



NTNU – Trondheim
Norwegian University of
Science and Technology

Singularity-Free Navigation System for an Autonomous Unmanned Aerial Vehicle

Erik Falmår Wilthil

Master of Science in Cybernetics and Robotics

Submission date: June 2015

Supervisor: Edmund Førland Brekke, ITK

Co-supervisor: Kristian Muri Knausgård, Kongsberg Defence Systems

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

For unmanned aerial vehicles (UAVs) to take off and land on limited areas vertical take-off and landing capabilities can be greatly beneficial. Such a system is implemented in Kongsberg's LocalHawk project.

A key challenge in vertical landing is vertical position and velocity estimation. Some airframe designs take-off and land with a 90 degree pitch. While the singularity could be solved by an alternative mechanization, the developmental nature of the project inspire a singularity free implementation.

In this thesis a singularity-free navigation system for the LocalHawk UAV is developed. This is done in the framework of the error-state formulation of the extended Kalman filter, which allows for a modular system where sensors and states can be added and removed easily. The error state formulation allows the higher-rate strapdown computer to work independently of the Kalman filter, and a navigation solution can be computed when the aiding sensors are unable to provide measurements. The attitude is represented by means of a direction cosine matrix, which is a singularity free representation, and vector measurements are used for the attitude. In addition, the added benefit from tightly coupled global navigation satellite systems (GNSS) improve the solution during periods of reduced signal coverage, particularly during maneuvers.

Simulations indicate that it may be possible to land vertically with onboard sensors. Landing without aiding sensors caused too large uncertainty in the estimates, and should not be attempted.

Sammendrag

Vertikal take-off og landing er et viktig verktøy for ubemannede luftfartøy (UAV). Med reduserte krav til utplasseringsområde blir bruksområdet utvidet betraktelig. Et slik take-off system er implementert på Kongsbergs LocalHawk UAV.

Nøyaktig beregning av vertikal posisjon og hastighet er en nøkkelutfordring på dette området. Enkelte flytyper tar av med nesen mot himmelen. Dette krever et godt estimat av orientering, og singulariteten i eulervinklene må unngås.

Denne masteroppgaven omhandler utvikling og simulering av et singularitetsfritt navigasjonssystem for LocalHawk-prosjektet. Dette gjøres ved hjelp av et utvidet Kalmanfilter. Orienteringen representeres ved en rotasjonsmatrise, og vektormålinger brukes for orientering. Det er også tatt i bruk satelittbasert navigasjon, hvor en tett kobling gir økt ytelse i svake signalforhold, særlig under manøvre.

Simuleringer indikerer at det kan være mulig å lande vertikalt med sensorkonfigurasjonen på flyet, men nærmere testing og undersøkelser må gjennomføres før dette kan fastslås. Landing uten støttesensorer anbefales ikke, siden usikkerheten i estimatet øker.

Preface

This thesis is submitted as the final requirement of the MSc. degree at the Norwegian University of Science and Technology (NTNU).

First, I would like to thank my advisers Edmund Fjørland Brekke at NTNU and Kristian Muri Knausgård at Kongsberg Defence Systems (KDS) for their support the last year. I would also like to thank KDS for letting me write this thesis for the LocalHawk project as an extension of my summer internship. I have learned a lot the last year, and I hope the results prove valuable to the project.

Finally, I would like to thank the 2015 graduate class of the department of engineering cybernetics for five exciting, challenging and demanding years.

Table of Contents

Abstract	i
Sammendrag	iii
Preface	v
I Background Material	1
1 Introduction	3
1.1 The LocalHawk Project	3
1.2 Thesis motivation	3
1.2.1 Extension of previous work	4
1.3 Navigation requirements	4
1.3.1 Vertical take-off and landing designs	4
1.3.2 Navigation data requirements	6
1.4 Sensor configuration	7
1.4.1 Inertial measurement unit	7
1.4.2 Global Navigation Satellite System	7
1.4.3 Altimeters	8
1.4.4 Magnetometers	8
1.4.5 Vision systems	8
1.4.6 Sensor configuration	8
2 Litterature review	11
2.1 Kalman filtering	11
2.1.1 Nonlinear extensions	11
2.1.2 Total- and error-state Kalman filters	12
2.2 Singularity-free navigation	13
2.3 Nonlinear observers	13
2.4 GNSS under reduced conditions	14
2.5 Relative navigation and obstacle detection	14

3	Background theory	15
3.1	Coordinate frames	15
3.1.1	Earth-centered inertial frame	15
3.1.2	Earth-centered earth-fixed frame	15
3.1.3	Local navigation frame	16
3.1.4	Body frame	16
3.1.5	IMU frame	18
3.2	Kalman Filtering	18
3.2.1	Nonlinear variations	19
3.3	Mathematical review	20
3.3.1	Skew-symmetric form	20
3.3.2	Matrix exponential	20
II	Methods	23
4	States and Sensor Measurements	25
4.1	Navigation states	25
4.1.1	Error states	25
4.2	Inertial measurement unit	26
4.2.1	Measurement equations	26
4.2.2	Sensor states	27
4.3	Barometer	27
4.3.1	Measurement equation	29
4.4	Magnetometer	29
4.4.1	Measurement equation	29
4.4.2	Magnetometer errors	29
4.5	Leveling	30
4.5.1	Measurement equation	30
4.6	Global navigation satellite system	30
4.6.1	Measurement equations	31
4.6.2	Sensor states	33
4.7	Overview	33
5	Filter Implementation	35
5.1	Filter structure	35
5.1.1	State transition model	35
5.1.2	Measurement Jacobians	36
5.2	Inertial measurement unit	36
5.2.1	Strapdown algorithm	36
5.2.2	State transition matrix	37
5.3	Barometer	39
5.3.1	Jacobian	39
5.3.2	State transition model	40
5.4	Magnetometer	40
5.4.1	Jacobian	40
5.5	Leveling	41

5.5.1	Jacobian	41
5.5.2	Acceleration calculation	41
5.6	GNSS	42
5.6.1	Jacobian	42
5.6.2	State transition matrix	44
5.6.3	GNSS Standalone Solution	44
5.7	Filter Reset	46
III Results		49
6	System Testing and Results	51
6.1	Sensor measurement generation	51
6.1.1	IMU	51
6.1.2	Barometer	53
6.1.3	Magnetometer	53
6.1.4	GNSS	53
6.1.5	GNSS realization	55
6.2	Initialization test	56
6.2.1	Setup and parameters	56
6.2.2	Results	57
6.3	Reduced GNSS signal test	59
6.3.1	Setup and parameters	59
6.3.2	Results	60
6.4	Landing test	63
6.4.1	Setup and parameters	63
6.4.2	Results	63
7	Simulink Models	69
IV Discussion and Conclusion		75
8	Discussion	77
8.1	Initialization	77
8.2	Reduced GNSS signals	77
8.3	Landing and dead reckoning	78
8.4	Further work	79
9	Conclusion	81
Appendices		83
A	Coordinate Transformations	85
A.1	Earth-centered inertial and earth-centered earth-fixed coordinates	85
A.2	ECEF cartesian and ellipsoidal coordinates	85
A.3	ECEF to NED	86
A.4	NED to body	87

TABLE OF CONTENTS

A.5 Transposition of navigation solutions 87

Nomenclature

Acronyms

DCM	Direction Cosine Matrix
EGM08	Earth Gravitational Model 2008
ECEF	Earth-centered Earth-fixed
ECI	Earth-centered Inertial
EKF	Extended Kalman Filter
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GNC	Guidance, Navigation and Control
HDOP	Horizontal Dilution of Precision
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LOS	Line Of Sight
MEKF	Multiplicative EKF
NED	North-east-down Frame
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
VTOL	Vertical Take-off and Landing
WGN	White Gaussian Noise
WMM	World Magnetic Model

Mathematical notation

Let a , b and c be generic coordinate frames with origin \mathcal{O}^a , \mathcal{O}^b and \mathcal{O}^c .

TABLE OF CONTENTS

$\delta\boldsymbol{\psi}_{ab}^c$	Angle error of frame b to a , resolved in c
$\boldsymbol{\omega}, \boldsymbol{\alpha}$	Angular velocity and integrated angular velocity
\mathbf{C}_b^a	Direction cosine matrix from a to b
\mathbf{I}_n	Identity matrix of size n
\mathbf{m}^a	Magnetic field vector resolved in the a -frame
\mathbf{m}	Magnetic field
\mathbf{r}_{ab}^c	Position of b with respect to a resolved in c
p_a	Pressure at \mathcal{O}^a
$\rho, \dot{\rho}$	Pseudorange and pseudorange-rate
ρ_j	pseudorange from user antenna to satellite j
$\dot{\rho}_j$	pseudorange-rate from user antenna to satellite j
\mathbf{R}	Sensor measurement noise covariance
$[\mathbf{u}]^\times$	Skew-symmetric form of \mathbf{u}
$\mathbf{f}, \boldsymbol{\nu}$	Specific force and integrated specific force
\mathbf{Q}	State transition noise covariance
\mathbf{l}_{ab}^c	Unit vector pointing from a to b resolved in c .
\mathbf{v}_{ab}^c	Velocity of b relative to a resolved in c
\mathbf{w}	White gaussian noise

Part I
Background Material

Introduction

1.1 The LocalHawk Project

The LocalHawk student project is a long-term student project managed by Kongsberg Defence Systems (KDS). This project is the foundation for this thesis, and the proposed navigation system will be implemented in the LocalHawk framework. The overall project goal is to develop an unmanned aerial vehicle (UAV) capable of filming a folk derby race autonomously. The UAV will need to take off, track the participants and film them, and land without human aiding. Project specifications have stated that the UAV should be capable of take-off and landing from a limited area. This suggests a UAV capable of vertical take-off and landing (VTOL). This imposes several requirements on the navigation system, particularly in the context of landing. In addition, the airframe design is not finalized and could change in the future.

1.2 Thesis motivation

In order to track the participating race cars safely and effectively, the guidance, navigation and control (GNC) system must be able to give accurate state estimates and control surface deflections. An overview of a GNC system can be seen in Figure 1.1. The guidance system generates reference trajectories based on a predefined goal or inputs from the flight computer, and generates reference signals for the control system. The control system gives actuator values to the UAV control surfaces, which affects the motion of the system. A sensor configuration measures physical quantities related to the UAV motion, and a navigation system provides estimates of the motion for feedback to the control and guidance systems.

