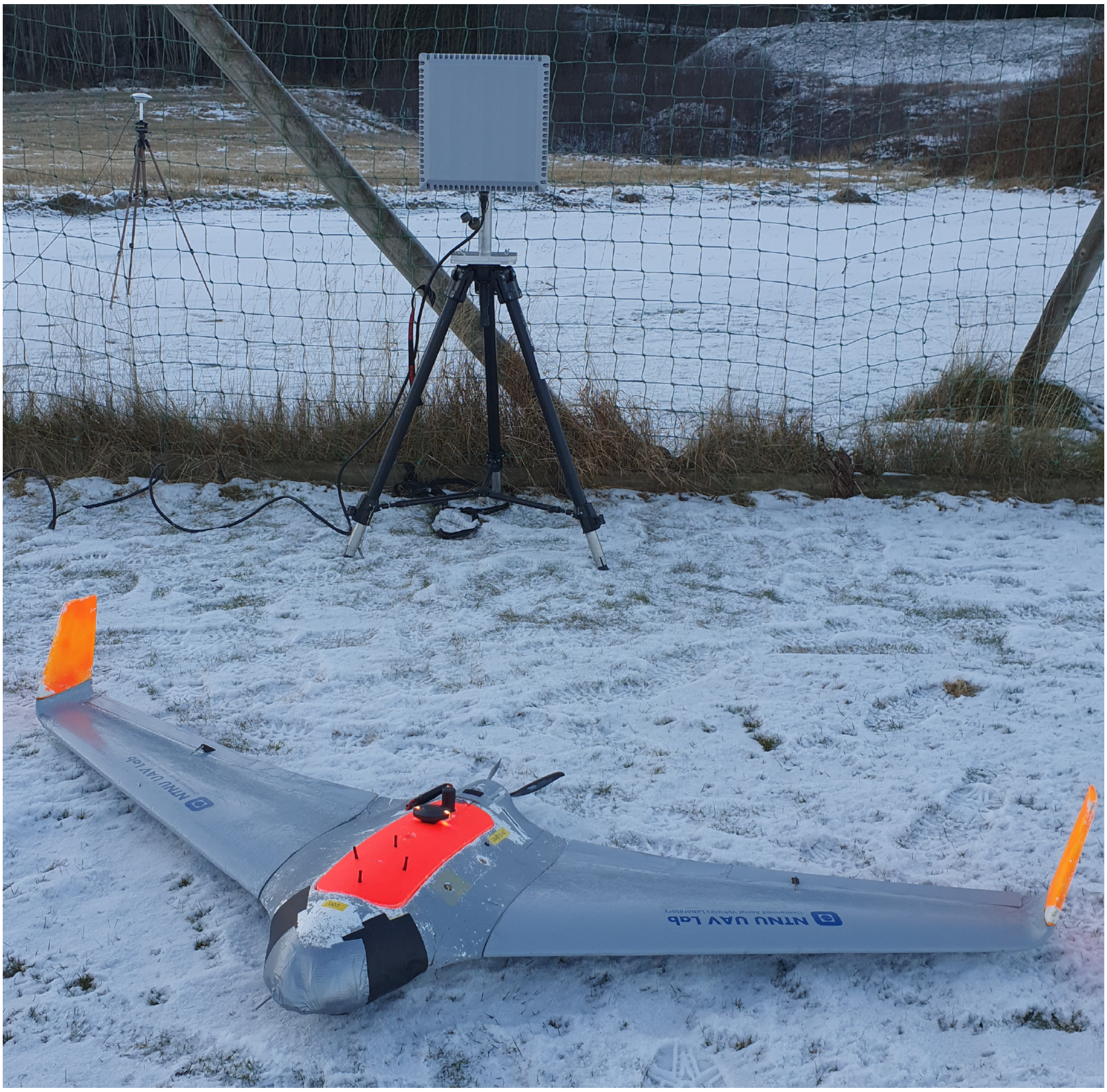

Figure 5.2: Image of the Skywalker X8 UAV and a CRE189 ground unit. Taken from [22].

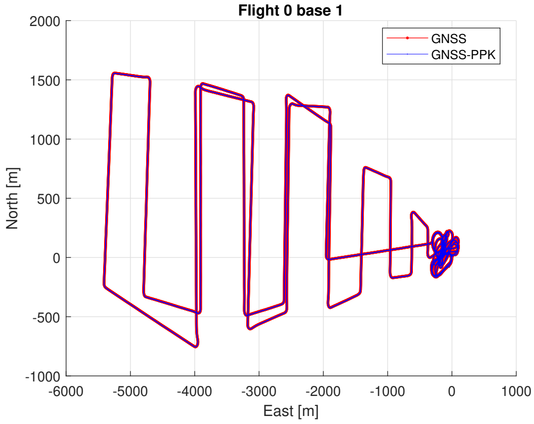

Figure 5.3: GNSS measurements versus GNSS-PPK measurements in 2D for flight 1.

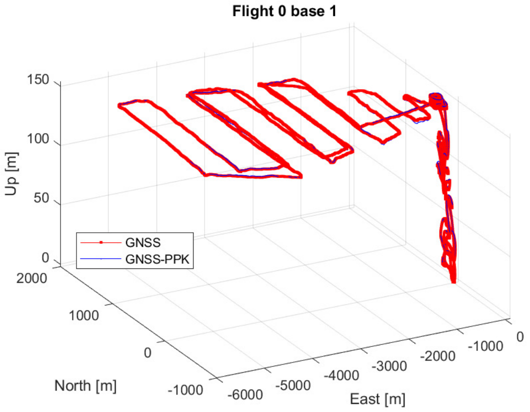

Figure 5.4: GNSS measurements versus GNSS-PPK measurements in 3D for flight 1.

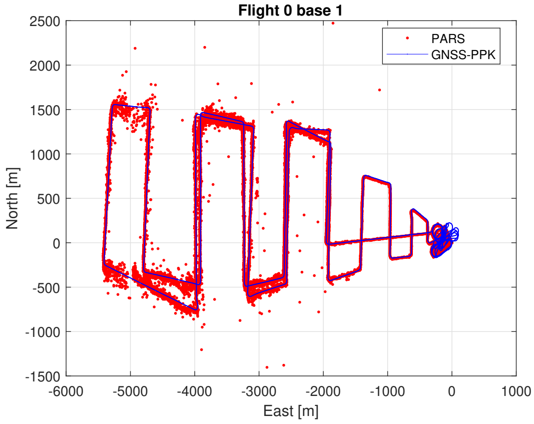

Figure 5.5: PARS measurements versus GNSS-PPK measurements in 2D for flight 1.

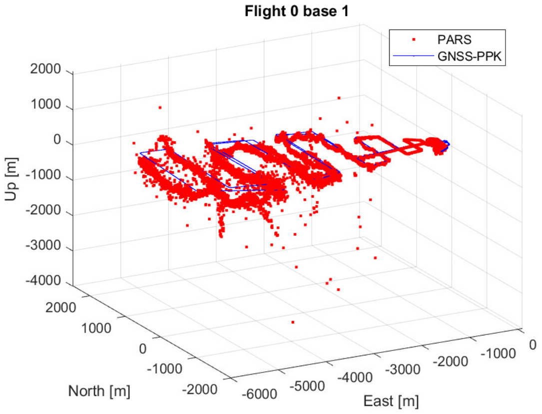

Figure 5.6: PARS measurements versus GNSS-PPK measurements in 3D for flight 1.

transform

$$\boldsymbol{R}_{pars} = \boldsymbol{R}_{nr}\boldsymbol{R}^{r}_{pars}\boldsymbol{R}_{rn} \tag{5.2}$$

The GNSS measurements provide position vectors relative to the ECEF frame, $\boldsymbol{p}^{e}_{eb}$. These are converted into the NED frame according to

$$\boldsymbol{p}^{n}_{nb} = \boldsymbol{R}^{T}_{en}(\boldsymbol{p}^{e}_{eb} - \boldsymbol{p}^{e}_{er}) \tag{5.3}$$

The rotation matrix $\boldsymbol{R}_{en}$ is the orientation of the NED frame relative to the ECEF frame. This is calculated with a function from the navigation toolbox using the location of the PARS. The resulting rotation matrix for flight 1 is stated in (B.3) in Appendix B.

Lastly, the IMU measurements are given as specific force, $\boldsymbol{f}^{m}_{nb}$, and angular velocity, $\boldsymbol{\omega}^{m}_{nb}$, relative to the measurement frame. These must be transformed into the body frame to propagate the position. This is accomplished with

$$\begin{aligned} \boldsymbol{a}^{b}_{nb} &= \boldsymbol{R}^{T}_{mb}\boldsymbol{a}^{m}_{nb} \\ \boldsymbol{\omega}^{b}_{nb} &= \boldsymbol{R}^{T}_{mb}\boldsymbol{\omega}^{m}_{nb} \end{aligned} \tag{5.4}$$

The rotation matrix $\boldsymbol{R}_{mb}$ is the installed orientation of the IMU given in (B.1) in Appendix B. Specific force, and angular velocity are converted into velocity and orientation by multiplying with the timestep assuming zero-order hold according to (2.70).

5.3 MEKF Implementation

The MEKF is implemented using the MATLAB coding program. The implementation in this thesis is inspired by the implementations in the navigation toolbox and based on the algorithm stated in Section 3.1.3. In the following implementation, the PARS and GNSS are converted to the NED frame, and IMU is converted to the body frame according to (5.1)-(5.4), respectively.

5.3.1 Prediction step

The prediction step involves Euler integration of position and velocity in (3.12), an attitude update step presented by Groves and computation of (3.13). This implies that the attitude propagation is not computed on the manifold. The predictions are computed from time $t = i$ and the timestep $t = i + \Delta t$.

Propagate attitude

The attitude is updated based on Groves [9, Appendix E, p. E-13]. Groves present a formulation using an attitude update quaternion, $\boldsymbol{\Delta q}$, defined as

$$\boldsymbol{\Delta q}_{nb} = \begin{bmatrix} \cos\left(\frac{||\Delta\boldsymbol{\Theta}_{nb}||_2}{2}\right) \\ \sin\left(\frac{||\Delta\boldsymbol{\Theta}_{nb}||_2}{2}\right)\frac{\boldsymbol{\omega}^{m}_{nb}}{||\boldsymbol{\omega}^{m}_{nb}||} \end{bmatrix} \tag{5.5}$$

and is computed using the measured angular velocity. The orientation is then updated according to

$$\boldsymbol{q}_{nb,i+\Delta t} = \boldsymbol{q}_{nb,i} \otimes \boldsymbol{\Delta q}_{nb} \tag{5.6}$$

For every prediction the quaternion is normalised according to

$$\boldsymbol{q}_{nb,i+\Delta t} = \frac{\boldsymbol{q}_{nb,i+\Delta t}}{\sqrt{\eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2}} \tag{5.7}$$

to ensure that the orientation belong to $SO(3)$.

Propagate Position

The position and velocity is updated using a first-order approximate Euler integration of (3.12). This yields

$$\begin{aligned} \boldsymbol{v}^n_{nb,i+\Delta t} &= \boldsymbol{v}^n_{nb,i} + \boldsymbol{a}^n_{nb,i}\Delta t && \text{(5.8a)} \\ \boldsymbol{p}^n_{nb,i+\Delta t} &= \boldsymbol{p}^n_{nb,i} + \frac{(\boldsymbol{v}^n_{nb,i} + \boldsymbol{v}^n_{nb,i+\Delta t})}{2}\Delta t && \text{(5.8b)} \end{aligned}$$

between timestep $t = i$ and $t = i + \Delta t$. In (5.8b), the propagated velocity is taken as an average over a timestep. Based on the predicted attitude in (5.6), (5.8a) expand to

$$\begin{aligned} \boldsymbol{v}^n_{nb,i+\Delta t} &= \boldsymbol{v}^n_{nb,i} + \boldsymbol{a}^n_{nb}\Delta t \\ &= \boldsymbol{v}^n_{nb,i} + (\bar{\boldsymbol{R}}_{nb}\boldsymbol{f}^b_{imu} + \boldsymbol{g}^n(\boldsymbol{p}^n_{nb}) - \boldsymbol{b}^b_{acc} - \boldsymbol{w}^b_{acc})\Delta t \\ &= \boldsymbol{v}^n_{nb,i} + \bar{\boldsymbol{R}}_{nb}\boldsymbol{f}^b_{imu}\delta t + \boldsymbol{g}^n(\boldsymbol{p}^n_{nb})\delta t - \boldsymbol{b}^b_{acc}\Delta t - \boldsymbol{w}^b_{acc}\Delta t \\ &= \boldsymbol{v}^n_{nb,i} + \bar{\boldsymbol{R}}_{nb}\Delta\boldsymbol{v}^b_{nb} + \boldsymbol{g}^n(\boldsymbol{p}^n_{nb})\Delta t - \boldsymbol{b}^b_{acc}\Delta t - \boldsymbol{w}^b_{acc}\Delta t \end{aligned} \tag{5.9}$$

The rotation matrix $\bar{\boldsymbol{R}}_{nb}$ is an average of two timesteps similar to velocity in (5.8b) and is approximated as

$$\bar{\boldsymbol{R}}_{nb} \approx \frac{\boldsymbol{R}_{nb}(\boldsymbol{q}_{nb,i}) + \boldsymbol{R}_{nb}(\boldsymbol{q}_{nb,i+\Delta t})}{2} \tag{5.10}$$

See, [9, Ch. 5] for details.

Bias propagation

The biases are propagated with

$$\begin{aligned} \boldsymbol{b}^b_{acc,i+\Delta t} &= e^{-\frac{\Delta t}{T_{acc}}}\boldsymbol{b}^b_{acc,i} && \text{(5.11a)} \\ \boldsymbol{b}^b_{ars,i+\Delta t} &= e^{-\frac{\Delta t}{T_{ars}}}\boldsymbol{b}^b_{ars,i} && \text{(5.11b)} \end{aligned}$$

Covariance matrix propagation

The covariance matrix $\boldsymbol{P}$ is predicted according to

$$\boldsymbol{P}_i = \boldsymbol{A}_{d,i}\boldsymbol{P}_{i-1}\boldsymbol{A}_{d,i}^T + \boldsymbol{Q}_{d,i}. \tag{5.12}$$

The discrete-time transition matrix, $\boldsymbol{A}_d$, and process noise covariance, $\boldsymbol{Q}_d$ is calculated using Van Loan algorithm, similar to [60, p. 126]. To ensure a symmetric covariance matrix it is normalised using

$$\boldsymbol{P}_i = \frac{\boldsymbol{P}_i + \boldsymbol{P}_i^T}{2} \tag{5.13}$$

5.3.2 Update step

The Update step is dependent on the type of measurement but is straightforward after converting to NED. The measurements are implemented as a loosely coupled system and used directly as 3D position measurement, $\boldsymbol{p}_{nb}^n$ in (3.14).

5.4 FGO Implementation

The FGO is implemented using C++ and the GTSAM framework. Additionally, since the dataset is given as a MATLAB file format, ROS and MATLAB are utilised to send the data to the FGO system. Using the GTSAM framework simplified the implementation process considerably since FG, optimisation and lie theory are mathematically complicated and time-consuming if implemented from scratch.

A preintegration object and an IMU factor object handle the preintegration and IMU accumulation in Section 3.2.1. An in-between factor object simulates the bias factor in Section 3.2.2. Lastly, a GNSS factor object handles the GNSS factor stated in Section 3.2.3. PARS position measurements are included through the GNSS factor. Including PARS measurements directly requires a PARS factor to become a tightly coupled system. Some initial computations were produced for a PARS factor; however, it was not implemented because of time constraints.

The factors are added into a nonlinear FG using an add function in an FG object. Lastly, an optimiser object noted a BatchFixedLagSmoother solve the optimisation problem using the Levenberg Marquardt algorithm over a specific batch defined by the lag size. The lag determines how many seconds of history is included in the optimisation problem.

GTSAMs standard configuration is preintegration on the tangent plane; therefore, to integrate on the manifold, it is necessary to set this specifically.

5.5 Tuning

Tuning of the priors and correction measurements is straightforward. The priors are tuned based on the accuracy of the initial guess, while the correction is tuned based on the accuracy of the measurements. Tuning the predictions are more complicated because of unit conversions. The methods

Table 5.1: Prior tuning of the MEKF and FGO algorithms.

Variable	Tuning	Unit
$\boldsymbol{\sigma}_{p_0}$	$10 \cdot \boldsymbol{I}_3$	$[\mathrm{m}]$
$\boldsymbol{\sigma}_{v_0}$	$2 \cdot \boldsymbol{I}_3$	$\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$
$\boldsymbol{\sigma}_{\Theta_0}$	$\mathrm{diag}(15, 15, 50) \cdot \frac{\pi}{180}$	$[\mathrm{rad}]$
$\boldsymbol{\sigma}_{b_0^{acc}}$	$7 \cdot g_0 \cdot 10^{-3} \boldsymbol{I}_3$	$\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$
$\boldsymbol{\sigma}_{b_0^{ars}}$	$\frac{360}{3600} \frac{\pi}{180} \boldsymbol{I}_3$	$\left[\frac{\mathrm{rad}}{\mathrm{s}}\right]$

Table 5.2: Prediction tuning of the MEKF and FGO algorithms.

Variable	Tuning	Unit
$\boldsymbol{\sigma}_{q^{ars}}$	$9.5993 \cdot 10^{-4}$	$\left[\frac{\mathrm{rad}}{\mathrm{s}} \frac{1}{\mathrm{Hz}}\right]$
$\boldsymbol{\sigma}_{q_{bias}^{ars}}$	$5.7136 \cdot 1{-}8$	$\left[\frac{\mathrm{rad}}{\mathrm{s}^{1.5}}\right]$
$\boldsymbol{\sigma}_{q^{acc}}$	0.0257	$\left[\frac{\mathrm{m}}{\mathrm{s}^2} \frac{1}{\mathrm{Hz}^{0.5}}\right]$
$\boldsymbol{\sigma}_{q_{bias}^{acc}}$	$2.3179 \cdot 10^{-4}$	$\left[\frac{\mathrm{m}}{\mathrm{s}^{2.5}}\right]$
$\boldsymbol{\sigma}_{int}^{*}$	10^{-8}	$\left[\frac{\mathrm{m}}{\mathrm{s}\,\mathrm{s}^{0.5}}\right]$

* Only for FG.

used here are found in the work by Farrel [61] and Farrell et al. [62]. Relevant information for the STIM300 IMU, Ublox F9P-ZED GNSS and CRE189 ground antenna is found in their respective datasheets, [56–58]. All variables are tuned in continuous time.

Prior

The initial guess is often initialised using GNSS-PPK for position and AHRS for attitude for an accurate guess, which should yield a small initial variance. For this project, the values stated in Table 5.1 are used. The Euler angle standard deviation for attitude is converted to the MRP representation for MEKF. The accelerometer bias is computed by extracting the one-year maximum bias from the datasheet. It lies between 1.5 and 7.5 mg, and 7 mg is used for the prior. The gyroscope bias is found by extracting the bias range. The datasheet state +/- 250 $\frac{\circ}{h}$ but it is often larger, so $360\frac{\circ}{h}$ is used for the prior.

Prediction

The tuned standard deviations for prediction are stated in Table 5.2. The gyroscope random walk is computed according to

$$\boldsymbol{\sigma}_{q^{ars}} = 22 \cdot \frac{0.15\pi}{60 \cdot 180} \tag{5.14}$$

where the angular random walk 0.15 $\frac{\circ}{\sqrt{\mathrm{h}}}$ is extracted from the STIM 300 datasheet and then converted to the appropriate unit. It is adjusted by the UAV lab with a tuning factor of 22. The gyroscope bias is computed according to

$$\boldsymbol{\sigma}_{q^{ars}_{bias}} = \sqrt{\frac{2}{3600}\left(\frac{0.5\pi}{180 \cdot 3600}\right)^2} \tag{5.15}$$

where the bias instability, 0.5 $\frac{\circ}{\sqrt{\mathrm{h}}}$, is extracted from the datasheet and converted to the appropriate unit. The accelerometer random walk is computed as

$$\boldsymbol{\sigma}_{q^{acc}} = 22 \cdot \frac{0.07}{60} \tag{5.16}$$

where the velocity random walk 0.07 $\frac{\mathrm{m}}{\mathrm{s}}\frac{1}{\sqrt{\mathrm{h}}}$ is extracted from the datasheet and converted to the appropriate unit. Also here an adjustable tuning factor of 22 is added as a result of tuning by the UAV lab. The accelerometer bias is computed by extracting the bias instability 0.05 mg from the datasheet and converted to correct units

$$\boldsymbol{\sigma}_{q^{acc}_{bias}} = \sqrt{\frac{2}{3600}\left(\frac{0.05g_0}{1000}\right)^2} \tag{5.17}$$

The integration uncertainty in the FGO is set to 10^{-8}. This was not adjusted and tested much, and the value was extracted from an example by GTSAM.

Correction

The tuned GNSS and PARS standard deviations are stated in Table 5.3.

Table 5.3: Correction measurment tuning of the MEKF and FGO algorithms.

Variable	Tuning	Unit
$\sigma_{p_x^{gnss}}$	0.2	[m]
$\sigma_{p_y^{gnss}}$	0.2	[m]
$\sigma_{p_z^{gnss}}$	0.4	[m]
$\sigma_{\rho^{pars}}$	15	[m]
$\sigma_{\xi^{pars}}$	2	[deg]
$\sigma_{\alpha^{pars}}$	2	[deg]

Chapter 6

Results and Discussion

This chapter investigates the implemented MEKF and FGO algorithms. Estimations of both position and attitude with GNSS- and PARS-aided INS are analysed. Firstly, the MEKF and FGO are compared using GNSS measurements. Then MEKF and FGO are inspected using PARS-aided INS, followed by an examination of various fixed lags for the FGO with PARS-aided INS. The estimation schemes are assessed using ME (2.54), MAE (2.55), STD (2.56) and RMSE (2.57) as indexes to determine their accuracy. Each section includes an evaluation of the results, incorporating relevant observations before a final discussion concludes the chapter. A complete set of plots are provided in Appendix D for each section.

6.1 Simulation scenario

The estimation schemes are tested with the dataset reported in Section 5.2, with the flight visualised in Figure 5.1. GNSS is a primary navigation tool and is used throughout the entire flight. On the other hand, PARS is a backup sensor, and to simulate a realistic scenario, the GNSS is assumed to lose signal, and after that, PARS switches on during the flight. This also ensures reliable initial information on the pose, which is often necessary for these schemes.

Furthermore, some choices are made that affect the results. Because of complications with GTSAM, it is yielded necessary to use every other GNSS and PARS measurement. This is a known problem; further reading is found in the GTSAM user forum [63]. Further, the validated PARS data through outlier rejection with the MEKF is also applied for the FGO for comparable data. Lastly, it is essential to note that the implemented systems are loosely coupled.

6.2 GNSS-aided INS filtering

Both the MEKF and FGO successfully estimate the position and attitude of the UAV with GNSS-aided INS. Table 6.1 state the performance of both estimation schemes showing exceedingly similar and high accuracy. The FGO has a slightly higher precision in the Down and Roll, while MEKF performs marginally better in estimating Pitch and Yaw.

Figure 6.1 visualise the flight in 3D with MEKF and FGO estimations. Both schemes follow the trajectory closely, with no considerable deviations. The position error is plotted in Figure 6.2, which is of small scale, at the largest up to 2 m in the North and Down direction. They produce equivalent errors, which implies that both compute the optimal position estimate given the noisy measurements. It also seems to be a positive ME present, and removing it results in a low STD, according to Table 6.1. It is expected that this constant ME is present in the GNSS measurements and is corrected for when computing the GNSS-PPK data.

Furthermore, the schemes have more or less equivalent attitude estimation, with the attitude plotted in Figure 6.4. Both algorithms struggle the most with estimating yaw, as seen in Figure 6.3. Their estimates are similar but constantly slightly off the AHRS ground truth. An inaccurate autopilot could explain this since it is more realistic for the heading to be zero than a few degrees above, which is the case for multiple sequences depicted in Figure 6.4. Also, the autopilot takes longer to get a steady heading after a change in direction. Additionally, both schemes struggle with estimating pitch which could be expected as the pitch is rarely exited during flight. Since it is rarely exited, the errors are minor because the pitch angle is generally of small scale.

The accelerometer and gyroscope bias estimations are depicted in Figure 6.5. The estimated bias is of a small proportion where the FGO manages to converge faster than the MEKF. They mainly estimate a similar bias, which can originate from various offsets, such as the IMU installation. Their is no fluctuation in the estimates indicating no complications in estimation of attitude and position.

Table 6.1: Statistics of MEKF and FGO position and attitude estimates, with GNSS aided INS, relative to GNSS-PPK.