Up until this point, a commercial off-the-shelf navigation system has been used for the LocalHawk, the MTi-G from Xsens Technologies [3]. The main motivation for replacing the commercial system is that it is provided as a “black-box” from the manufacturer. This makes it difficult to combine the output with additional sensors, and it is not known how the integration filter weighs the aiding sensor data. Furthermore, the global positioning system (GPS) module does not output raw data, which makes tight integration impossible.

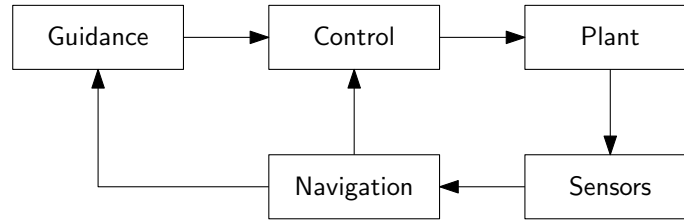


Figure 1.1: A general overview of GNC systems.

1.2.1 Extension of previous work

The LocalHawk project has been running each summer for several years, and the project as a whole has a solid foundation. However, since a commercial navigation system has been used, there were no existing modules for navigation.

The navigation system were also a topic in the project report written during the fall 2014 semester at NTNU [47]. The strapdown equations were tested extensively on rate and implementation methods, and a simple test of inertial navigation system/global navigation satellite system (INS/GNSS) were also conducted.

The work done in this thesis extends the navigation system into a multisensor navigation system, with the addition of a magnetometer and barometer. Methods for generating measurement data and simple kinematic test trajectories were also implemented. Everything is implemented in the LocalHawk framework, enabling the project participants to test and extend the system immediately.

1.3 Navigation requirements

1.3.1 Vertical take-off and landing designs

Several types of aerial vehicles capable of vertical take-off and landing (VTOL) have been designed since the 1950s. For this thesis, they will be divided into two main categories: Lift-off vehicles and tailsitters. It is possible to divide the categories further, and there are several design choices to make with respect to size, payload weight, stability and other criteria. However, for navigation purposes, the categories described below will suffice.

Lift-off vehicles

The common factor of lift-off vehicles is the fact that the attitude of the vehicle body with respect to the ground does not change in the process from takeoff to nominal flight, as seen in Figure 1.2. Generally, there are two ways to achieve this: Thrust vectoring, or separated actuators for upwards and forward thrust. Thrust vectoring is the ability to change the direction of thrust from an actuator, such that it can be directed at will. An example of this is shown in Figure 1.3a, where the four propellers can be tilted to achieve forward thrust. Helicopters are also included in this category, as they change the pitch of the rotor blades to change the thrust vector.

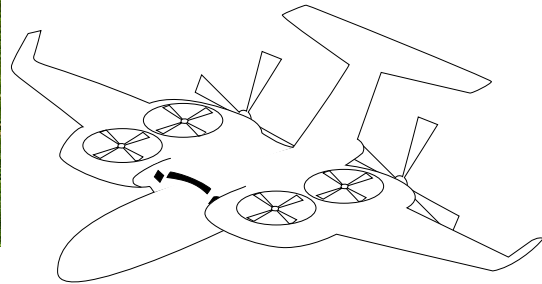
Figure 1.3b shows an example of an aircraft with several actuators, where vertical and horizontal thrust can be independently controlled. Quadcopters also belong to



Figure 1.2: Expected behavior of lift-off vehicles.



(a) The LocalHawk 2014 prototype, by Kongsberg Defence Systems.



(b) Fan-in-wings with fixed thrusters.

Figure 1.3: Concepts for lift-off UAVs.

this category, as the difference in lift from the different propellers can be used to control the attitude and position of the copter.

It should be noted that any directional sensors, for example ultrasonic range finders, attached to lift-off aircraft will point in the same nominal direction during all operation phases. Antennas for data transfer, such as GNSS and telemetry, can also be placed for optimal reception at all times.

Tailsitters

The term “tailsitter” will be used to describe vehicles aligned vertically at takeoff, followed by a transition to horizontal alignment during nominal flight, as seen in Figure 1.4. This means we can expect an attitude change of approximately 90° pitch.

Some tailsitters are incapable of vertical landing, and must employ conventional

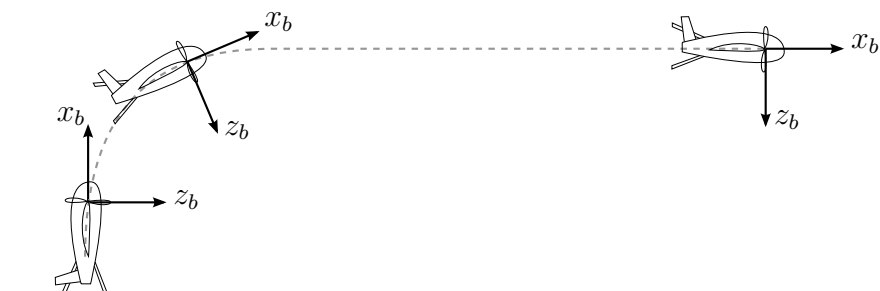


Figure 1.4: Expected behavior of tailsitter vehicles.

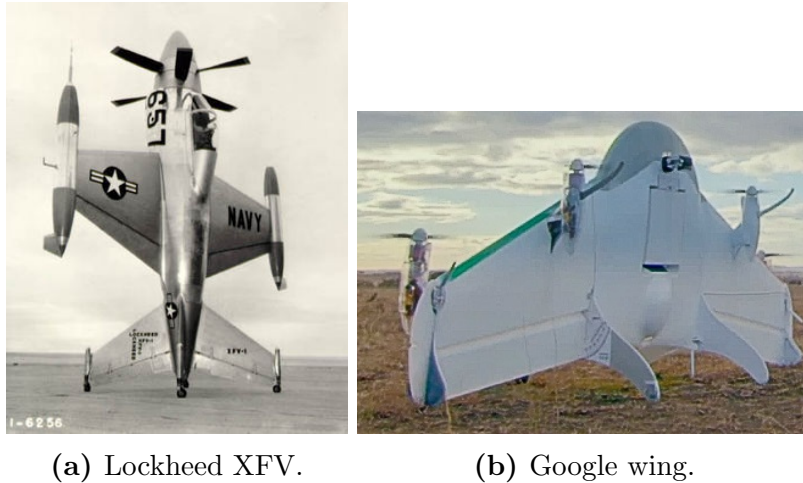


Figure 1.5: Tailsitter designs.

means to land. Others can align the tail back down again, and land back on the tail. Examples can be seen in Figure 1.5. Directional navigation sensors will now change directions during the take-off process, and they might not be able to provide consistent results during the transition phase. Considering that the sensors are fixed to the UAV body, the measurement equations will not change. However, the placement of the sensors must be considered to ensure the data are valid during critical operations, such as landing.

1.3.2 Navigation data requirements

In [6], *navigation* is defined as “Determination of the position, velocity, and, optionally, attitude of a vehicle in a reference coordinate system.” As previously discussed in [47], there are several advantages in using the earth-centered inertial (ECI) coordinate frame for representing the attitude, position and velocity of the UAV. The main benefit of this is that the IMU outputs data relative to the inertial frame, which means the data does not have to be converted to local navigation systems before propagating them through the strapdown system. Figure 1.6 shows some of the sensors and their respective sensor frames. by resolving the navigation states in the ECI-frame, there are no need to transform the measurements to other coordinate frames. Another benefit is that there is no need to track any local navigation frame.

The principal disadvantage of this method is that other systems on the UAV might require data in other formats, such as linear velocity in the body-frame. This is solved by converting the attitude, position and velocity by a series of kinematic transformations, described in Appendix A.

Navigation data accuracy

The operation consist of several phases, each of which impose requirements on the navigation system. Attitude is important during both take-off and landing, since the lack of aerodynamic forces means the actuator thrust forces must be directed properly to ensure a safe take-off and landing. Landing is probably the most critical

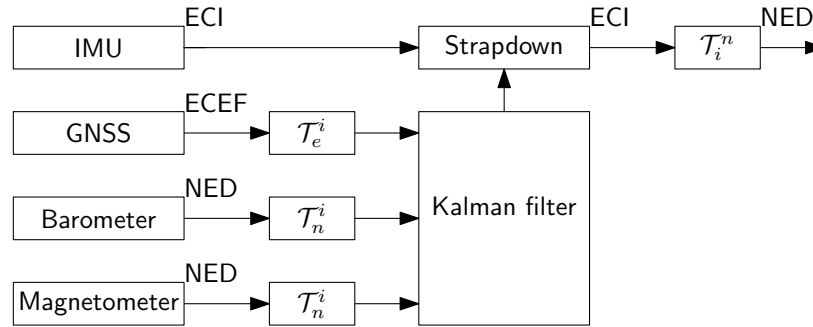


Figure 1.6: Some sensors and their coordinate frames. \mathcal{T} denotes a generic coordinate transformation, which are discussed in Appendix A. The output is resolved in the NED-frame for evaluation by for example the guidance module.

phase, where both vertical position and velocity, as well as horizontal velocity must be accurately determined.

The exact toleration limits for landing autonomously depends on several factors. Too large vertical velocity uncertainty might cause the landing gear to break, which expose the gimbaled camera. Rotorcraft experiments [35] reports successful landing with a vertical velocity between 18 and 35 cm/s, and horizontal velocities in the magnitude of 15cm. The navigation system will need to detect errors in that range if the control system should be able to react on them.

1.4 Sensor configuration

1.4.1 Inertial measurement unit

Inertial measurement units (IMU) are able to provide high-rate measurements of acceleration and angular velocity. These can be processed in a strapdown system to produce estimates of attitude, velocity and position. However, their error characteristics leads to unbounded position and velocity errors, which means they are unsuited for standalone navigation over longer periods of time [23]. Nonetheless, the IMU and strapdown equations provide a solid foundation for the navigation solution, and the additional sensors will correct the long-term errors.

1.4.2 Global Navigation Satellite System

A global navigation satellite system (GNSS) is able to provide an absolute position measurement, and most UAVs operating outdoors are equipped with a receiver [13]. The complementary properties of IMU and GNSS sensor suites means they often are sufficient for UAVs operating without any obstacles [30].