Position	**North [m]**		**East [m]**		**Down [m]**	
Method	MEKF	FGO	MEKF	FGO	MEKF	FGO
ME	1.37	1.37	0.424	0.425	0.0856	0.0713
MAE	1.37	1.37	0.427	0.427	0.693	0.688
STD	0.419	0.419	0.232	0.230	0.905	0.904
RMSE	1.43	1.43	0.483	0.483	0.909	0.906
Attitude	**Roll [°]**		**Pitch [°]**		**Yaw [°]**	
Method	MEKF	FGO	MEKF	FGO	MEKF	FGO
ME	0.875	0.866	0.701	0.751	2.295	2.37
MAE	1.66	1.59	1.82	1.83	19.94	19.96
STD	2.18	2.05	2.19	2.16	23.23	23.27
RMSE	2.35	2.22	2.30	2.29	23.34	23.39

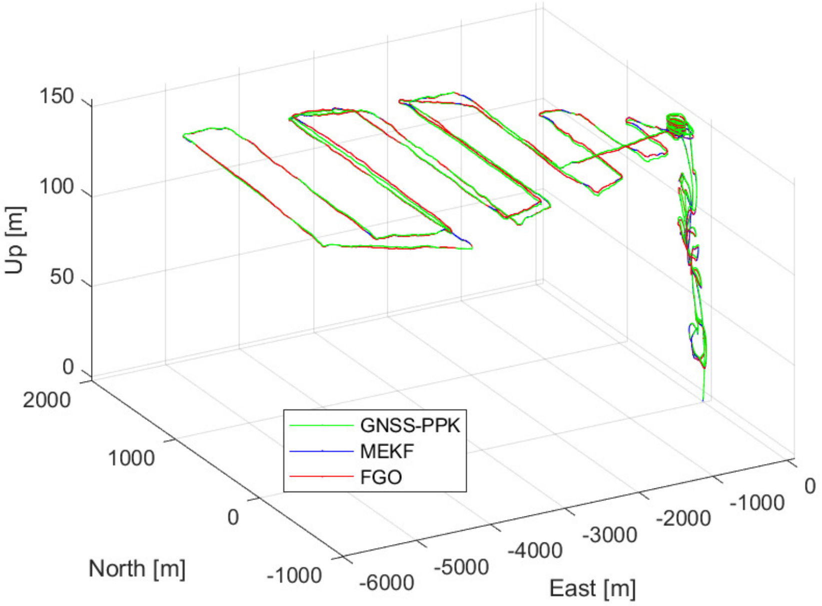

Figure 6.1: Position estimates by the MEKF and FGO algorithms with GNSS as input, plotted with the GNSS-PPK in 3D.

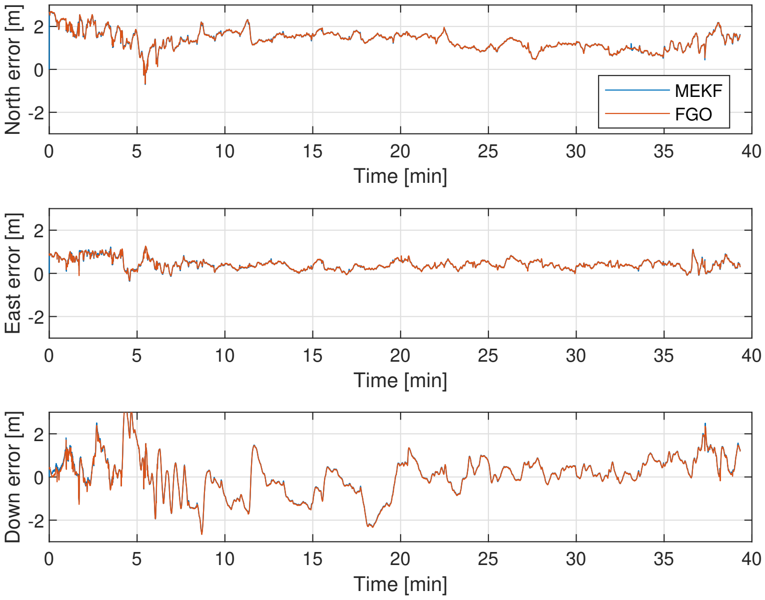

Figure 6.2: Position error by MEKF and FGO estimates relative to GNSS-PPK in North, East and Down direction.

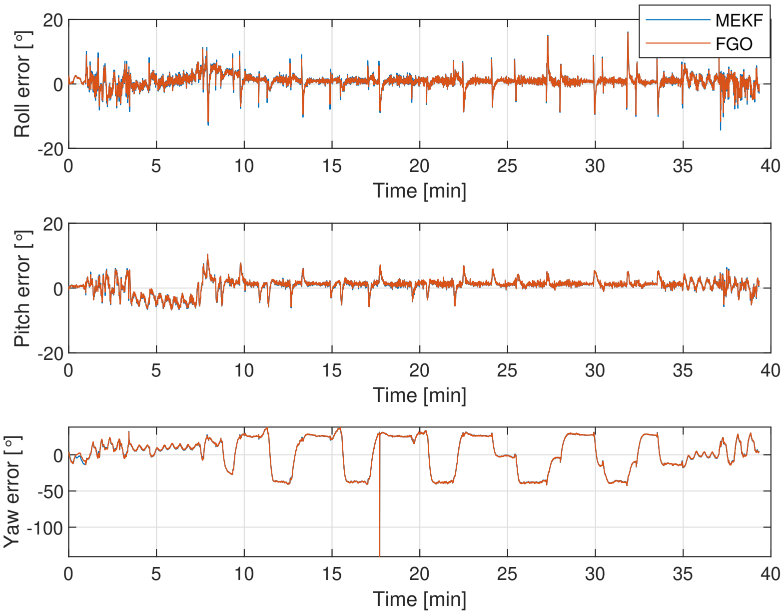

Figure 6.3: Attitude error by MEKF and FGO relative the AHRS with GNSS measurements as input.

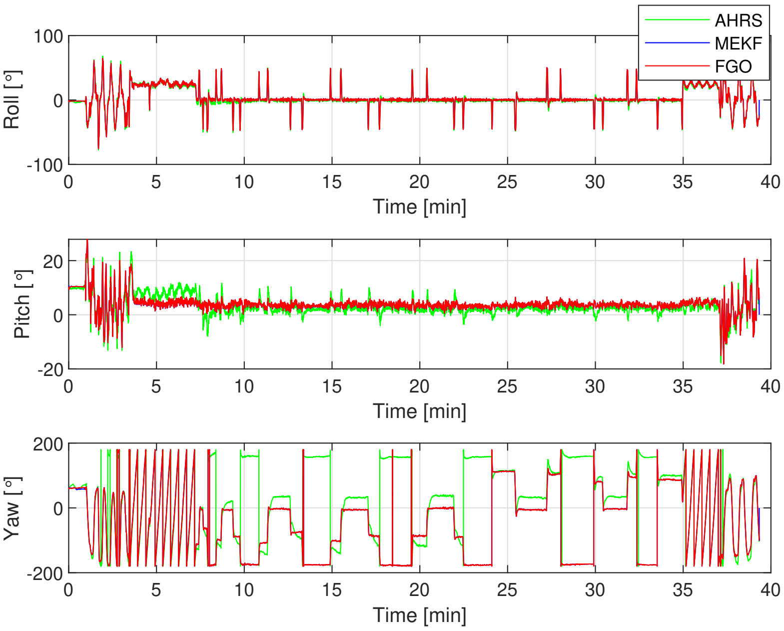

Figure 6.4: Attitude estimates by MEKF and FGO, plotted with the AHRS using GNSS as input.

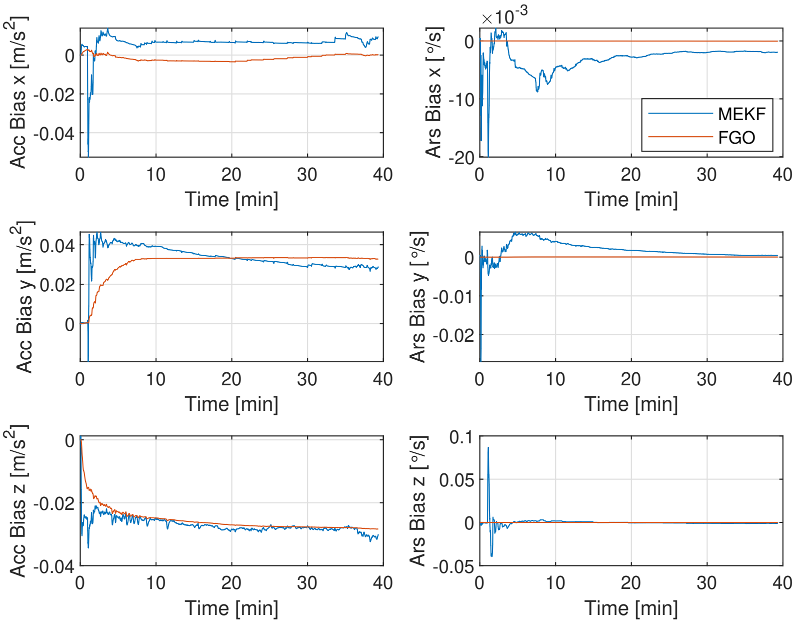

Figure 6.5: Bias estimates by MEKF and FGO algorithms with GNSS as measurment input.

6.3 PARS-aided INS filtering

For PARS-aided INS estimation, the measurements in Figure 5.5 and Figure 5.6 are used. GNSS is utilised for the first part of the flight before the PARS takes over, defined by the Switch in the various plots. There is a considerable amount of outliers in the data, especially in the Down direction, which will challenge the estimation schemes. The measurement error between the PARS and GNSS-PPK measurements is depicted in Figure 6.6. Some of these are large; therefore, a validation gate is applied to filter out the worst outliers, depicted in Figure 6.7. In this section, the focus is on how each scheme performs in the presence of outliers.

6.3.1 Without validation gate

Both schemes struggle heavily to estimate the position accurately, with an apparent offset in the estimations and a significant variance stated in Table 6.2. This is also depicted in Figures 6.8 and 6.9 in 3D and 2D, respectively. Nevertheless, they provide similar estimates, implying they find the best fit according to the given data. FGO yield slightly better performance in the Down direction, while the MEKF is more accurate in the North and East direction. On the other hand, they provide precise attitude estimations visualised in Figure 6.11, challenging the accuracy of GNSS-aided PARS.

Both schemes are expected to produce better estimations in the North and East directions than the Down direction because of the large number of multipath measurements. This is also the case, as stated in Table 6.2 and visualised in Figure 6.8. Also, measurements further away from the radio have a more significant error since the errors propagate with the range. Therefore, the error is smaller at

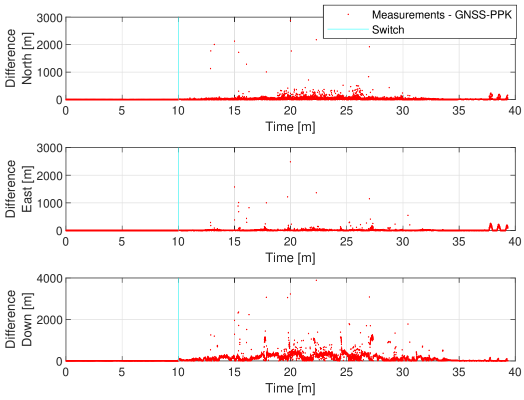

Figure 6.6: Difference between correction measurements, PARS and GNSS, and the GNSS-PPK measurements with no validation gate.

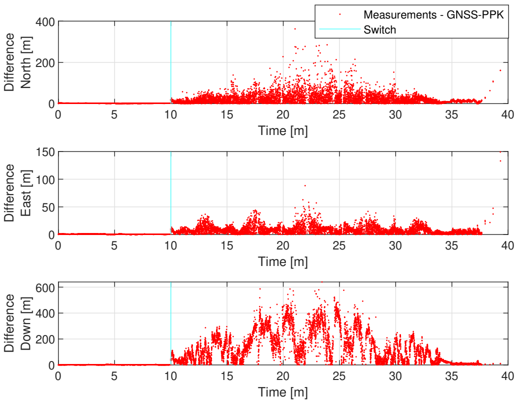

Figure 6.7: Difference between correction measurements, PARS and GNSS, and the GNSS-PPK measurements with validation gate using the chi squared distribution test with $\chi^2_{\alpha=0.1,\eta_y=3} = 6.251$.

the switch's start and the dataset's end. This is clearly visualised in the North and Down direction in Figure 6.10. Additionally, the error is more extensive in the North direction compared to the East direction, which is supported by Table 6.2 and Figure 6.10. Therefore, some limitations with the PARS are present when the UAV travels away or towards the radio versus flying perpendicular to it.

These results motivate the usage of outlier rejection to reduce multipath propagation and the need for alternative sensors to increase accuracy, such as a barometer and a magnetometer, as stated by Albrektsen et al. and Gryte et al. [17, 45].

Table 6.2: Statistics of MEKF and FGO position and attitude estimates, with PARS aided INS without validation gate, relative to GNSS-PPK.

Position	**North [m]**		**East [m]**		**Down [m]**	
Method	MEKF	FGO	MEKF	FGO	MEKF	FGO
ME	-9.26	-9.89	-14.96	-17.09	169.96	154.08
MAE	70.33	79.10	22.33	24.42	174.18	157.97
STD	193.42	222.68	48.54	53.36	223.80	203.09
RMSE	193.63	222.88	50.79	56.02	281.01	254.92

Attitude	**Roll [°]**		**Pitch [°]**		**Yaw [°]**	
Method	MEKF	FGO	MEKF	FGO	MEKF	FGO
ME	0.853	0.784	1.14	1.25	-1.49	-1.79
MAE	1.83	1.78	1.95	2.04	21.37	21.66
STD	2.45	2.43	2.43	2.49	24.85	25.21
RMSE	2.60	2.55	2.68	2.78	24.90	25.27

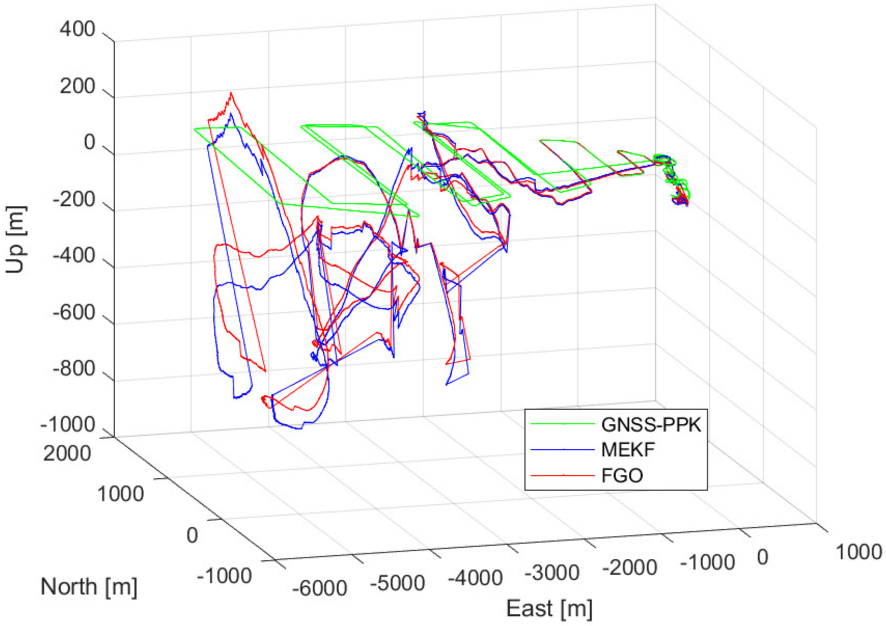

Figure 6.8: Position estimates by the MEKF and FGO algorithms with PARS, without validation gate, as input, plotted with the GNSS-PPK in 3D.

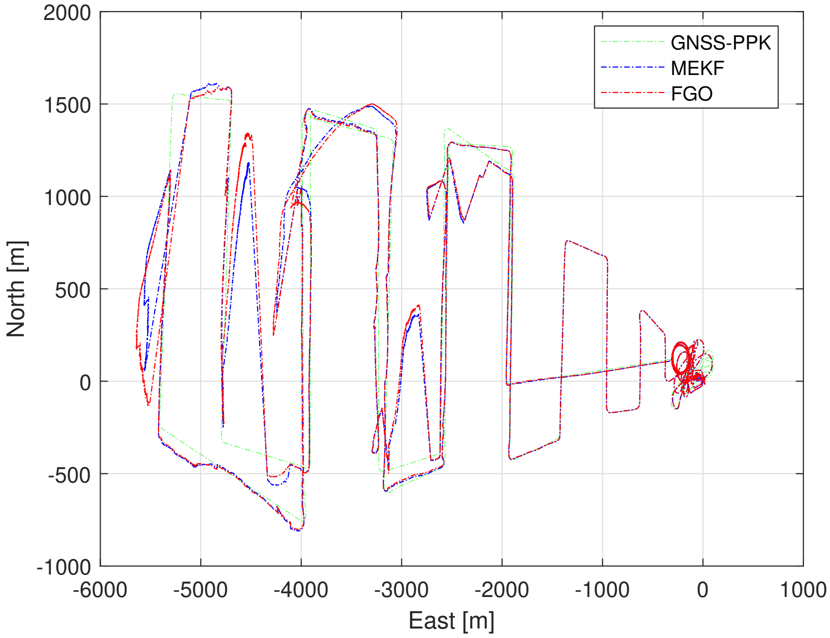

Figure 6.9: Position estimates by the MEKF and FGO algorithms with PARS, without validation gate, as input, plotted with the GNSS-PPK in 2D.

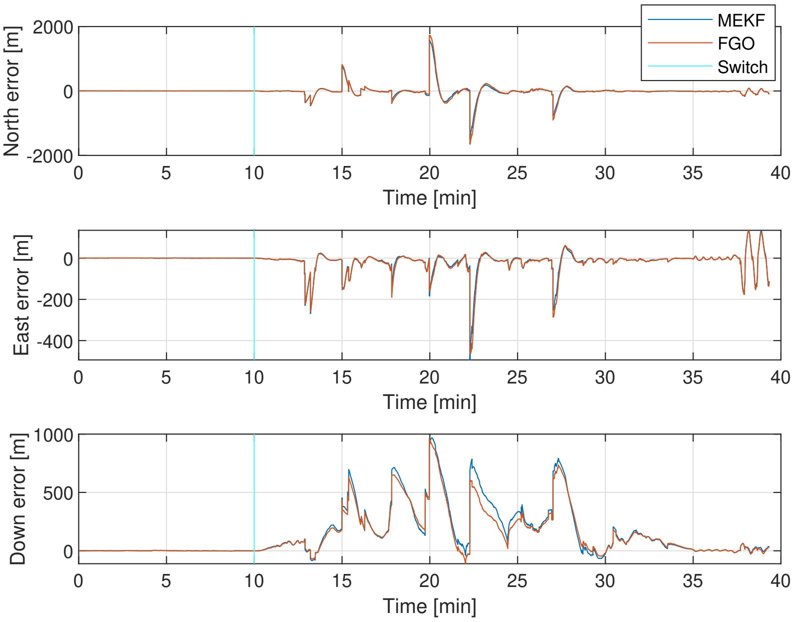

Figure 6.10: Position error by MEKF and FGO estimates relative to GNSS-PPK in North, East and Down direction, with PARS measurments without validation gate.

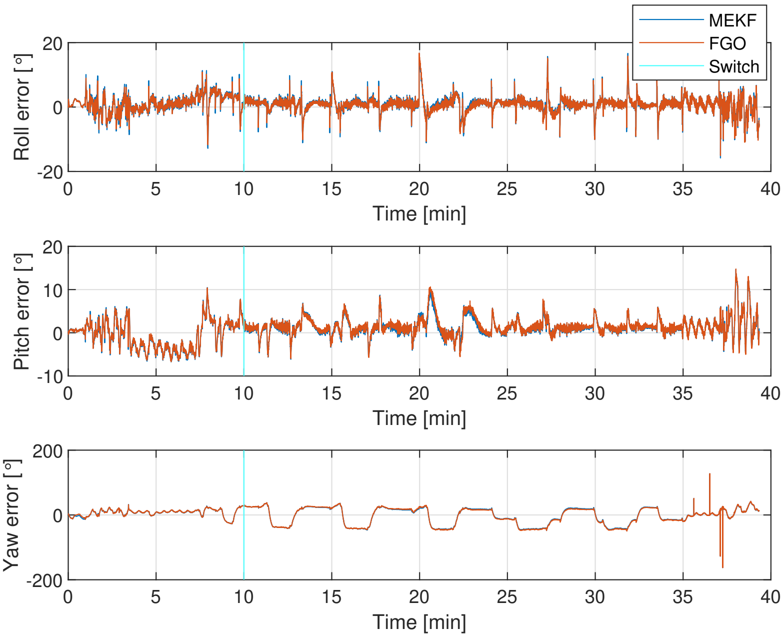

Figure 6.11: Attitude error by MEKF and FGO relative the AHRS with PARS measurements, without validation gate, as input.

6.3.2 With MEKF validation gate

Introducing the validation gate, with $\chi^2_{\alpha=0.1,\eta_y=3} = 6.251$, a total of 1794 measurements are rejected out of 15308 PARS measurements. This results in a considerable reduction in the difference between PARS and GNSS-PPK measurements, as visualised in Figure 6.7, compared to Figure 6.6.

The performance has increased considerably compared to those without a validation gate, according to Table 6.3. Still, the precision falls short to the level achieved by GNSS. Therefore, even with outlier rejection, the schemes still struggle to estimate the position accurately, especially the Down direction visualised in Figure 6.13. Nevertheless, they reasonably estimate the long stretches in the North and South direction, where the PARS data is more precise, depicted in Figure 6.12. Considering these findings, the PARS is inadequate in the Down direction, and it is necessary to utilise other measurements, such as a barometer, for real-time navigation with PARS.

Both schemes yield similar results, with MEKF slightly better in the North and East direction and FGO better in the Down direction. At the same time, the attitude is more or less equivalent with high accuracy, visualised in Figure 6.15.

Table 6.3: Statistics of MEKF and FGO position and attitude estimates, with PARS aided INS with validation gate $\chi^2_{\alpha=0.1,\eta_y=3} = 6.251$, relative to GNSS-PPK.

Position	**North [m]**		**East [m]**		**Down [m]**	
Method	MEKF	FGO	MEKF	FGO	MEKF	FGO
ME	-7.67	-7.92	-6.22	-6.55	98.9	97.19
MAE	10.44	10.61	6.79	7.10	105.31	101.89
STD	11.55	11.61	6.16	6.07	111.04	105.65
RMSE	13.87	14.05	8.76	8.93	148.70	143.56

Attitude	**Roll [°]**		**Pitch [°]**		**Yaw [°]**	
Method	MEKF	FGO	MEKF	FGO	MEKF	FGO
ME	0.903	0.870	0.928	0.986	0.859	0.841
MAE	1.54	1.48	1.70	1.74	21.61	21.67
STD	1.98	1.90	1.91	1.89	24.77	24.83
RMSE	2.18	2.09	2.13	2.13	24.79	24.85

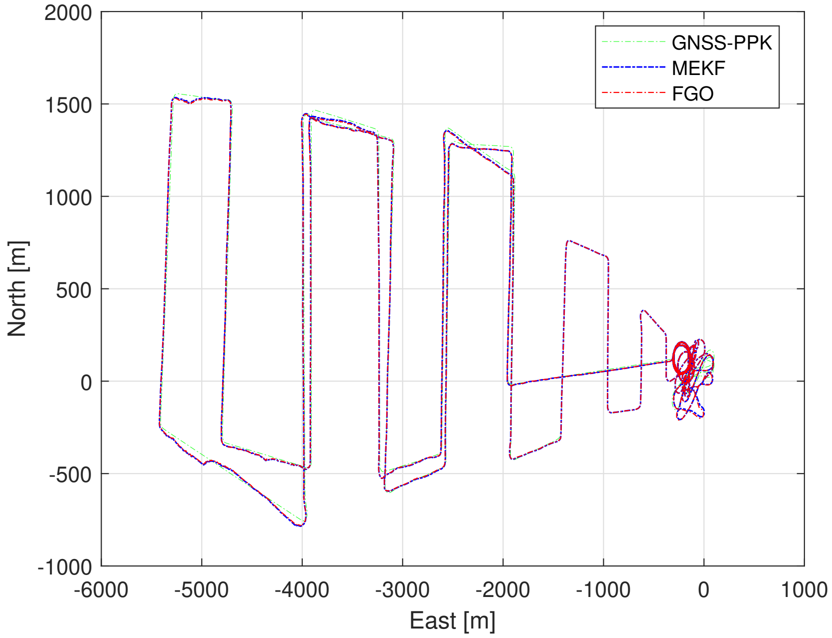

Figure 6.12: Position estimates by the MEKF and FGO algorithms with PARS, with validation gate, as input, plotted with the GNSS-PPK in 2D.

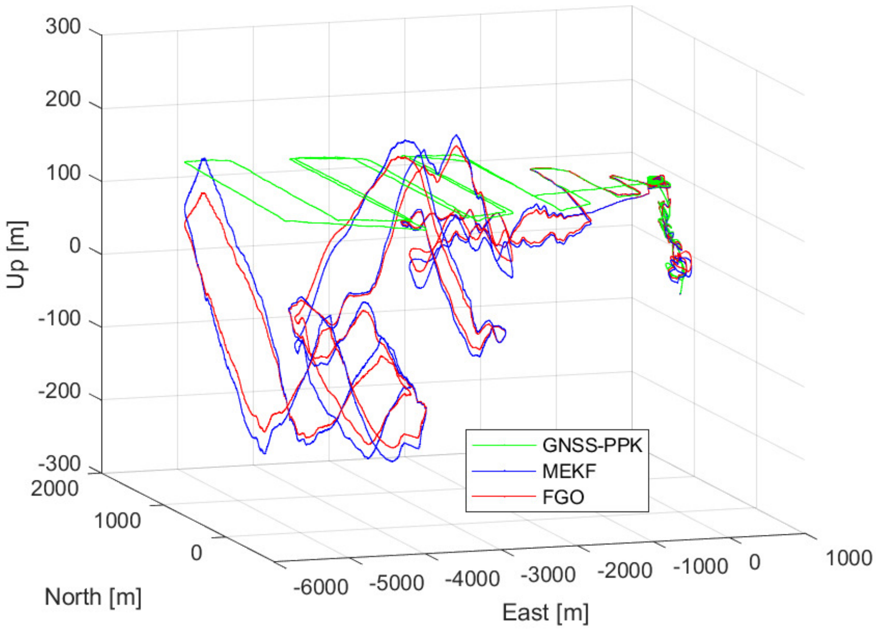

Figure 6.13: Position estimates by the MEKF and FGO algorithms with PARS measuremnt, with valdiation gate as input, plotted with the GNSS-PPK in 3D.