However, GNSS systems are effected by a lot of error sources. Depending on the GNSS used, the horizontal dilution of precision (HDOP) range from 1 to 3 meters at 95% confidence level [50]. Combining different GNSS can lower this to about 0.5 meters, but this require more advanced hardware. This means the accuracy of standalone GNSS is insufficient during landing, and other sensors have to be applied.

1.4.3 Altimeters

Altimeters can be integrated into the system when improved accuracy in the vertical direction is needed. This could be either measurements from a barometer attached to the UAV, or some form of distance measurement from a ranging sensor. These range finders can either be light based, or sound based [30]. Light detection and ranging (LIDAR) have a better range, but are heavier and more power consuming. The sound based ranging sensors are typically smaller and have a more limited range.

1.4.4 Magnetometers

Three-axis magnetometers can be an independent sensor, or contained in an IMU. They measure the magnetic field of the earth, which is useful in determining the heading of the UAV. Together with accelerometers and gyroscopes, magnetometers can be utilized in a complementary filter to determine the attitude of the UAV without other measurement sensors [33]. Additionally, many IMUs come equipped with magnetometers aligned with the accelerometers and gyroscopes, which means all the measurements are readily available.

However, they are not able to produce sufficiently accurate estimates of roll and pitch on its own. In addition, it has been suggested that electric motors and other electric components on small-scale UAVs will distort the magnetic field to the point where the magnetometer measurements are rarely useful [17].

1.4.5 Vision systems

It is possible to calculate translational and angular velocities of an object by using the optical flow from a sequential set of images. There are several methods for calculating optical flow [34], and vision-based autonomous landing of UAVs is a promising research field [35].

1.4.6 Sensor configuration

In view of the sensors above, a suitable sensor configuration will have to be suggested for the LocalHawk. The ADIS 16488 IMU [8] provides measurements of acceleration, angular velocity and magnetometer in three orthogonal axes, as well as pressure data from a barometer. Since most UAVs are equipped with an IMU for acceleration and angular rate measurements, it will be advantageous to use the magnetometer and barometer measurements, despite the challenges described in Section 1.4.4.

A GNSS receiver is almost always present in an UAV [13] due to the ability to provide absolute position measurements. The IMU/GNSS complementary properties make this a good baseline for general navigation [11], and the additional magnetometers and barometer could mean this sensor suite is sufficient for normal operations. Similar UAV setups with a tactical grade IMU, a GNSS receiver and a barometer reports a standard deviation of about 1.5 meters in the down-direction [31]. Other studies report considerably better accuracy with the barometer, down to 10 cm [49]. It will be investigated if the barometer is sufficient for landing.

The survey [30] notes that vision-based navigation is still very active, and a limited amount of tests have been conducted. However, more recent papers [34] has shown promising results on the topic. Since the LocalHawk will be equipped with a camera, the navigation system should be designed such that integration of camera measurements at a later stage is possible.

Litterature review

2.1 Kalman filtering

The Kalman Filter were introduced in 1960 [28], and has since been a popular tool for state estimation in general [11, 12] and navigation applications [23]. The original Kalman filter is a discrete and recursive formulation. Both of these properties are desirable in computer implementations. The Kalman filter assumes a linear system model:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{u}_k \tag{2.1a}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k \tag{2.1b}$$

Where \mathbf{u}_k and \mathbf{w}_k are assumed to be white gaussian noise. If the system in Equation (2.1) is observable and the initial state has a gaussian distribution, the Kalman filter has the following properties [39]:

- The state estimate is unbiased and optimal in the sense of minimum variance.
- The Kalman filter is asymptotically stable.

2.1.1 Nonlinear extensions

The most popular nonlinear extension of the Kalman filter is the extended Kalman filter, which is the traditional workhorse for attitude estimation [15]. The matrices \mathbf{F}_k and \mathbf{H}_k in the system model, Equation (2.1), are replaced by nonlinear functions $\mathbf{f}_k(\mathbf{x}_k)$ and $\mathbf{h}_k(\mathbf{x}_k)$, respectively. These functions are linearized at the current state estimate to yield a linear model that can be used to update the covariance.

Also note that there exists other Kalman filtering techniques for state estimation, such as the sigma-point Kalman filter, also known as the unscented Kalman filter (UKF) Instead of linearization, the UKF propagates a deterministic set of samples through the state space [24]. Experimental results show that EKF and UKF have similar performance for integrated inertial and satellite navigation systems [22, 21, 48, 46].

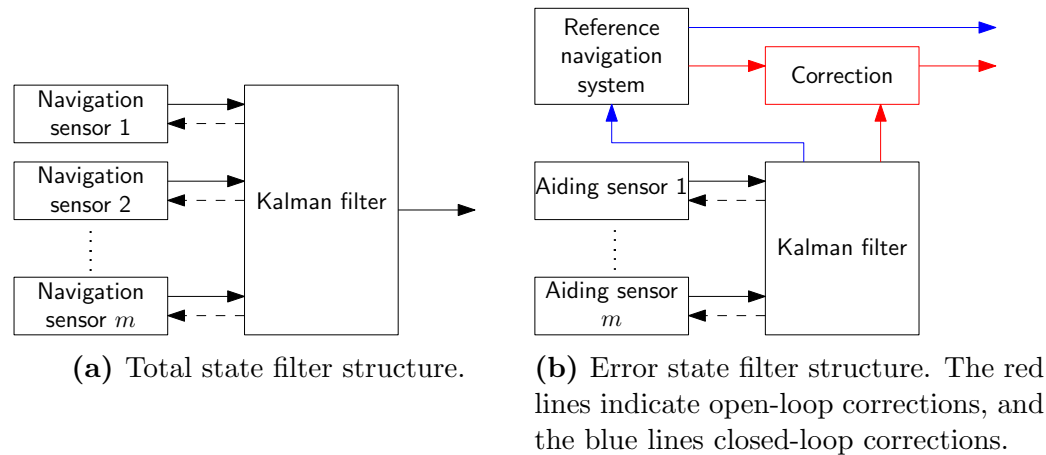


Figure 2.1: General block diagrams of the total and error state filter. The dashed lines indicate optional sensor corrections. Adopted from [23].

2.1.2 Total- and error-state Kalman filters

There are two main strategies for approaching the navigation problem, largely independent of filter type. A total state filter, also known as a direct filter, calculates the navigation state (attitude, velocity and position) of the host vehicle from sensor measurements. In an error state filter, also known as an indirect filter, a single sensor is used as a reference navigation sensor. This could be an IMU, or other dead-reckoning sensors [23]. The reference sensors dead-reckoning capabilities are used to provide an estimated navigation solution. The INS is used as a reference navigation system, and the other sensors provide estimates of the error in the INS solution. These errors can be either fed back to the INS to provide closed-loop corrections or provided as corrections to the INS output as open-loop corrections. The two designs can be seen in Figure 2.1.

The main advantages of the error state formulation are

- There are no need for a dynamic model of the host vehicle, as a dynamic model of the IMU errors are used instead. The accelerometer and gyroscope error model has slower dynamics, and a linear model can be used [40]. This also means that the Kalman filter will be tuned to the sensor model instead of the vehicle model. Changes in the aerodynamic properties of the UAV does not affect the filter.
- The navigation system can be split into a high-rate component, the strap-down system, and a low-rate component, the Kalman filter. This means the high output rate of the IMU can be fully utilized even on platforms with low computational power.
- If the Kalman filter is unable to provide error estimates for some reason, the strapdown solution still provide a navigation solution by integrating the IMU output. This will give the position relative to where the aiding sensors were lost, and is referred to as *dead reckoning* [45].

As indicated in Figure 2.1b, there are two main methods of handling the error state. The estimated error can be applied in an open-loop correction. This means

the error may continue to grow, and sensor errors will not be corrected before the measurements are integrated in the INS. Closed-loop correction remedy this by applying and resetting the error. This means the estimated sensor errors are applied before they are integrated in the INS, and the dynamic model are working around the linearization point.

The main advantage of the total state filter is that the desired quantity, such as position, velocity and attitude, are available directly from the filter without any calculations [43]. A weakness is the increased computational complexity added by integrating the high-rate IMU in the Kalman filter loop. However, the effects can be reduced by applying an asynchronous filter, where only the prediction step is placed in the INS loop [43].

To the author's best knowledge, no error state formulation for UKF has been derived and tested against the EKF. Thus, it is currently not known to what degree an error state UKF would benefit from the points discussed above.

2.2 Singularity-free navigation

It is well known that all three-parameter attitude representations are singular [15, 19]. Using a three-parameter attitude representation for the global attitude will impose restrictions on the navigation system. Although it is possible to avoid the particular singularity of the tailsitter at 90° by choosing a different mechanization than the classical "aerospace" Euler angle sequence, any sequence of rotations will impose constraints on the navigation system. Switching the mechanization is impractical, and thus a singularity-free representation will be sought.

The two most common nonsingular attitude representations are the DCM and the unit quaternion [45]. It has been indicated that the four-parameter quaternion are more computationally efficient than the nine-parameter DCM, although this is a diminishing advantage as computational power increase. DCM manipulations consist of matrix multiplications, which can be more intuitive than the quaternion product. Matrix multiplication is also a native operation in the Matlab/Simulink framework.

A popular approach for extended Kalman filtering in the context of attitude estimation is the multiplicative extended Kalman filter (MEKF) [15]. The idea is to represent the attitude as the composition between a nonsingular global parametrization, such as the quaternion, and a three-parameter local error. Although all three-parameter attitude representations are known to have singularities [15], this is not a problem with local errors.

2.3 Nonlinear observers

More recently, nonlinear deterministic observers have been developed for attitude estimation purposes [33, 20] as an alternative to the EKF. Their stability are proven using Lyapunov-based method. The proofs are mostly done with a DCM, although it is possible to represent the observer with quaternions as well [33].

Another feature is that there are no covariance propagation, which significantly decreases computational requirements [20]. Although this could be an advantage

in low-cost navigation technology, the covariance can provide valuable information about the estimate accuracy. Another weakness of observers is that it is less modular, such that it is nontrivial to add or remove sensors without affecting the Lyapunov candidate solution.

2.4 GNSS under reduced conditions

INS/GNSS integration has seen a lot of research due to the complimentary benefits, and navigation in obstacle-free environments is usually achieved with commercially available systems [30]. This means researchers are focusing more on GNSS-denied navigation.