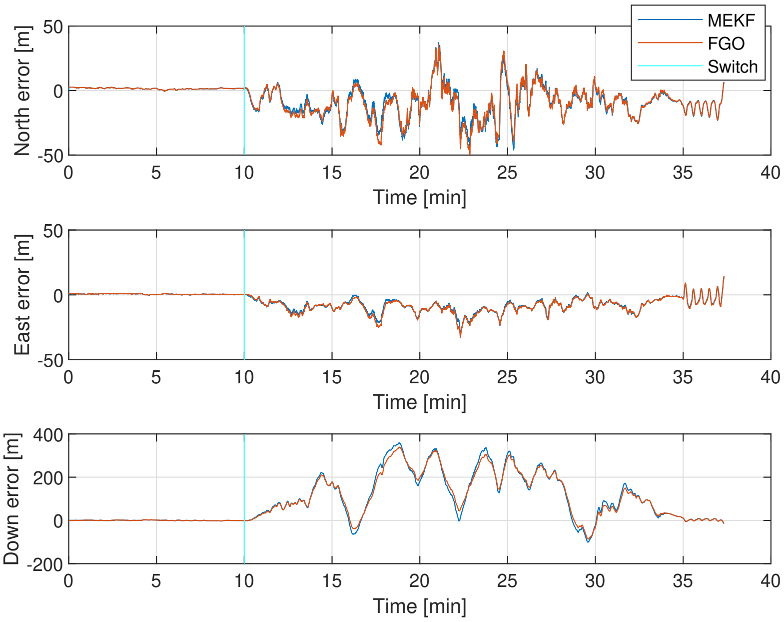

Figure 6.14: Position error by MEKF and FGO estimates relative to GNSS-PPK in North, East and Down direction using PARS with validation gate as input.

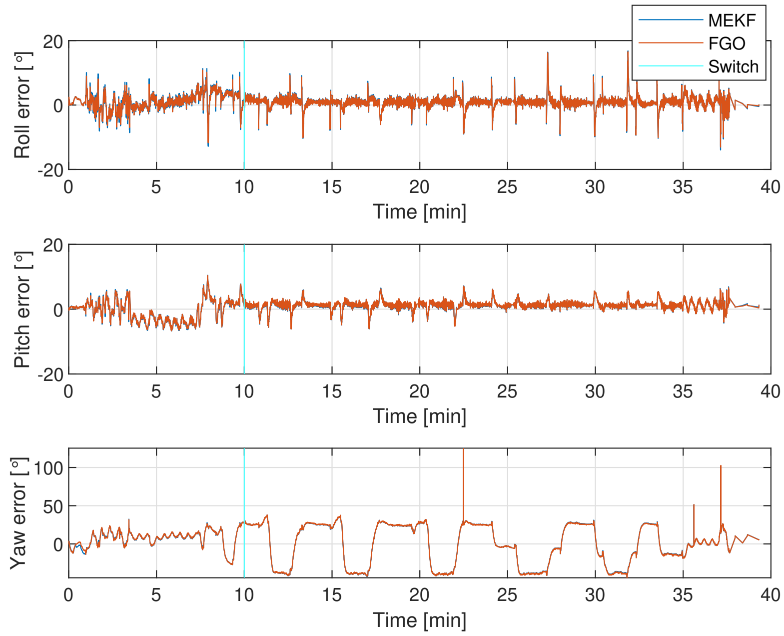

Figure 6.15: Attitude error by MEKF and FGO relative the AHRS with PARS measurements, with validation gate, as input.

6.4 Fixed-lag FGO filtering

As mentioned in Section 3.2 and Section 4.2, FGO can adjust the fixed lag for optimisation to define to what degree one desires filtering or smoothing. In an attempt to improve the filtering of the last node, the built-in functionality in GTSAM is used to include a set number of previous nodes when optimising to. Three cases are addressed with 0.1 s, 10 s and 100 s lag time. A lag of 0.1 s is traditional filtering, where only the last node is optimised, including the three variables; state $\boldsymbol{x}_i$, velocity $\boldsymbol{v}_i$ and bias $\boldsymbol{b}_i$. Also, for a 10 s lag all data in the last 10 seconds are included in the optimisation problem, and all data before that is marginalised out. Same for 100 s lag. Therefore, the number of optimised variables changes based on lag time according to Table 6.4, and the total optimisation time is affected accordingly. The total optimisation time defines the time it takes to solve the optimisation problem at every correction. The more variables to optimise, the longer each optimisation takes. Further, the total real-time computation includes data processing, preintegration of IMU measurements, creating factors, assembly of graph and optimisation. Also, the number of optimised nodes and variables are approximate because the PARS timestamps intervals vary.

Table 6.4: Number of optimised nodes, parameters and total optimisation time for each fixed-lag estimator.

Lag	0.1s	10s	100s
Num. opt. Nodes	1	~ 46	~ 466
Num. Opt. Params	3	~ 140	~ 1400
Tot. Opt. Time [s]	9.77	105.86	1038.19
Percentage Opt. Time of Real-time [%]	64.54	68.70	99.48

The resulting performance of the FGO algorithm is stated in Table 6.5, for each lag. They all yield similar results, so including past information when filtering does not improve performance. This contradicts the conclusion by Zhao et al. and Wen et al. [42, 44], whom both conclude that increasing the window size and including historical data produce a better estimation, especially when in the presence of outliers. The reason for the little impact is the choice of the loss function. The LS formulation optimises all nodes equally based on their variance; therefore, it is not considered a robust loss function. On the other hand, GTSAM provides a robust formulation which weights the nodes based on their difference on the last node. Also, other loss functions can be applied to get a more robust estimation in the presence of outliers.

Table 6.5: Statistics of FGO position and attitude estimates, with PARS aided INS with validation gate $\chi^2_{\alpha=0.1,\eta_y=3} = 6.251$, relative to GNSS-PPK for different fixed-lags. Position on the left column and attitude on the right column.

	North [m]			**Roll [°]**		
Lag	0.1s	10s	100s	0.1s	10s	100s
ME	-7.92	-7.95	-7.96	0.870	0.870	0.870
MAE	10.61	10.63	10.65	1.48	1.48	1.48
STD	11.61	11.82	11.84	1.90	1.90	1.90
RMSE	14.05	14.25	14.26	2.09	2.09	2.09

	East [m]			**Pitch [°]**		
Lag	0.1s	10s	100s	0.1s	10s	100s
ME	-6.55	-6.57	-6.58	0.986	0.986	0.986
MAE	7.10	7.13	7.13	1.74	1.74	1.74
STD	6.07	6.42	6.42	1.89	1.89	1.89
RMSE	8.93	9.18	9.19	2.13	2.13	2.13

	Down [m]			**Yaw [°]**		
Lag	0.1s	10s	100s	0.1s	10s	100s
ME	97.19	97.21	97.25	0.841	0.856	0.839
MAE	101.89	101.91	101.95	21.67	21.67	21.67
STD	105.65	105.69	105.73	24.83	24.83	24.83
RMSE	143.56	143.59	143.64	24.85	24.85	24.85

6.5 Discussion

This section takes on a general discussion of the results linked to the problem to be addressed. Firstly, limitations and assumptions affecting the results are considered, followed by an evaluation of the performance of the algorithms concerning their optimality and robustness. Finally, a discussion on the system's real-time applicability is undertaken.

6.5.1 Limitations and assumptions

The most central assumption in this thesis is that $\mathcal{F}_n$ is assumed inertial. The Earth's rotation should be considered, especially for this UAV flight scenario, covering several kilometres. This could be included by using the ECEF formulation in (2.73) or an equivalent non-inertial NED formulation. Adding this rotation is expected to yield a higher prediction accuracy and, consequently, better estimated trajectory. Additionally, a constant gravity is assumed, which is also incorrect. The gravity vector differs depending on the location of the UAV and should be computed at each position.

Another aspect that impacts the accuracy is the calibration of the radio orientation Again, the radio installation is extracted from Okuhara et al. [49] and is estimated to an acceptable degree. Deviations in this estimation result in an offset in the estimation. This also relates to IMU misalignment and computation of NED frame relative to the ECEF frame. However, according to the biases, it is difficult to determine whether this is a considerable issue. Lastly, the GTSAM framework is comprehensive, with much happening underneath the interface, including their assumption. The use and understanding of the GTSAM framework is a limiting factor in what can be done and analysed.

6.5.2 Optimality of algorithms

One of the key differences between the two alternatives is that the FGO iterates multiple times when optimising, while MEKF optimises only once. This is also concluded by Wen et al. and Chang et al. [41, 42] as an essential factor for FGO's high precision. Additionally, MEKF and FGO are different methods in their formulations, but despite their diversity, they produce highly similar estimations of the UAV states. This implies that both optimisation formulations are close to the same optimal solution. The most significant impact on the resulting estimates is the noisy data.

Furthermore, according to Wen et al., the FGO outperforms the EKF formulation. However, the FGO formulation in this study does not perform better than the MEKF. Therefore, the MEKF have managed to mitigate the limitations in the EKF and ESKF formulations. The FGO propagates IMU measurements on the manifold, which is exact mathematics. Additionally, the MEKF produces highly accurate calculations with the propagation of the quaternion with normalisation. Therefore, the propagation itself should be similar for both techniques. Moreover, the MEKF computes the optimisation on the tangent plane, contrasting with the FGO. This is expected to induce errors in the MEKF estimation, but for minor orientation errors, the nonlinear error kinematics, presented by Sola [5], are close to the linearised kinematics in (3.11), which results in accurate optimisation and correction, as demonstrated by the high precision in the results. Consequently, the MEKF is only optimal as long as the orientation errors are small and all noise is Gaussian. Once this is not the case, the MEKF will struggle considerably with finding the optimal solution. On the other hand, FGO handles non-Gaussian noise well, according to Dellaert and Kaess [23, Ch. 7.4].

This thesis finds better estimates in the North and East direction than the work done by Albrektsen et al. and Gryte et al. [22, 46]. However, the estimates of the Down direction are far from optimal compared to Gryte et al., who compute sub-meter precision. Also, Rapp [50] concludes on the challenge of estimating the Down axis with PARS. Although, the high precision in the Down direction by Gryte et al. is because they utilised an external height measuring unit, which, according to their impressive accuracy, is strongly advised.

6.5.3 Outlier sensitivity and robustness

Several studies [41–44] reported that one of the many benefits of FGO is its robustness to outliers. However, with the traditional least square formulation, the FGO is shown to be highly sensitive to them. PARS-aided INS prove difficult for both algorithms and requires additional sensors for improved accuracy, as implemented by Albrektsen et al. and Gryte et al. [39, 40]. However, FGO has promising methods to increase the robustness compared to the MEKF. Different cost functions are formulated to solve this problem, such as Huber or Turkey loss [23, Ch. 7.4]. GTSAM also provides this; however, GTSAM does not yet support cross terms in the covariances when applying robust formulations. Hence, a tightly coupled PARS factor implementation is necessary to utilise it. Alternatively, another framework could be considered to utilise robust loss function with loosely coupled PARS measurements, such as WOLF [40]. Still, this makes the FGO more flexible than the MEKF. Nonetheless, ultimately it is expected that FGO should yield better results by introducing outlier rejection based on the chi-squared method and innovation, but this cannot be confirmed or refuted without further implementation and testing, [21].

6.5.4 Real-time applicability

FGO filtering has a short optimisation time listed in Table 6.4. This ensures it is applicable for real-time systems without meeting challenges with time constraints. The lag can be increased, which will take longer, but finding a fixed lag that holds the real-time constraints and includes historical data is favourable. However, when using the traditional LS formulation, there is no motivation to apply a larger lag, as discussed in Section 6.4.

Furthermore, the total optimisation time is considerable when increasing the lag, and therefore extensive lags are unrealistic for time-constrained real-time estimation problems, especially full batch optimisation. Generally, it is the specific optimisation process that comprise the largest percentage of the total processing time. Therefore, the focus should be on efficient optimisation with a reasonable lag size. Important to note is that the computation time is heavily dependent on the available computational power.

Furthermore, the FGO is flexible with adding sensor measurement constraints to the optimisation problem. Also, the implemented FGO system is integrated with ROS, which simplifies its conversion into a real-time system.

Chapter 7

Conclusion

In this thesis, the implemented FGO estimation scheme successfully estimates the position and orientation with GNSS-aided INS. Additionally, it achieves satisfactory estimates of the North and East position along with the attitude with PARS-aided INS. These results highlight the competitiveness of the traditional LS formulation of the FGO as an alternative to the MEKF used by the UAV lab today. Compared to the MEKF, the FGO estimates are approximately as accurate estimations.

While FGO is theoretically expected to outperform EKF due to its iterative optimisation process and propagation of IMU measurements on the manifold, the MEKF produces similar results to the FGO. This indicates that the MEKF has successfully addressed the limitations of previous KF algorithms.

PARS proves to be a good alternative for local navigation. However, it is inadequate for estimating the Down direction, making it necessary to utilise additional sensors such as a barometer. Also, the PARS have limitations over long distances, with erroneous data increasing with more considerable distances. Therefore, PARS-aided INS is challenging for both the FGO and MEKF to compute accurate estimates.

Nevertheless, FGO is versatile towards outliers, with opportunities to implement robust cost functions, which is expected to greatly improve the performance. Furthermore, its robustness towards non-gaussian noise and flexibility with adding any measurement at any time into the graph and including history in the optimisation favours the FGO over the MEKF.

Additionally, FGO applies to real-time applications with its rapid computation time and integrated ROS system for sensor data flow. Lastly, the GTSAM framework provides several features for expanding the system but does not yet support cross-covariance when using robust loss function. Therefore, alternative frameworks could be considered, or preferably, a tightly coupled PARS factor can be implemented.

7.1 Further Work

The research demonstrates FGO as a promising alternative to the MEKF. Still, assumptions, limitations, and future research and development must be addressed to enhance the performance. Generally, the estimations schemes should be continuously researched and updated with advancements in the state estimation and optimisation field.

Inertial frame

The predictions should be formulated relative to a non-inertial frame. Therefore, to increase precision, the implemented systems should convert to using the ECEF frame or a non-inertial NED frame.

Gravitation vector

The implemented system should compute the gravitation vector at each iteration for every new estimation and prediction, instead of using a constant gravitation vector.

Tightly coupled PARS factor and robust objective function

To improve the estimation, the GNSS and the PARS should be implemented as tightly coupled systems. Especially, for the FGO, a PARS factor should be created to improve measurement incorporation. This would also make it possible to utilise GTSAM's robust loss functions, which is expected to greatly improve the performance in the presence of outliers.

Computational efficiency

Further research should go into the computation time of the algorithms. Specifically, the optimal lag-to-performance ratio should be found where the FGO meets the real-time constraint but produce the most optimal estimate by including history.

Optimisation algorithms

In this thesis, the Levenberg-Marquardt optimisation method was applied. Alternative methods, such as incremental smoothing and mapping, iSAM, should be investigated to find potential improvements.

Incorporating supplementary sensors

Additional sensors need to be added to improve the accuracy of the estimation of the UAV. Especially, the barometer to minimise the large errors in the Down direction. However, Rapp [50] mentioned that the barometer has difficulties with the environment and measures erroneous data also. Therefore, incorporating an external height measuring unit as Gryte et al. [22] is strongly advised to remove the largest problem with PARS. Other sensors, such as magnetometer and camera, should also be considered.

Multipath mitigation techniques

Continuous research on mitigating multipath propogation is essential as PARS is heavily erroneous in the elevation angle.

Experimental testing

The algorithms should be tested on different experimental data to ensure the state estimations techniques' robustness in different conditions.

Fault tolerance and testing

The FGO implementation in this thesis is not properly tested and fault tolerance should be incorporated.

References

[1] B. W. Randal and T. W. McLain, *Small Unmanned Aircraft Theory and Practice*. Princeton University Press, 2012.

[2] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2nd ed. Wiley, 2021.

[3] J.-H. Won and T. Pany, 'Signal processing,' in *Springer Handbook of Global Navigation Satellite Systems*, ser. Springer Handbooks, P. J. Teunissen and O. Montenbruck, Eds., Springer Cham, 2017. DOI: `10.1007/978-3-319-42928-1`.

[4] L. A. McGee and S. F. Schmidt, 'Discovery of the Kalman filter as a practical tool for aerospace and industry,' Nov. 1985.

[5] J. Solà, *Quaternion kinematics for the error-state kalman filter*, 2017. arXiv: `1711.02508 [cs.RO]`.

[6] 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, Dec. 2006. DOI: `10.1177/0278364906072768`.

[7] J. Shurman, 'Euclidean space,' in *Calculus and Analysis in Euclidean Space*. Cham: Springer International Publishing, 2016, pp. 23–58, ISBN: 978-3-319-49314-5. DOI: `10.1007/978-3-319-49314-5_2`. [Online]. Available: `https://doi.org/10.1007/978-3-319-49314-5_2`.

[8] T. V. Haavardsholm, *A handbook in visual slam*, `https://github.com/tussedrotten/vslam-handbook`, Accessed: 2023-05-25, 2021.

[9] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, 2nd. Artech House, 2013, ISBN: 1608070050.

[10] J. Solà, J. Deray and D. Atchuthan, *A micro lie theory for state estimation in robotics*, 2021. arXiv: `1812.01537 [cs.RO]`.

[11] C. Forster, L. Carlone, F. Dellaert and D. Scaramuzza, 'Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation,' Jul. 2015. DOI: `10.15607/RSS.2015.XI.006`.

[12] E. Brekke, *Fundamentals of Sensor Fusion. Target tracking, navigation and SLAM*. 2021.

[13] *Mean error: Definition*, `https://www.statisticshowto.com/mean-error/`, Accessed: 2023-05-2.

[14] *Standard deviation formula and uses vs. variance*, `https://c3.ai/glossary/data-science/root-mean-square-error-rmse/`, Accessed: 2023-05-2.

[15] *Root mean square error (rmse)*, `https://www.investopedia.com/terms/s/standarddeviation.asp`, Accessed: 2023-05-2.

[16] T. Constructor, *What is post processed kinematic (ppk) survey?* `https://theconstructor.org/surveying/post-processed-kinematic-ppk-survey/562079/`, Accessed: 2023-05-27.

[17] K. Gryte, T. H. Bryne, S. M. Albrektsen and T. A. Johansen, 'Field test results of gnss-denied inertial navigation aided by phased-array radio systems for uavs,' in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 1398–1406. DOI: `10.1109/ICUAS.2019.8798057`.

[18] T. Maruyama, K. Kihira and H. Miyashita, 'Phased arrays,' in *Handbook of Antenna Technologies*, Z. N. Chen, D. Liu, H. Nakano, X. Qing and T. Zwick, Eds. Singapore: Springer Singapore, 2016, pp. 1113–1162, ISBN: 978-981-4560-44-3. DOI: `10.1007/978-981-4560-44-3_37`. [Online]. Available: `https://doi.org/10.1007/978-981-4560-44-3_37`.

[19] Z. N. Chen, D. Liu, H. Nakano, X. Qing and T. Zwick, *Handbook of Antenna Technologies*. 2016, pp. 1114–1153.

[20] C. Urrea and R. Agramonte, 'Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation,' Sep. 2021.

[21] F. Gustafsson, *Statistical Sensor Fusion*, 2nd. Studentlitteratur, 2012.

[22] K. Gryte, T. Bryne and T. Johansen, 'Unmanned aircraft flight control aided by phased-array radio navigation,' *Journal of Field Robotics*, vol. 38, Dec. 2020. DOI: `10.1002/rob.22002`.

[23] F. Dellaert and M. Kaess, 'Factor graphs for robot perception,' *Foundations and Trends in Robotics*, vol. 6, No. 1-2, no. 1–2, pp. 1–139, 2017. DOI: `10.1561/2300000043`.

[24] C. Montella, 'The kalman filter and related algorithms: A literature review,' May 2011.

[25] M. R. Fernandes, G. M. Magalhães, Y. Cáceres and J. B. R. do Val, *Gnss/mems-ins integration for drone navigation using ekf on lie groups*, 2022. arXiv: `2210.02983 [cs.RO]`.

[26] T. Van, T. Van, D. Nguyen, T. Chu Duc and D.-T. Tran, '15-state extended kalman filter design for ins/gps navigation system,' *Journal of Automation and Control Engineering*, vol. 3, pp. 109–114, Jan. 2015. DOI: `10.12720/joace.3.2.109-114`.

[27] J. Crassidis, 'Sigma-point kalman filtering for integrated gps and inertial navigation,' *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 2, pp. 750–756, 2006. DOI: `10.1109/TAES.2006.1642588`.

[28] J. WENDEL, J. METZGER, R. MOENIKES, A. MAIER and G. F. TROMMER, 'A performance comparison of tightly coupled gps/ins navigation systems based on extended and sigma point kalman filters,' *NAVIGATION*, vol. 53, no. 1, pp. 21–31, 2006. DOI: `https://doi.org/10.1002/j.2161-4296.2006.tb00368.x`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2161-4296.2006.tb00368.x`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2161-4296.2006.tb00368.x`.

[29] E. Edwan, J. Zhou, J. Zhang and O. Loffeld, 'A new loosely coupled dcm based gps/ins integration method,' *NAVIGATION*, vol. 59, no. 2, pp. 93–106, 2012. DOI: `https://doi.org/10.1002/navi.7`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/navi.7`. [Online]. Available: `https://onlinelibrary.wiley.com/doi/abs/10.1002/navi.7`.

[30] Q. Zhang, X. Meng, S. Zhang and Y. Wang, 'Singular value decomposition-based robust cubature kalman filtering for an integrated gps/sins navigation system,' *Journal of Navigation*, vol. 68, Feb. 2014. DOI: `10.5194/isprsarchives-XL-3-W1-149-2014`.

[31] E. LEFFERTS, L. Markley and M. SHUSTER, 'Kalman filtering for spacecraft attitude estimation,' Jan. 1982. DOI: `10.2514/6.1982-70`.

[32] V. Madyastha, V. Ravindra, S. Mallikarjunan and A. Goyal, 'Extended kalman filter vs. error state kalman filter for aircraft attitude estimation,' in *AIAA Guidance, Navigation, and Control*

Conference. DOI: 10.2514/6.2011-6615. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2011-6615. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2011-6615.

[33] W. Liu, D. Song, Z. Wang and K. Fang, 'Comparative analysis between error-state and full-state error estimation for kf-based imu/gnss integration against imu faults,' *Sensors*, vol. 19, no. 22, 2019, ISSN: 1424-8220. DOI: 10.3390/s19224912. [Online]. Available: https://www.mdpi.com/1424-8220/19/22/4912.

[34] Y. Yin, J. Zhang, M. Guo, X. Ning, Y. Wang and J. Lu, 'Sensor fusion of gnss and imu data for robust localization via smoothed error state kalman filter,' *Sensors*, vol. 23, p. 3676, Apr. 2023. DOI: 10.3390/s23073676.

[35] L. Markley, 'Attitude error representations for kalman filtering,' *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM*, vol. 26, pp. 311–317, Mar. 2003. DOI: 10.2514/2.5048.

[36] G. Bourmaud, R. Mégret, A. Giremus and Y. Berthoumieu, 'Discrete extended Kalman filter on lie groups,' in *European Signal Processing Conference*, Marrakech, Morocco, Sep. 2013. [Online]. Available: https://hal.science/hal-00903252.

[37] G. Bourmaud, R. Mégret, A. Giremus and Y. Berthoumieu, 'From Intrinsic Optimization to Iterated Extended Kalman Filtering on Lie Groups,' *Journal of Mathematical Imaging and Vision*, vol. 55, no. 3, Jul. 2016. DOI: 10.1007/s10851-015-0622-8. [Online]. Available: https://hal.science/hal-01311169.

[38] V. Indelman, S. Williams, M. Kaess and F. Dellaert, 'Factor graph based incremental smoothing in inertial navigation systems,' pp. 2154–2161, 2012.

[39] V. Indelman, S. Williams, M. Kaess and F. Dellaert, 'Information fusion in navigation systems via factor graph based incremental smoothing,' *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 721–738, 2013, ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2013.05.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188901300081X.

[40] J. Sola, J. Vallve, J. Casals, J. Deray, M. Fourmy, D. Atchuthan, A. Corominas-Murtra and J. Andrade-Cetto, *Wolf: A modular estimation framework for robotics based on factor graphs*, 2022. arXiv: 2110.12919 [cs.RO].

[41] S.-F. Ch'ng, A. Khosravian, A.-D. Doan and T.-J. Chin, 'Outlier-robust manifold pre-integration for ins/gps fusion,' pp. 7489–7496, 2019. DOI: 10.1109/IROS40897.2019.8967643.