In GNSS-reduced environments, the effect of combining independent GNSS systems can be significant [9]. Tight coupling benefit from the additional satellites in the second GNSS system, and loosely coupled systems are also improved in environments with moderate availability. This will be demonstrated later in this thesis. When GNSS signals are in abundance, the effect is limited.

2.5 Relative navigation and obstacle detection

Relative navigation by obstacle detection is relevant for autonomous landing, and several studies shows promising experiments by using vision sensors [35, 41, 32]. There are, however, challenges in combining relative and globally referenced states [32]. Note that these reports make use of a rotorcraft UAV which means the results are most easily applied to lift-off vehicles.

Placement of vision sensors on a tail-sitter is usually optimized for nominal flight. A more light-weight detection system, such as an ultrasonic ranging sensor, could be fitted at the tail of the UAV. Ultrasonic sensors have successfully been applied for indoor navigation [44, 14]. Outdoor experiments have also been conducted with a wireless sensor network and an ultrasonic sensor [37].

Background theory

3.1 Coordinate frames

There are several reference frames in use for navigation. This section provides a summary of the definitions. Transformations between them are found in Appendix A.

3.1.1 Earth-centered inertial frame

An earth centered inertial (ECI) frame is an inertial coordinate frame [6]. This means the attitude of the coordinate frames stay fixed: However, the origin of the frame follows the center of the earth as it orbits the sun, which again rotates in the galaxy. The effects of these rotations are usually ignored, since their impact is negligible compared to other error sources [23].

The most commonly used inertial frames defines a z-axis along the earth rotation axis. Some authors [39] define the x-axis to point towards the *vernal equinox*, while others [42] does not put this limitation to frames termed “ECI”. We will define the ECI-frame to coincide with the ECEF-frame described below at system startup, which simplifies the relationship between the ECI-frame and the ECEF-frame. They can be seen in Figure 3.1

3.1.2 Earth-centered earth-fixed frame

The earth-centered earth-fixed (ECEF) frame is a cartesian coordinate system which, as opposed to the ECI-frame, rotates with the earth. We will use the World Geodetic System 1984 (WGS 84) coordinate system as our fixed-earth global system [5]. This standard defines several important parameters, such as the direction of the axes and the parameters of the ellipsoidal approximation of the earth. Key parameters are shown in Table 3.1. The origin of the coordinate frame is defined in the earth’s center of mass. The z-axis coincides with the z-axis of the ECI-frame, while the x-axis goes through a meridian defined by the International Earth Rotation Service (IERS). The y-axis simply complete an orthogonal system [5].

It is also possible to reference ECEF-coordinates as ellipsoidal coordinates, and will be called *geodetic coordinates* to clarify which coordinates are used. The refer-

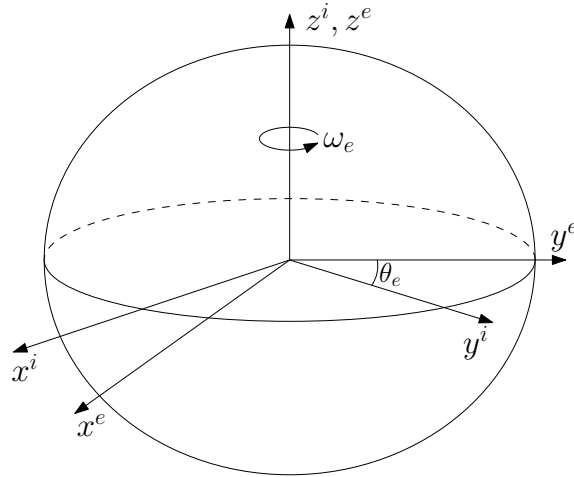


Figure 3.1: The ECEF-frame rotates with the earth, while the ECI-frame is fixed.

Table 3.1: Defining parameters of WGS 84, as well as some useful derived values.

r_e	6378137 m	Equatorial radius
f^{-1}	298.257223563	Reciprocal of flattening
ω_e	7.292115×10^{-5} rad/s	Angular velocity of earth
GM	3986004.418×10^8 m ³ /s ²	Earth's Gravitational constant
$r_p = r_e(1 - f)$	6356752 m	Polar radius
$e = \sqrt{1 - r_p^2/r_e^2}$	0.0818	Eccentricity of ellipse

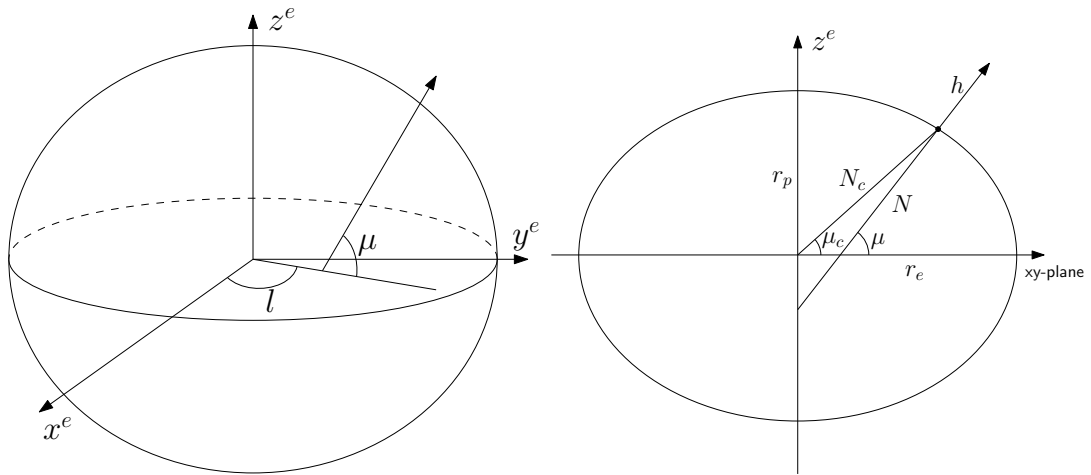
ence ellipsoid is, again, defined in [5]. The geodetic coordinates is frequently used in global navigation, and are defined by two angles, longitude and latitude, as well as a height above the reference ellipsoid. They can be seen in Figure 3.2, where the latitude is denoted as μ , the longitude is denoted as l and the height by h . Note how the *geodetic* latitude is defined differently from the *geocentric* latitude μ_c .

3.1.3 Local navigation frame

The local navigation frame is a cartesian coordinate system, which varies along the location on the earth. The origin follows the UAV, and coincides with the main sensor axes. There are several ways of defining a local navigation frame: Some authors use an east-north-up (ENU) or north-west-up (NWU) frame. However, the most common frame [23] is the north-east-down frame (NED). It has an x-axis pointing towards the geographic north pole, and the z-axis is normal to the ellipsoid at the chosen origin, downwards into the earth. See Figure 3.3 for an illustration. The y-axis completes the right-hand system, and points east.

3.1.4 Body frame

The body frame is a fixed frame with respect to the target vehicle. The center of rotation or a sensor system is conventional choices for the origin [23]. The x , y and z -axes are commonly chosen along the longitudinal, lateral and normal axes of the



(a) Longitude and geodetic latitude angles. (b) The geodetic latitude, normal radius of curvature and height.

Figure 3.2: Relationship between cartesian and ellipsoidal coordinates for the ECEF-frame.

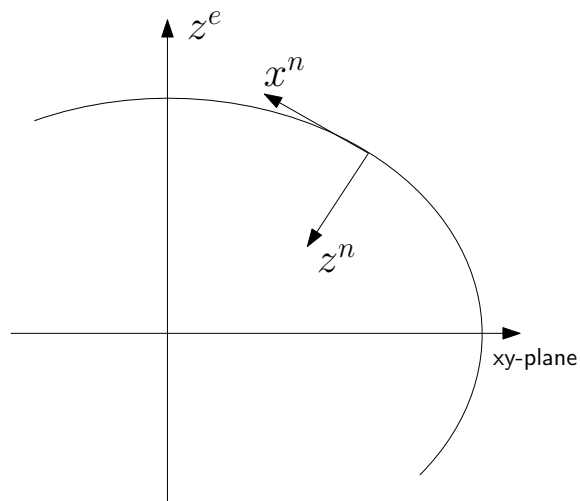


Figure 3.3: The ECEF- and NED-frame relations.

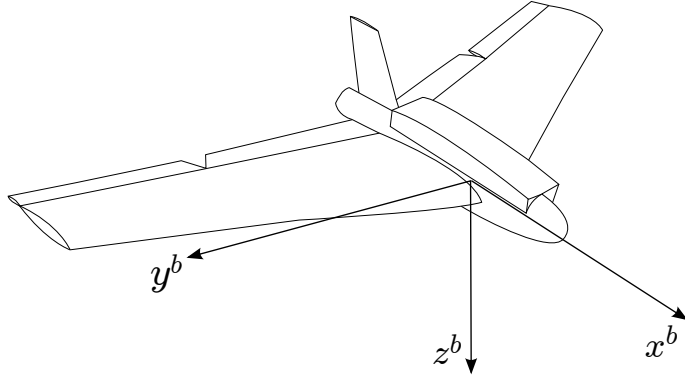


Figure 3.4: The conventional choice of the body frame for an aircraft.

Table 3.2: The coordinate frames used in this thesis.

Frame	abbreviation	index
Earth-centered inertial	ECI	i
Earth-centered Earth-fixed	ECEF	e
Local navigation frame	NED	n
Body frame		b
IMU sensor frame		m

vehicle, as shown in Figure 3.4 [18].

3.1.5 IMU frame

In practical UAV design, it is not always possible to align the IMU with the center of rotation. This is known as size effect [27]. It is convenient to use the IMU frame for the navigation solution, but the measurements will have to be converted to the body frame for guidance and control purposes. This transformation is done by using lever arm compensation as described in Appendix A.