[42] W. Wen, T. Pfeifer, X. Bai and L.-T. Hsu, 'Factor graph optimization for gnss/ins integration: A comparison with the extended kalman filter,' *NAVIGATION*, vol. 68, no. 2, pp. 315–331, 2021. DOI: https://doi.org/10.1002/navi.421. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/navi.421. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/navi.421.

[43] S. Das, R. Watson and J. Gross, *Review of factor graphs for robust gnss applications*, 2022. arXiv: 2112.07794 [cs.RO].

[44] S. Zhao, Y. Chen and J. A. Farrell, 'High-precision vehicle navigation in urban environments using an mem's imu and single-frequency gps receiver,' *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2854–2867, 2016. DOI: 10.1109/TITS.2016.2529000.

[45] M. Okuhara, T. H. Bryne, K. Gryte and T. A. Johansen, 'Phased array radio navigation system on uavs: Gnss-based calibration in the field,' in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 210–218. DOI: 10.1109/ICUAS51884.2021.9476807.

[46] S. M. Albrektsen, T. H. Bryne and T. A. Johansen, 'Robust and secure uav navigation using gnss, phased-array radio system and inertial sensor fusion,' in *2018 IEEE Conference on Control Technology and Applications (CCTA)*, 2018, pp. 1338–1345. DOI: `10.1109/CCTA.2018.8511354`.

[47] S. M. Albrektsen, T. H. Bryne and T. A. Johansen, 'Phased array radio system aided inertial navigation for unmanned aerial vehicles,' in *2018 IEEE Aerospace Conference*, 2018, pp. 1–11. DOI: `10.1109/AERO.2018.8396433`.

[48] M. Okuhara, T. Bryne, K. Gryte and T. Johansen, 'Phased array radio navigation system on uavs: Gnss-based calibration in the field,' Jul. 2021. DOI: `10.1109/ICUAS51884.2021.9476807`.

[49] M. Okuhara, T. H. Bryne, K. Gryte and T. A. Johansen, 'Phased Array Radio Navigation System on UAVs: In-flight Calibration,' Nov. 2022. DOI: `10.36227/techrxiv.21456615.v1`. [Online]. Available: `https://www.techrxiv.org/articles/preprint/Phased_Array_Radio_Navigation_System_on_UAVs_In-flight_Calibration/21456615`.

[50] C. Rapp, *Unmanned aerial vehicle positioning using a phased array radio and gnss independent sensors*, 2019. [Online]. Available: `https://www.diva-portal.org/smash/record.jsf?pid=diva2%5C%3A1323465&dswid=-4561`.

[51] *Matlab*, `https://se.mathworks.com/products/matlab.html`, Accessed: 2023-05-2.

[52] *Standard c++*, `https://isocpp.org/`, Accessed: 2023-05-2.

[53] *Gtsam*, `https://gtsam.org/`, Accessed: 2023-05-2.

[54] *Ros*, `https://www.ros.org/`, Accessed: 2023-05-2.

[55] *Ubuntu*, `https://ubuntu.com/`, Accessed: 2023-05-2.

[56] *Stim300*, `https://www.sensonor.com/products/inertial-measurement-units/stim300/`, Accessed: 2023-06-07.

[57] *Zed-f9p module*, `https://www.u-blox.com/en/product/zed-f9p-module#Product-Selection`, Accessed: 2023-06-07.

[58] *Cre2-189*, `https://radionor.no/product/cre2-189-2/`, Accessed: 2023-06-07.

[59] M. Okuhara, T. H. Bryne, K. Gryte and T. A. Johansen, 'Phased array radio navigation system on uavs: Gnss-based calibration in the field,' in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 210–218. DOI: `10.1109/ICUAS51884.2021.9476807`.

[60] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, 4th. Hoboken, New Jersey: John Wiley & Sons, Inc., 2012.

[61] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*, 1st ed. USA: McGraw-Hill, Inc., 2008, ISBN: 0071493298.

[62] J. A. Farrell, F. O. Silva, F. Rahman and J. Wendel, 'Inertial measurement unit error modeling tutorial: Inertial navigation system state estimation with real-time sensor calibration,' *IEEE Control Systems Magazine*, vol. 42, no. 6, pp. 40–66, 2022. DOI: `10.1109/MCS.2022.3209059`.

[63] *Interpolating states with consecutive combinedimufactor*, Accessed: 2023-04-26, 2018. [Online]. Available: `https://groups.google.com/g/gtsam-users/c/WskPzqexiNk/m/1mjY0VQLBwAJ`.

Appendix A

Summary of Related Work

This appendix provides a complete summary of the reviewed literature in this thesis.

A.1 Kalman Filtering

Table A.1 lists the articles reviewed in this thesis relating to kalman filtering.

Table A.1: List of articles reviewed in the meta-analysis on Kalman Filtering.

Study	**Year**	**Focus Area**	**Sensors**	**Source**
Corey Montella	2011	KF theoretical review	-	[24]
Van et al.	2015	EKF	GNSS-aided INS	[26]
Jogn L. Crassidis	2006	Sigma-Point KF	GNSS-aided INS	[27]
Wendel et al.	2006	EKF, Sigma-Point KF	Tightly coupled GNSS aided INS	[28]
Edwan et al.	2012	Unscented KF	GNSS aided INS	[29]
Zhang et al.	2014	KF	GNSS-aided INS	[30]
Joan Sola	2017	ESKF	GNSS aided INS	[5]
Madyastha et al.	2012	EKF, ESKF	Attitude sensors	[32]
Liu et al.	2019	ESKF	GNSS aided INS	[33]
Yin et al.	2023	ESKF	GNSS aided INS	[33]
Landis Markley	1983	KF, Attitude representation	Attitude sensors	[31]
Landis Markley	2003	KF, Attitude representation	-	[35]
Bourmaud et al.	2013	EKF on manifold	Camera	[36]
Bourmaud et al.	2016	EKF on manifold	Camera	[37]
Fernandes et al.	2022	EKF on manifold	GNSS/MEMS-aided INS	[25]

A.2 Factor Graph Optimization

Table A.2 lists the articles reviewed in this thesis relating to FGO.

Table A.2: List of articles reviewed in the meta-analysis on FG estimation.

Study	**Country**	**Focus Area**	**Sensors**	**Source**
Forster et al.	2015	IMU on manifold	IMU, Camera	[11]
Indelman et al.	2012	Incremental smoothing	IMU, GPS, Camera	[38]
Indelman et al.	2013	Incremental smoothing	IMU, GPS, Camera	[39]
Dellaert & Kaess	2006	FG Theory	-	[6]
Dellaert & Kaess	2017	FG review	-	[23]
Chang et al.	2019	Robust FG	GNSS-aided INS	[41]
Wen et al.	2021	FG vs EKF	GNSS-aided INS	[42]
Zhao et al.	2016	MAP optimization	GNSS-aided IMU	[44]
Das et al.	2016	FG review	GNSS	[43]
Sola et al.	2022	FG framework WOLF	-	[40]

A.3 PARS-aided Navigation

Table A.3 lists the articles reviewed in this thesis relating to PARS-aided navigation.

Table A.3: List of articles reviewed in the meta-analysis on PARS-aided navigation. Mostly completed work by the UAV lab at NTNU in Norway.

Study	Country	Year	Focus Area	Sensors	Estimation	Source
Albrektsen et al.	Norway	2017	UAV pose estimation	PARS Barometer Magnetometer	None	[45]
Albrektsen et al.	Norway	2018	UAV pose estimation	PARS, INS Bareometer Magnetometer	Nonlinear Observer	[47]
Albrektsen et al.	Norway	2018	UAV pose estimation	GNSS, PARS INS Bareometer Magnetometer	Nonlinear Observer	[46]
Gryte et al.	Norway	2019	UAV pose estimation	GNSS, PARS INS Bareometer Magnetometer	MEKF	[17]
Gryte et al.	Norway	2020	UAV pose estimation	GNSS, PARS INS Bareometer Magnetometer	MEKF	[22]
Okuhara et al.	Norway	2021	PARS pose calibration	GNSS, PARS	MEKF	[48]
Okuhara et al.	Norway	2022	PARS pose Calibration	GNSS, PARS	MEKF	[49]
Carl Rapp	Sweden	2019	PARS pose estimation	PARS,IMU, barometer	KF	[49]

Appendix B

Experimental Configuration

The IMU installation relative to the body frame according to the rotation matrix

$$\boldsymbol{R_{mb}} \approx \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{B.1}$$

The PARS ground radio antenna is oriented according to the rotation matrix,

$$\boldsymbol{R_{nr}} \approx \begin{bmatrix} 0.2655561 & 0.9640954 & -0.0000244 \\ -0.9640954 & 0.2655561 & -0.0000698 \\ -0.0000608 & 0.0000421 & 1.0000000 \end{bmatrix} \tag{B.2}$$

which is extracted from estimated orientation based on a calibration algorithm, [48], producing the Euler angles $\boldsymbol{\Theta}_{nr} = [0.004, -0.0014, -74.6]$ in degrees. The NED frame centred at the radio ground antenna is oriented relative to the ECEF frame according to

$$\boldsymbol{R_{en}} \approx \begin{bmatrix} -0.8833 & -0.1666 & -0.4382 \\ -0.1493 & 0.9860 & -0.0740 \\ 0.4444 & 0 & -0.8958 \end{bmatrix} \tag{B.3}$$

Appendix C

Code

All code produced in this thesis is located at GitLab in the UAV lab's repository, specifically in the branch **trym-branch**. For questions or access to the code, refer to the UAV lab at NTNU or the supervisor Torleiv Håland Bryne.

Appendix D

Results

This appendix provides a complete set of plots for all experimental results produced in this thesis. Firstly, the GNSS-aided INS results are given, followed by the PARS-aided INS, with and without outlier rejection.

D.1 MEKF and FGO with GNSS-aided INS

2D and 3D position estimates are provided in Figures D.1 and D.2, attitude is depicted in Figure D.3, and position error and attitude error are given in Figures D.4 and D.5, respectively. Lastly, bias estimates are provided in Figure D.6.

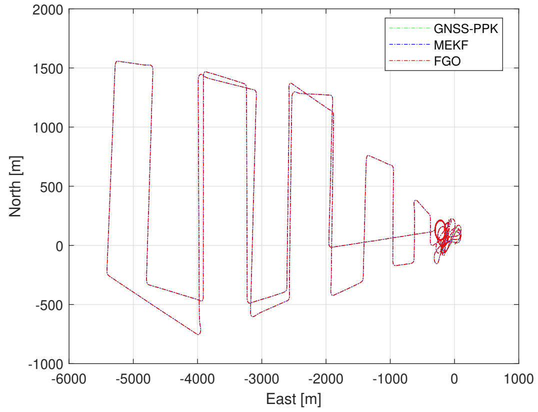

Figure D.1: Position estimates by the MEKF and FGO algorithms with GNSS as input, plotted with the GNSS-PPK in 2D.

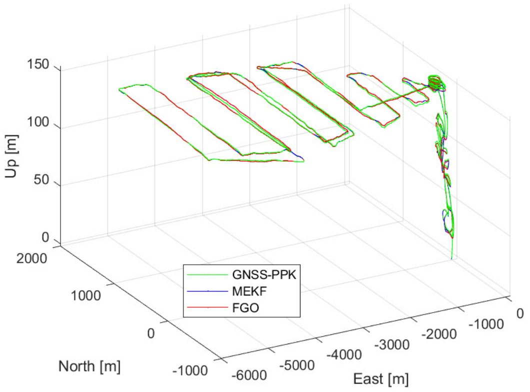

Figure D.2: Position estimates by the MEKF and FGO algorithms with GNSS as input, plotted with the GNSS-PPK in 3D.

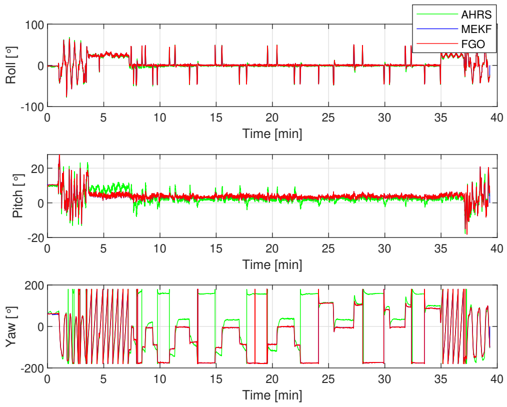

Figure D.3: Attitude estimates by MEKF and FGO, plotted with the AHRS using GNSS as input.

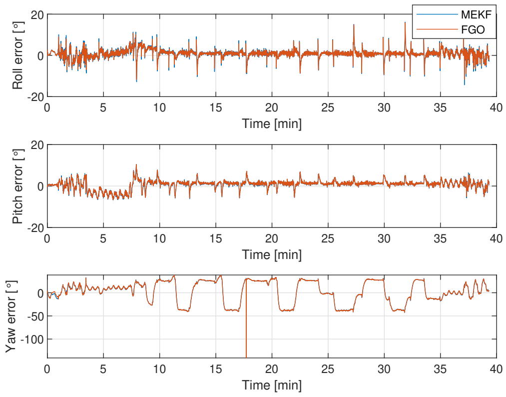

Figure D.4: Attitude error by MEKF and FGO relative the AHRS with GNSS measurements as input.

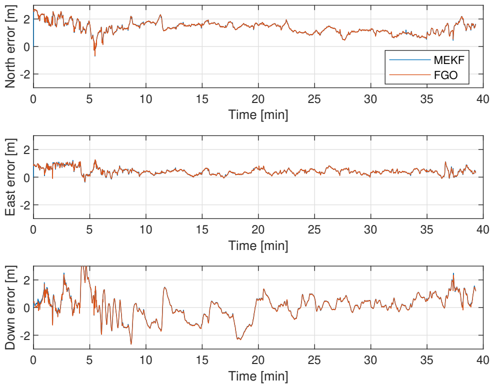

Figure D.5: Position error by MEKF and FGO estimates relative to GNSS-PPK in North, East and Down direction.

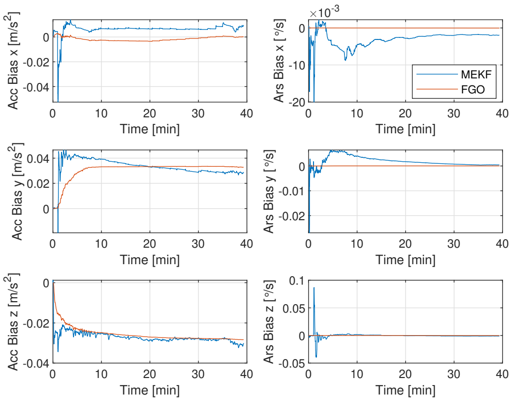

Figure D.6: Bias estimates by MEKF and FGO algorithms with GNSS as measurement input.

D.2 MEKF and FGO with PARS-aided INS

This section provide full set for plots for the results for PARS-aided INS.

D.2.1 Without Validation Gate

Without a validation gate, the 2D and 3D positions are depicted in Figures D.7 and D.8, attitude is depicted in Figure D.9, position error and attitude error are given in Figures D.10 and D.11, respectively, and bias estimates are provided in Figure D.12.

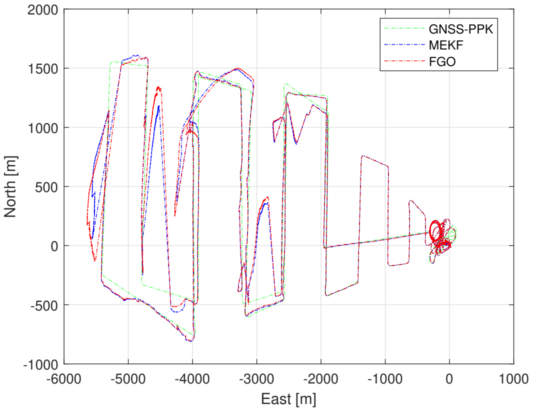

Figure D.7: Position estimates by the MEKF and FGO algorithms with PARS, without validation gate, as input, plotted with the GNSS-PPK in 2D.

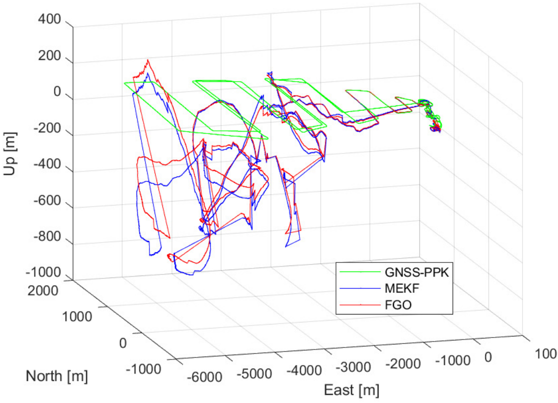

Figure D.8: Position estimates by the MEKF and FGO algorithms with PARS, without validation gate, as input, plotted with the GNSS-PPK in 3D.

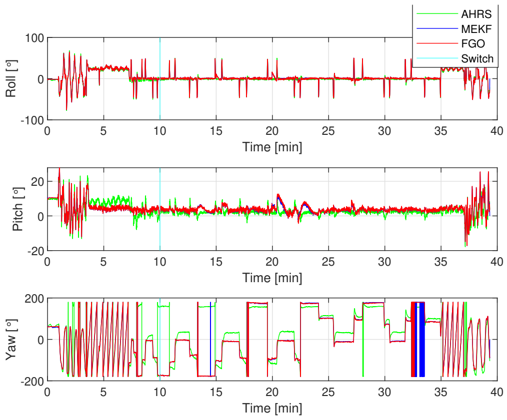

Figure D.9: Attitude estimates by MEKF and FGO, plotted with the AHRS using PARS, without validation gate, as input.

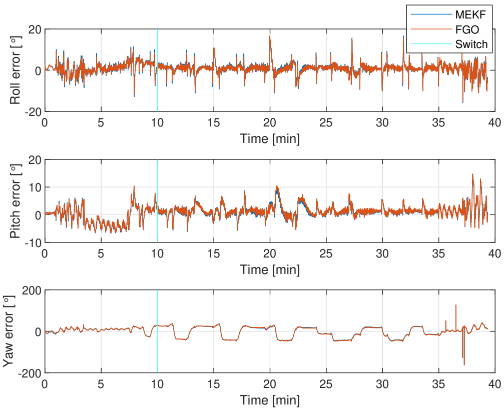

Figure D.10: Attitude error by MEKF and FGO relative the AHRS with PARS measurements, without validation gate, as input.

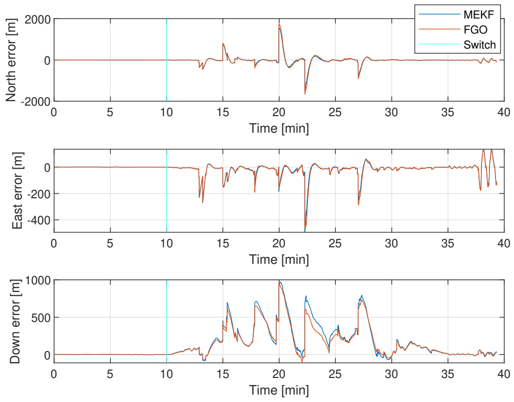

Figure D.11: Position error by MEKF and FGO estimates relative to GNSS-PPK in North, East and Down direction, with PARS measurments without validation gate.

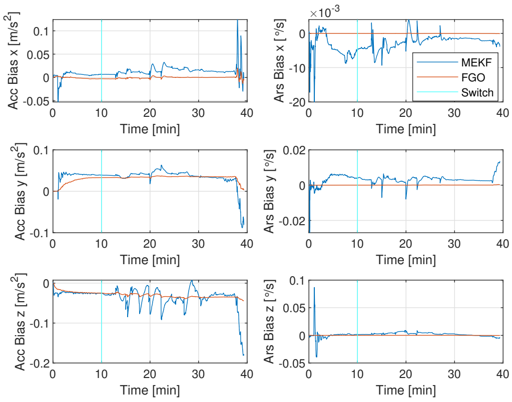

Figure D.12: Bias estimates by MEKF and FGO algorithms with PARS, without validation gate, as measurement input.

D.2.2 With validation gate

Position and attitudes estimates after applying a validation gate with $\chi^2_{\alpha=0.1,\eta_y=3} = 6.251$, are plotted in this section. The 2D and 3D position is provided in Figures D.13 and D.14, attitude is depicted in Figure D.15, and position error and attitude error are given in Figures D.16 and D.17,respectively. Lastly, bias estimates are provided in Figure D.18.

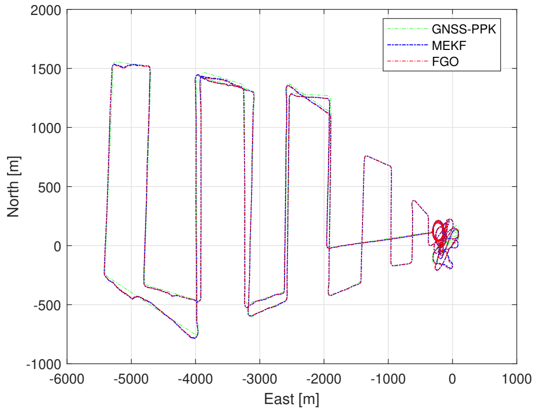

Figure D.13: Position estimates by the MEKF and FGO algorithms with PARS, with validation gate, as input, plotted with the GNSS-PPK in 2D.

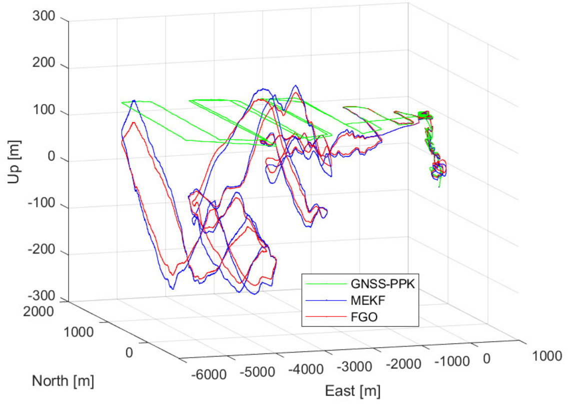

Figure D.14: Position estimates by the MEKF and FGO algorithms with PARS measuremnt, with valdiation gate as input, plotted with the GNSS-PPK in 3D.

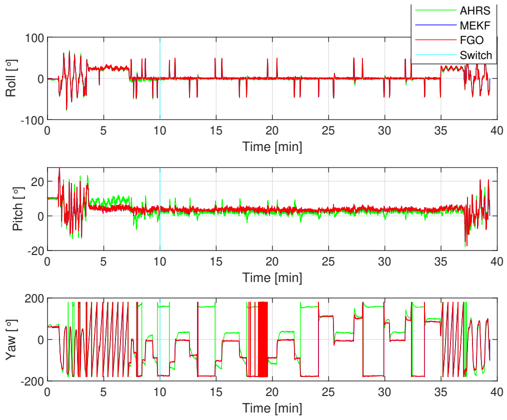

Figure D.15: Attitude estimates by MEKF and FGO, plotted with the AHRS using PARS, with validation gate, as input.

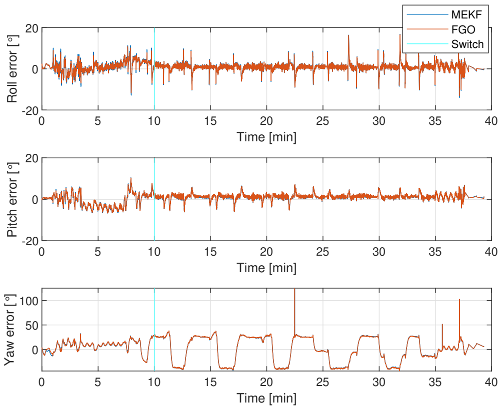

Figure D.16: Attitude error by MEKF and FGO relative the AHRS with PARS measurements, with validation gate, as input.

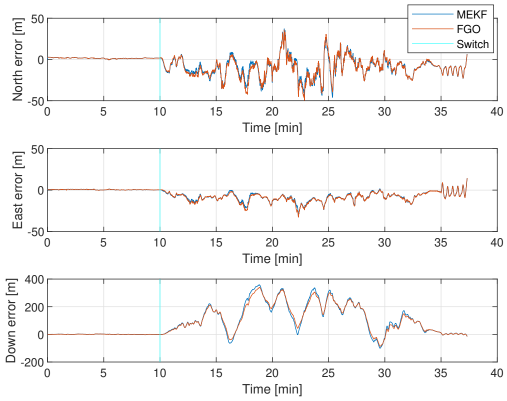

Figure D.17: Position error by MEKF and FGO estimates relative to GNSS-PPK in North, East and Down direction using PARS with validation gate as input.

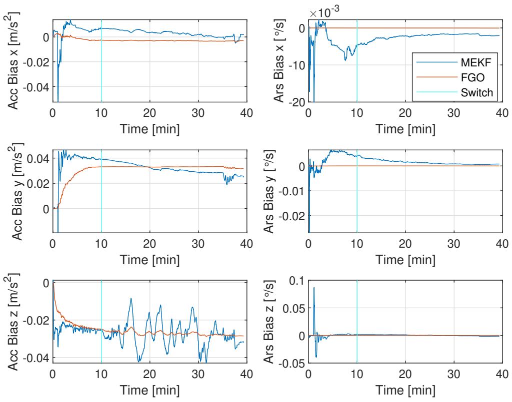

Figure D.18: Bias estimates by MEKF and FGO algorithms with PARS, with validation gate, as measurement input.

NTNU

Norwegian University of
Science and Technology