A summary of the coordinate frames are given in Table 3.2

3.2 Kalman Filtering

This section will give the Kalman filter recursion for the system in Equation (2.1), repeated here:

$$\mathbf{x}_{k+1} = \mathbf{F}_k \mathbf{x}_k + \mathbf{u}_k \quad (3.1a)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{w}_k \quad (3.1b)$$

where k is the time index, \mathbf{x}_k is the system state, \mathbf{F}_k is the state transition matrix, \mathbf{z}_k is the measurement and \mathbf{H}_k is the measurement matrix. The vectors \mathbf{u}_k and \mathbf{w}_k are assumed to be white gaussian noise (WGN), which means they have the following

properties:

$$\mathbb{E}[\mathbf{u}_k] = \mathbf{0} \qquad \mathbb{E}[\mathbf{w}_k] = 0 \qquad (3.2a)$$

$$\mathbb{E}[\mathbf{u}_k \mathbf{v}_j] = \begin{cases} \mathbf{Q}_k & k = j \\ \mathbf{0} & k \neq j \end{cases} \qquad \mathbb{E}[\mathbf{w}_k \mathbf{w}_j] = \begin{cases} \mathbf{R}_k & k = j \\ \mathbf{0} & k \neq j \end{cases} \qquad (3.2b)$$

That is, they have zero mean and are uncorrelated in time. The covariances are described by the matrices \mathbf{Q} and \mathbf{R} . The initial state estimate is also assumed to have a gaussian distribution, with expectation $\hat{\mathbf{x}}_{0|0}$ and covariance $\mathbf{P}_{0|0}$.

With the assumptions described above, the optimal estimate of \mathbf{x}_{k+1} given measurements $\mathbf{z}_0, \dots, \mathbf{z}_k$, denoted $\hat{\mathbf{x}}_{k+1|k}$ can be predicted as

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{F}_k \hat{\mathbf{x}}_{k|k} \qquad (3.3)$$

The covariance of this estimate can be predicted as

$$\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{P}_{k|k} \mathbf{F}_k^\top + \mathbf{Q}_k \qquad (3.4)$$

Given a measurement \mathbf{z}_k , the *measurement innovation* is given by the difference between the observed and predicted measurement:

$$\delta \mathbf{z}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \qquad (3.5)$$

The measurement innovation has covariance given by

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k \qquad (3.6)$$

The measurement innovation and the corresponding covariance can be used to compute the Kalman gain \mathbf{K}_k , which can be used to update the state estimate and covariance with the new information:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k \mathbf{S}_k^{-1} \qquad (3.7)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \delta \mathbf{z}_k \qquad (3.8)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\top \qquad (3.9)$$

3.2.1 Nonlinear variations

The extended Kalman filter assumes the following model [11]:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, k) + \mathbf{u}_k \qquad (3.10)$$

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{w}_k \qquad (3.11)$$

The EKF solves the nonlinearities by linearizing the models around the state estimate $\hat{\mathbf{x}}_{k|k}$ and prediction $\hat{\mathbf{x}}_{k|k-1}$:

$$\mathbf{F}_k = \left. \frac{\partial \mathbf{f}(\mathbf{x}_k, k)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} \qquad (3.12)$$

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}(\mathbf{x}_k, k)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k-1}} \qquad (3.13)$$

The computation of the estimate covariance (3.4), measurement innovation (3.6) and the Kalman gain (3.7) can then be done with the linearized Jacobians. It should be noted that the state prediction (3.3) and the measurement prediction in (3.5) can be calculated with the nonlinear functions [11, 39]:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{f}(\hat{\mathbf{x}}_{k|k}, k) \quad (3.14)$$

$$\delta \mathbf{z}_k = \mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}, k) \quad (3.15)$$

3.3 Mathematical review

3.3.1 Skew-symmetric form

Let \mathbf{u} and \mathbf{v} be three-dimensional vectors. The cross product of \mathbf{u} and \mathbf{v} can be written as the following matrix multiplication [16]:

$$\mathbf{u} \times \mathbf{v} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \mathbf{v} = [\mathbf{u}]^\times \mathbf{v} \quad (3.16)$$

where u_i are the elements of \mathbf{u} .

3.3.2 Matrix exponential

This section is based on the results in [36].

The matrix exponential of a matrix \mathbf{A} can be defined by the power series expansion

$$\exp \mathbf{A} = \mathbf{I}_3 + \mathbf{A} + \frac{\mathbf{A}^2}{2} + \dots = \sum_{j=0}^{\infty} \frac{\mathbf{A}^j}{j!} \quad (3.17)$$

In [47], the matrix exponential were calculated by truncating the power series. However, the survey [36] notes that this is not a very precise method.

Since the implementation is done in Matlab/Simulink software, it is desirable to use the methods provided from this framework. Matlab documentation [2] notes that a Padé approximation with scaling and squaring is used. The scaling and squaring part exploits that

$$\exp \mathbf{A} = \left(\exp(\mathbf{A}/2^j) \right)^{2^j} \quad (3.18)$$

which means that it is possible to compute the matrix exponential of $\mathbf{A}/2^j$, and then multiply the solution by itself j times. The value of j is commonly selected such that $\|\mathbf{A}\|/2^j \leq 1$. This ensures the exponential of the resulting matrix can be efficiently computed.

The Padé approximation of the matrix exponential of a matrix is given as

$$\exp \mathbf{A} \approx \mathbf{D}_q^{-1} \mathbf{N}_q \quad (3.19)$$

where

$$\mathbf{N}_q = \sum_{i=0}^q \frac{(2q-i)!q!}{(2q)!i!(q-i)!} \mathbf{A}^i, \quad \mathbf{D}_q = \sum_{i=0}^q \frac{(2q-i)!q!}{(2q)!i!(q-i)!} (-\mathbf{A})^i \quad (3.20)$$

The survey [36] includes a list of tables for selecting the value of q to obtain desired accuracy.

It is also possible to use the Taylor series in Equation (3.17) on the scaled matrix. However, the Padé approximation with proper scaling is a very effective algorithm.

Part II

Methods

States and Sensor Measurements

This chapter defines the states used in the navigation system, as well as the sensor models. This include the measurement equations and error characteristics, as well as the models used for generating sensor data based on the true pose of the UAV.

4.1 Navigation states

The navigation state is given as

$$\mathbf{x}_N = \begin{bmatrix} \mathbf{C}_m^i \\ \mathbf{v}_{im}^i \\ \mathbf{r}_{im}^i \end{bmatrix} \quad (4.1)$$

where \mathbf{C}_m^i , \mathbf{v}_{im}^i and \mathbf{r}_{im}^i is the attitude, velocity and position of the IMU with respect to the ECI-frame. The IMU-frame has been chosen as the main navigation frame. This is advantageous since the IMU is chosen as the reference navigation solution, described below. It is also a fixed point on the airframe, such that changes in the payload does not affect the navigation solution. This is the case with the center of gravity (CG), which could change during operation. An example of this is fuel-based aircraft. However, tilt-rotor aircraft might also have a slight shift of the CG.

It is also seen that the ECI-frame has been chosen as the reference frame. The advantages of this is that there is no need to track the local navigation frame. The reference navigation sensor, the IMU, also outputs data relative to this frame. This simplifies the strapdown algorithm, as the IMU output does not have to be converted.

4.1.1 Error states

To use an error state formulation, a consistent error definition must be decided. Let $\hat{\mathbf{C}}_m^i$, $\hat{\mathbf{v}}_{im}^i$ and $\hat{\mathbf{r}}_{im}^i$ be the attitude, velocity and position estimates from the reference navigation system. The errors are defined as

$$\delta \mathbf{C}_m^i = \mathbf{C}_m^{\hat{m}} = \hat{\mathbf{C}}_m^i \top \mathbf{C}_m^i \quad (4.2)$$

$$\delta \mathbf{v}_{im}^i = \mathbf{v}_{im}^i - \hat{\mathbf{v}}_{im}^i \quad (4.3)$$

$$\delta \mathbf{r}_{im}^i = \mathbf{r}_{im}^i - \hat{\mathbf{r}}_{im}^i \quad (4.4)$$

The velocity and position errors are related by a difference, while the attitude error is represented as an error DCM. Note that the error DCM represents a rotation from the estimated attitude to the true attitude. The small-angle approximation can be used to represent this error as a three-element vector [23]:

$$\begin{aligned}\delta\mathbf{C}_i^m &\approx \mathbf{I}_3 + [\delta\boldsymbol{\psi}_{im}^m]^\times{}^\top \\ &= \mathbf{I}_3 - [\delta\boldsymbol{\psi}_{im}^m]^\times\end{aligned}\quad (4.5)$$

where $\delta\boldsymbol{\psi}_{im}^m$ is the rotation vector from the m -frame to the \hat{m} -frame. This approximation is usually valid in a closed-loop error state filter [23], and the three-parameter vector $\delta\boldsymbol{\psi}_{im}^m$ can substitute the nine-parameter error DCM from Equation (4.2) in the error state. The error angle has no total state counterpart. Due to singularity issues, no global three-parameter attitude representation will be used. This means the error state can be written

$$\delta\mathbf{x}_N = \begin{bmatrix} \delta\boldsymbol{\psi}_{im}^m \\ \delta\mathbf{v}_{im}^m \\ \delta\mathbf{r}_{im}^m \end{bmatrix}\quad (4.6)$$

4.2 Inertial measurement unit

The IMU measures specific force and angular velocity, or their integrated values. These measurements will be propagated through a set of strapdown equations to provide estimates of linear velocity, position and attitude. As previously mentioned, it will be used as the reference navigation sensor. This section describes the measurement model, while Section 5.2.1 provides an overview of the reference navigation system, the strapdown algorithm.

4.2.1 Measurement equations

The output from the IMU can be modeled as

$$\mathbf{f}_I = \mathbf{f}^m + \mathbf{b}_a + \mathbf{w}_a\quad (4.7)$$

$$\boldsymbol{\omega}_I = \boldsymbol{\omega}_{im}^m + \mathbf{b}_g + \mathbf{w}_g\quad (4.8)$$

where \mathbf{f}^m and $\boldsymbol{\omega}_{im}^m$ are the true values of specific force and angular velocity. The biases are assumed to consist of one known part, the bias estimate $\hat{\mathbf{b}}_a$ and $\hat{\mathbf{b}}_g$, and the residual bias $\delta\mathbf{b}_a$ and $\delta\mathbf{b}_g$:

$$\mathbf{b}_a = \hat{\mathbf{b}}_a + \delta\mathbf{b}_a\quad (4.9)$$

$$\mathbf{b}_g = \hat{\mathbf{b}}_g + \delta\mathbf{b}_g\quad (4.10)$$

The measurement noise \mathbf{w}_a and \mathbf{w}_g are assumed to have covariances \mathbf{R}_a and \mathbf{R}_g , where

$$\mathbf{R}_a = \sigma_a^2 \mathbf{I}_3\quad (4.11)$$

$$\mathbf{R}_g = \sigma_g^2 \mathbf{I}_3\quad (4.12)$$

which means the accelerometer and gyroscope triads are assumed to consist of sensors with equal characteristics.

When integrated specific force and angular rate is output from the IMU, the measurements are

$$\begin{aligned}
 \mathbf{v}_I &= \int_t^{t+\Delta t} \mathbf{f}_I \, d\tau \\
 &= \int_t^{t+\Delta t} \mathbf{f}^m + \mathbf{b}_a + \mathbf{w}_a \, d\tau \\
 &= \mathbf{v}^m + \mathbf{b}_v + \mathbf{w}_v
 \end{aligned} \tag{4.13}$$

where \mathbf{v} denotes integrated specific force. The greek letter \mathbf{v} has been chosen to denote the integrated specific force, which should not be confused with the linear velocity \mathbf{v} . The integrated angular rate is denoted $\boldsymbol{\alpha}$, and the measurements are

$$\begin{aligned}
 \boldsymbol{\alpha}_I &= \int_t^{t+\Delta t} \boldsymbol{\omega}_I \, d\tau \\
 \boldsymbol{\alpha}_I &= \int_t^{t+\Delta t} \boldsymbol{\omega}_{im}^m + \mathbf{b}_g + \mathbf{w}_g \, d\tau \\
 &= \boldsymbol{\alpha}_{im}^m + \mathbf{b}_\alpha + \mathbf{w}_\alpha
 \end{aligned} \tag{4.14}$$

4.2.2 Sensor states

Since the output from the IMU is used as a reference navigation solution, the IMU state will contain the attitude, velocity and position solution. The bias is also included in the IMU state such that

$$\mathbf{x}_I = \begin{bmatrix} \mathbf{C}_m^i \\ \mathbf{v}_{im}^i \\ \mathbf{r}_{im}^i \\ \mathbf{b}_a \\ \mathbf{b}_g \end{bmatrix} \tag{4.15}$$

The IMU error state is given as the navigation errors and the bias errors:

$$\delta \mathbf{x}_I = \begin{bmatrix} \delta \boldsymbol{\psi}_{im}^m \\ \delta \mathbf{v}_{im}^i \\ \delta \mathbf{r}_{im}^i \\ \delta \mathbf{b}_a \\ \delta \mathbf{b}_g \end{bmatrix} \tag{4.16}$$

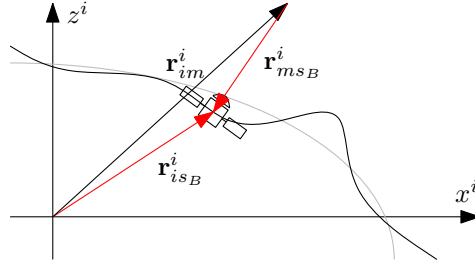
4.3 Barometer

The barometer provides measurements of the pressure at the current location:

$$p_m = p_0 \left[\frac{T_0}{T_0 + k_T (h_m - h_0)} \right]^{\frac{g_0}{Rk_T}} \tag{4.17}$$

Table 4.1: Barometer parameters. The left column assumes mean sea level.

T_0	288.15 K	k_T	-0.0065 K/m
p_0	101.325 kPa	R	287.05 J/(kgK)
h_0	0 m	g_0	9.80655 N/m


Figure 4.1: The barometer “satellite” geometry.

where T_0 , p_0 and h_0 are the temperature, pressure and altitude at a known reference point. h_m is the IMU height, referenced in the same system as the reference point. This reference point will be taken to be zero *geopotential* height, which means the zero level is the *geoid*. The constant k_T is the temperature *lapse rate*, which is determined using the altitude and the U.S. Standard atmosphere [4]. It is assumed to be constant, which is valid for heights of 11 km. This should be sufficient for the applications of the LocalHawk UAV. For altitudes above this, different values for k_T has to be used. R and g_0 are the ideal gas constant and the average surface acceleration due to gravity. The parameter values can be found in Table 4.1.

Since the pressure is a monotonically decreasing function of height, it is possible to invert Equation (4.17) to obtain the height as a function of pressure:

$$h_m = h_0 + \frac{T_0}{k_T} \left[\left(\frac{p_m}{p_0} \right)^{-\frac{Rk_T}{g_0}} - 1 \right] \quad (4.18)$$

The advantage of this transformation is that the barometer measurement can be written as a pseudorange measurement by introducing a satellite at zero geopotential height, as seen in Figure 4.1. The distance from the geopotential surface is given by

$$\begin{aligned} \rho_B &= \|\mathbf{r}_{msB}^i\| \\ &= \sqrt{(\mathbf{r}_{isB}^i - \mathbf{r}_{im}^i)^\top (\mathbf{r}_{isB}^i - \mathbf{r}_{im}^i)} \end{aligned} \quad (4.19)$$

$$\begin{aligned} &= h_m - h_0 \\ &= \frac{T_0}{k_T} \left[\left(\frac{p_m}{p_0} \right)^{-\frac{Rk_T}{g_0}} - 1 \right] \end{aligned} \quad (4.20)$$

4.3.1 Measurement equation

The measurement is the pseudorange ρ_B with an added bias b_B :

$$\begin{aligned} z_B &= \rho_B + b_B + w_B \\ &= \frac{T_0}{k_T} \left[\left(\frac{p_m}{p_0} \right)^{-\frac{Rk_T}{90}} - 1 \right] + b_B + w_B \\ &= h_B(\mathbf{x}) + w_B \end{aligned} \tag{4.21}$$

4.4 Magnetometer

By equipping the UAV with a magnetometer, it is possible to get measurements of the local magnetic field resolved in the magnetometer frame. With knowledge of the local magnetic field, the attitude of the UAV can be computed.

4.4.1 Measurement equation

The magnetometer triad outputs the magnetic field strength along the IMU axes, and the measurement is given by

$$\begin{aligned} \mathbf{z}_M &= \mathbf{m}^m + \mathbf{w}_M \\ &= \mathbf{C}_i^m \mathbf{m}^i + \mathbf{w}_M \end{aligned} \tag{4.22}$$

which gives

$$\mathbf{h}_M(\mathbf{x}) = \mathbf{C}_i^m \mathbf{m}^i \tag{4.23}$$

Take care to note the difference between the boldface \mathbf{m} , which is the magnetic field measurement, and m , which is the IMU frame.

The magnetic field \mathbf{m}^i is a function of the user position and time. Given information about the user position and current year, the model outputs the magnetic field component in the NED-frame.

4.4.2 Magnetometer errors

The main error sources on magnetometers are unmodeled variations in the magnetic field (local anomalies) and instrument errors [23]. The instrument errors are often split into two components, the hard- and soft-iron magnetism. Hard-iron magnetism is the addition of a magnetic field vector from magnets and electrical equipment, while soft-iron magnetism is errors in the local underlying field, such as magnetic field distortions from the material the magnetometer is mounted on.

These errors can often be determined through a process known as *swinging*, which can be done mechanically or electrically [29]. It will be assumed that this process has been completed, and that additional errors can be captured in the additive noise.

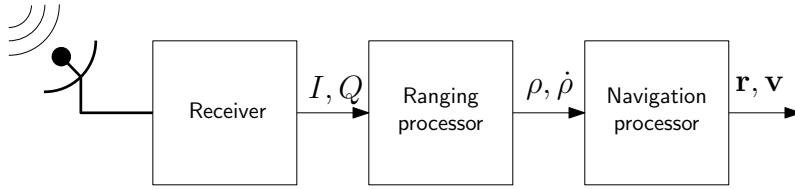


Figure 4.2: General GNSS hardware architecture.

4.5 Leveling

The gravity vector appears in the measurement equation for the accelerometer triad, which means it can be used to determine the attitude of the UAV. This process is known as leveling [23]. In order to do this, the acceleration of the UAV must be calculated from an external source or assumption.

4.5.1 Measurement equation

The corrected IMU output will be used as a measurement of the attitude:

$$\begin{aligned}
 \mathbf{z}_A &= \mathbf{v}_I - \hat{\mathbf{b}}_v \\
 &= \mathbf{v}^m + \mathbf{b}_v + \mathbf{w}_v - \hat{\mathbf{b}}_v \\
 &= \int_t^{t+\Delta t} \mathbf{f}^m(\tau) d\tau + \Delta t \mathbf{b}_a - \Delta t \mathbf{b}_a + \mathbf{w}_v \\
 &= \int_t^{t+\Delta t} \mathbf{C}_m^i \top (\mathbf{a}_{im}^i - \mathbf{g}^i) d\tau + \Delta t \delta \mathbf{b}_a + \mathbf{w}_v \\
 &\approx \Delta t \mathbf{C}_m^i \top (\mathbf{a}_{im}^i - \mathbf{g}^i) + \Delta t \delta \mathbf{b}_a + \mathbf{w}_v
 \end{aligned}$$

which gives

$$\mathbf{h}_A(\mathbf{x}) = \Delta t \mathbf{C}_m^i \top (\mathbf{a}_{im}^i - \mathbf{g}^i) + \Delta t \delta \mathbf{b}_a \quad (4.24)$$

Note that the subscript A is used for leveling, since the accelerometers are used.

4.6 Global navigation satellite system

The GNSS equipment tracks signals from satellites in view of the antenna. It is structured as shown in Figure 4.2 [23], and consists of the receiver, the ranging processor and the navigation processor. The operating principles of the receiver and ranging processor are only briefly described here. For a full treatment, see for example [23].

The receiver perform low-level signal processing, such as low-pass filtering and sampling. These signals, in-phase (I) and quadra-phase (Q) signals are sampled by the ranging processor and compared to predicted values from the ranging processor. The difference can be used to determine the signal transmit time. Multiplying the transmit time by the speed of light give the range from the antenna to the satellite. Similarly, it is possible to calculate the rate of change, known as the range-rate. In practice, inaccuracies in the clock receiver will perturb the measurements.

The predicted range due to transmit time and clock errors is denoted *pseudorange*, denoted ρ . The *pseudorange-rate* is denoted $\dot{\rho}$.

The last stage of conventional GNSS hardware is the navigation processor. Given pseudoranges and pseudorange-rates from at least 4 satellites, it is possible to resolve the antenna position, velocity and aforementioned clock errors. There are several methods to do this, some of which will be discussed in Section 5.6.3. The resulting position and velocity can then be used to correct the position and velocity estimates. This is called *loose* integration of GNSS, due to the separate navigation processor.

Another option is to skip the navigation processor and use the ranging processor output directly. This is called *tight* integration.

It should also be noted that there exists integration schemes where the I's and Q's are used as measurements in the integration filter. This, however, will not be pursued further.

4.6.1 Measurement equations

Loose integration

For loose integration, the navigation processor has calculated a standalone position and velocity estimate of the antenna, along with the receiver clock errors. The antenna position and velocity must be transposed using the techniques described in Appendix A to obtain the corresponding error states [23]:

$$\mathbf{r}_{ia}^i = \mathbf{r}_{im}^i + \mathbf{C}_m^i \mathbf{r}_{ma}^m \quad (4.25)$$

and

$$\mathbf{v}_{ia}^i = \mathbf{v}_{im}^i + \mathbf{C}_m^i [\boldsymbol{\omega}_{im}^m]^\times \mathbf{r}_{ma}^m \quad (4.26)$$

The receiver clock errors are also output, which give the total measurement as

$$\mathbf{z}_G = \begin{bmatrix} \mathbf{r}_{ia}^i \\ \mathbf{v}_{ia}^i \\ b_\rho \\ b_{\dot{\rho}} \end{bmatrix} + \mathbf{w}_G \quad (4.27)$$

where \mathbf{w}_G has covariance \mathbf{R}_G . This gives

$$\mathbf{h}_G(\mathbf{x}) = \begin{bmatrix} \mathbf{r}_{im}^i + \mathbf{C}_m^i \mathbf{r}_{ma}^m \\ \mathbf{v}_{im}^i + \mathbf{C}_m^i [\boldsymbol{\omega}_{im}^m]^\times \mathbf{r}_{ma}^m \\ b_\rho \\ b_{\dot{\rho}} \end{bmatrix} \quad (4.28)$$

The measurement noise \mathbf{w}_G is assumed to capture all errors in the GNSS system, ranging from the satellite motion to the resolution of position and velocity estimates from the pseudorange measurements.

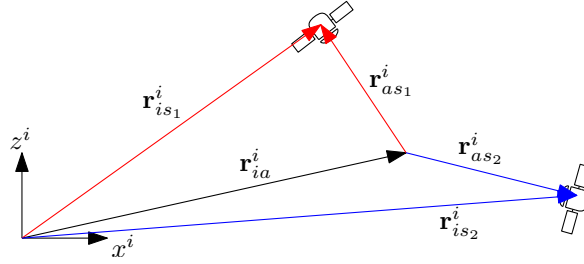


Figure 4.3: Geometry of satellite navigation, simplified to two dimensions.

Tight integration

For tight integration, consider Figure 4.3. The measurements for tightly measured systems consists of the pseudoranges and pseudorange-rates for each individual satellite to the receiver antenna. The pseudorange is given as

$$\begin{aligned}
 \rho_j &= \left\| \mathbf{r}_{as_j}^i \right\| + b_\rho \\
 &= \sqrt{\mathbf{r}_{as_j}^i \top \mathbf{r}_{as_j}^i} + b_\rho \\
 &= \sqrt{(\mathbf{r}_{is_j}^i - \mathbf{r}_{ia}^i) \top (\mathbf{r}_{is_j}^i - \mathbf{r}_{ia}^i)} + b_\rho
 \end{aligned} \tag{4.29}$$

for each satellite s_j . The pseudorange-rate is given by the differentiation of the pseudorange:

$$\begin{aligned}
 \dot{\rho}_j &= \frac{d}{dt} \left(\sqrt{\mathbf{r}_{as_j}^i \top \mathbf{r}_{as_j}^i} + b_\rho \right) \\
 &= \frac{2\mathbf{r}_{as_j}^i \top}{2\sqrt{\mathbf{r}_{as_j}^i \top \mathbf{r}_{as_j}^i}} \frac{d}{dt} \left(\mathbf{r}_{is_j}^i - \mathbf{r}_{ia}^i \right) + \dot{b}_\rho \\
 &= \mathbf{l}_{as_j}^i \top \left(\mathbf{v}_{is_j}^i - \mathbf{v}_{ia}^i \right) + \dot{b}_\rho
 \end{aligned} \tag{4.30}$$

where $\mathbf{l}_{as_j}^i$ is the unit vector pointing from the antenna to satellite j , also called the Line of sight (LOS) vector. The pseudoranges and pseudorange-rates can be grouped together as a single measurement:

$$\mathbf{z}_G = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_{N_s} \\ \dot{\rho}_1 \\ \vdots \\ \dot{\rho}_{N_s} \end{bmatrix} + \mathbf{w}_G \tag{4.31}$$

where N_s is the total number of satellites in view. This will vary by the location of the antenna and obstructions close to the antenna, but is obviously bounded by the

Table 4.2: Error and total state overview.

State	True value	Estimate	Error	Composition
Attitude (DCM)	\mathbf{C}_m^i	$\hat{\mathbf{C}}_m^i$	$\delta\mathbf{C}_i^m$	$\mathbf{C}_m^i = \hat{\mathbf{C}}_m^i \delta\mathbf{C}_i^m$
Attitude (Angle)			$\delta\boldsymbol{\psi}_{im}^m$	$\delta\mathbf{C}_i^m \approx \mathbf{I}_3 - [\delta\boldsymbol{\psi}_{im}^m]^\times$
Position	\mathbf{r}_{im}^i	$\hat{\mathbf{r}}_{im}^i$	$\delta\mathbf{r}_{im}^i$	$\mathbf{r}_{im}^i = \hat{\mathbf{r}}_{im}^i + \delta\mathbf{r}_{im}^i$
Velocity	\mathbf{v}_{im}^i	$\hat{\mathbf{v}}_{im}^i$	$\delta\mathbf{v}_{im}^i$	$\mathbf{v}_{im}^i = \hat{\mathbf{v}}_{im}^i + \delta\mathbf{v}_{im}^i$
Accelerometer Bias	\mathbf{b}_a	$\hat{\mathbf{b}}_a$	$\delta\mathbf{b}_a$	$\mathbf{b}_a = \hat{\mathbf{b}}_a + \delta\mathbf{b}_a$
Gyro Bias	\mathbf{b}_g	$\hat{\mathbf{b}}_g$	$\delta\mathbf{b}_g$	$\mathbf{b}_g = \hat{\mathbf{b}}_g + \delta\mathbf{b}_g$
GNSS Clock Bias	b_ρ	\hat{b}_ρ	δb_ρ	$b_\rho = \hat{b}_\rho + \delta b_\rho$
GNSS Clock Drift	$b_{\dot{\rho}}$	$\hat{b}_{\dot{\rho}}$	$\delta b_{\dot{\rho}}$	$b_{\dot{\rho}} = \hat{b}_{\dot{\rho}} + \delta b_{\dot{\rho}}$
Barometer Bias	b_B	\hat{b}_B	δb_B	$b_B = \hat{b}_B + \delta b_B$

total number of satellites in view. The measurement equation can then be written

$$\mathbf{h}_G(\mathbf{x}) = \begin{bmatrix} \sqrt{(\mathbf{r}_{is_1}^i - \mathbf{r}_{ia}^i)^\top (\mathbf{r}_{is_1}^i - \mathbf{r}_{ia}^i)} + b_\rho \\ \vdots \\ \sqrt{(\mathbf{r}_{is_{N_s}}^i - \mathbf{r}_{ia}^i)^\top (\mathbf{r}_{is_{N_s}}^i - \mathbf{r}_{ia}^i)} + b_\rho \\ \mathbf{l}_{as_1}^i \top (\mathbf{v}_{is_1}^i - \mathbf{v}_{ia}^i) + b_{\dot{\rho}} \\ \vdots \\ \mathbf{l}_{as_{N_s}}^i \top (\mathbf{v}_{is_{N_s}}^i - \mathbf{v}_{ia}^i) + b_{\dot{\rho}} \end{bmatrix} \quad (4.32)$$

4.6.2 Sensor states

The modeled errors consist of the clock bias and the clock drift. As with the accelerometer values, they are split into a known and unknown component:

$$b_\rho = \hat{b}_\rho + \delta b_\rho \quad (4.33)$$

$$b_{\dot{\rho}} = \hat{b}_{\dot{\rho}} + \delta b_{\dot{\rho}} \quad (4.34)$$

which give the GNSS total state and the GNSS error state as

$$\mathbf{x}_G = \begin{bmatrix} b_\rho \\ b_{\dot{\rho}} \end{bmatrix} \quad \delta\mathbf{x}_G = \begin{bmatrix} \delta b_\rho \\ \delta b_{\dot{\rho}} \end{bmatrix} \quad (4.35)$$

4.7 Overview

The states and sensor subscripts are summarized in Table 4.2 and Table 4.3, respectively.

Table 4.3: Sensor subscripts

IMU	<i>I</i>
Barometer	<i>B</i>
Magnetometer	<i>M</i>
Leveling	<i>A</i>
GNSS	<i>G</i>

Filter Implementation

5.1 Filter structure

This implementation make use of an error state extended Kalman filter. The measurement equations from the previous chapter will be linearized, and the error state transition model will be derived.

The error state Kalman filter has been chosen primarily for its modularity, as well as the advantages detailed in Section 2.1.2.

5.1.1 State transition model

A linear error process model will be used for the error state:

$$\frac{d}{dt}\delta\mathbf{x} = \mathbf{F}\delta\mathbf{x} \quad (5.1)$$

Since there are no interactions between the sensors in the system model [23], the state transition matrix can be written

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_I & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_B & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{F}_G \end{bmatrix} \quad (5.2)$$

where the $\mathbf{0}$ are zero matrices of appropriate size. The process model is discretized before it is input to the EKF. Due to the slow dynamics and fast update rate, a forward euler method is used to obtain the state transition model:

$$\delta\mathbf{x}_{k+1} = (\mathbf{I} + \mathbf{F})\delta\mathbf{x}_k = \mathbf{F}_k\delta\mathbf{x}_k \quad (5.3)$$

The discretization and linearization errors, as well as any unmodeled effects, are assumed to be captured by the state transition noise. This give the state transition model

$$\delta\mathbf{x}_{k+1} = \mathbf{F}_k\delta\mathbf{x}_k + \mathbf{u}_k \quad \mathbf{u}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \quad (5.4)$$

The state transition noise covariance \mathbf{Q}_k is given by

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{Q}_{I,k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{B,k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{G,k} \end{bmatrix} \quad (5.5)$$

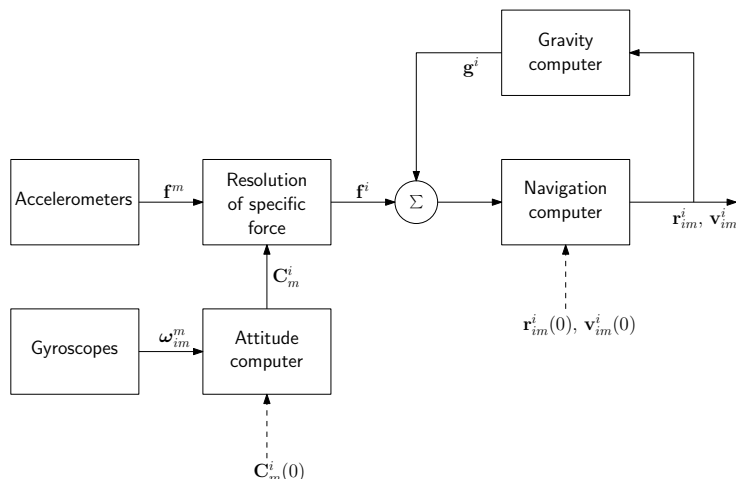


Figure 5.1: A continuous-time strapdown system.

Further details of the state transition model are provided in the following sections.

5.1.2 Measurement Jacobians

Although the process model is assumed to be linear, some of the measurements are nonlinear. This includes the attitude measurements and the tightly coupled system. The measurement Jacobian can be written

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_B \\ \mathbf{H}_M \\ \mathbf{H}_A \\ \mathbf{H}_G \end{bmatrix} \quad (5.6)$$

Further details of the individual Jacobians are also provided in the following sections.

5.2 Inertial measurement unit

5.2.1 Strapdown algorithm

The strapdown algorithm is the reference navigation solution, which integrates the output from the IMU into attitude, linear velocity and position. They are related by

$$\dot{\mathbf{C}}_m^i = \mathbf{C}_m^i [\boldsymbol{\omega}_{im}^m]^\times \quad (5.7)$$

$$\dot{\mathbf{v}}_{im}^i = \mathbf{C}_m^i \mathbf{f}^m - \mathbf{g}^i \quad (5.8)$$

$$\dot{\mathbf{r}}_{im}^i = \mathbf{v}_{im}^i \quad (5.9)$$

where \mathbf{g}^i is the gravity vector at the current position resolved in the inertial frame, and $[\cdot]^\times$ denotes the skew-symmetric form of a three-element vector. A block diagram is shown in Figure 5.1 Discretization of the strapdown has previously been discussed

at length [47], and this section summarizes the results. The attitude update is given by

$$\mathbf{C}_m^i(t + \Delta t) = \mathbf{C}_m^i(t) \exp([\boldsymbol{\alpha}_{im}^m]^\times) \quad (5.10)$$

where $\exp(\cdot)$ denotes the matrix exponential, and the implementation is discussed in Section 3.3.2.

The velocity update is given by

$$\mathbf{v}_{im}^i(t + \Delta t) = \mathbf{v}_{im}^i(t) + \mathbf{v}^i(t + \Delta t) + \Delta t \mathbf{g}^i(\mathbf{r}_{im}^i(t)) \quad (5.11)$$

where the gravitational acceleration is assumed to be constant over the time interval, and the rotated integrated specific force is given by

$$\mathbf{v}^i(t + \Delta t) = \mathbf{C}_m^i(t) \left[\mathbf{v}^m(t + \Delta t) + \frac{1}{2} \boldsymbol{\alpha}_{im}^m(t + \Delta t) \times \mathbf{v}^m(t + \Delta t) \right] \quad (5.12)$$

The gravitational term is modeled by EGM2008, and the details will be given in Section 6.1.1

The position calculation is a trapezoidal method with the velocity at the current and previous time step:

$$\mathbf{r}_{im}^i(t + \Delta t) = \mathbf{r}_{im}^i(t) + \Delta t \frac{\mathbf{v}_{im}^i(t) + \mathbf{v}_{im}^i(t + \Delta t)}{2} \quad (5.13)$$

These calculations form the basis for the navigation system, and the other sensors provide corrections to the strapdown solution and estimates of the IMU errors.

5.2.2 State transition matrix

Since the IMU is used as the reference navigation sensor, the navigation error state and the IMU errors will be coupled.

Attitude error dynamics

The attitude error angle dynamics can be found by considering the skew-symmetric component:

$$\begin{aligned} \frac{d[\delta\boldsymbol{\psi}_{im}^m]^\times}{dt} &\approx -\frac{d\delta\mathbf{C}_m^i}{dt} \\ &= -\frac{d\hat{\mathbf{C}}_m^i{}^\top \mathbf{C}_m^i}{dt} \\ &= -\frac{d\hat{\mathbf{C}}_m^i{}^\top}{dt} \mathbf{C}_m^i - \hat{\mathbf{C}}_m^i{}^\top \frac{d\mathbf{C}_m^i}{dt} \\ &= \left(\hat{\mathbf{C}}_m^i [\hat{\boldsymbol{\omega}}_{im}^m]^\times \right)^\top \mathbf{C}_m^i - \hat{\mathbf{C}}_m^i{}^\top \mathbf{C}_m^i [\boldsymbol{\omega}_{im}^m]^\times \\ &= -[\hat{\boldsymbol{\omega}}_{im}^m]^\times{}^\top \hat{\mathbf{C}}_m^i{}^\top \mathbf{C}_m^i - \hat{\mathbf{C}}_m^i{}^\top \mathbf{C}_m^i [\boldsymbol{\omega}_{im}^m]^\times \\ &= [\hat{\boldsymbol{\omega}}_{im}^m]^\times \delta\mathbf{C}_m^i - \delta\mathbf{C}_m^i [\boldsymbol{\omega}_{im}^m]^\times \end{aligned} \quad (5.14)$$

Note that the error DCM is reset at every filter iteration. Thus it is approximated by the identity matrix:

$$\begin{aligned}\frac{d[\delta\boldsymbol{\psi}_{im}^m]^\times}{dt} &\approx [\hat{\boldsymbol{\omega}}_{im}^m]^\times - [\boldsymbol{\omega}_{im}^m]^\times \\ &= [\hat{\boldsymbol{\omega}}_{im}^m - \boldsymbol{\omega}_{im}^m]^\times\end{aligned}$$

From the IMU output, Equation (4.8), the attitude error estimate is dependent on the gyro bias error:

$$\begin{aligned}\delta\dot{\boldsymbol{\psi}}_{im}^m &\approx \boldsymbol{\omega}_I - \hat{\mathbf{b}}_g - (\boldsymbol{\omega}_I - \mathbf{b}_g) \\ &= \mathbf{b}_g - \hat{\mathbf{b}}_g \\ &= \delta\mathbf{b}_g\end{aligned}\tag{5.15}$$

Velocity error dynamics

Consider the velocity error

$$\begin{aligned}\frac{d\delta\mathbf{v}_{im}^i}{dt} &= \frac{d}{dt} (\mathbf{v}_{im}^i - \hat{\mathbf{v}}_{im}^i) \\ &= \mathbf{C}_m^i \mathbf{f}^m - \mathbf{g}^i - (\hat{\mathbf{C}}_m^i \hat{\mathbf{f}}^m - \hat{\mathbf{g}}^i) \\ &= \hat{\mathbf{C}}_m^i \delta\mathbf{C}_i^m (\hat{\mathbf{f}}^m + \hat{\mathbf{b}}_a - \mathbf{b}_a) - \hat{\mathbf{C}}_m^i \hat{\mathbf{f}}^m \\ &= \hat{\mathbf{C}}_m^i (\delta\mathbf{C}_i^m (\hat{\mathbf{f}}^m - \delta\mathbf{b}_a) - \hat{\mathbf{f}}^m)\end{aligned}\tag{5.16}$$

where the various error definitions from Table 4.2 have been used to rewrite the velocity error dynamics as functions of the other errors. The small-angle approximation is applied to Equation (5.16) to obtain

$$\begin{aligned}\delta\dot{\mathbf{v}}_{im}^i &\approx \hat{\mathbf{C}}_m^i \left((\mathbf{I}_3 - [\delta\boldsymbol{\psi}_{im}^m]^\times) (\hat{\mathbf{f}}^m - \delta\mathbf{b}_a) - \hat{\mathbf{f}}^m \right) \\ &= \hat{\mathbf{C}}_m^i \left(\hat{\mathbf{f}}^m - [\delta\boldsymbol{\psi}_{im}^m]^\times \hat{\mathbf{f}}^m - \delta\mathbf{b}_a + [\delta\boldsymbol{\psi}_{im}^m]^\times \delta\mathbf{b}_a - \hat{\mathbf{f}}^m \right) \\ &= \hat{\mathbf{C}}_m^i \left([\hat{\mathbf{f}}^m]^\times \delta\boldsymbol{\psi}_{im}^m - \delta\mathbf{b}_a + [\delta\boldsymbol{\psi}_{im}^m]^\times \delta\mathbf{b}_a \right)\end{aligned}\tag{5.17}$$

The last term in Equation (5.17) is a product of two error states. To obtain a linear state transition model, this term must be assumed to be negligible, such that

$$\delta\dot{\mathbf{v}}_{im}^i \approx \hat{\mathbf{C}}_m^i [\hat{\mathbf{f}}^m]^\times \delta\boldsymbol{\psi}_{im}^m - \mathbf{C}_m^i \delta\mathbf{b}_a\tag{5.18}$$

Position error dynamics

The position error is simply an integral of the velocity error:

$$\delta\dot{\mathbf{r}}_{im}^i = \delta\mathbf{v}_{im}^i\tag{5.19}$$

Bias error dynamics

For the accelerometer and bias, we use a first order Markov model:

$$\dot{\mathbf{b}}_a = \mathbf{T}_a^{-1} \mathbf{b}_a + \mathbf{w}_{b_a} \quad (5.20)$$

$$\dot{\mathbf{b}}_g = \mathbf{T}_g^{-1} \mathbf{b}_g + \mathbf{w}_{b_g} \quad (5.21)$$

where \mathbf{T}_a and \mathbf{T}_g are diagonal time constant matrices.

IMU state transition matrix

The error state transition for the IMU can be formed from the previous states:

$$\delta \dot{\mathbf{x}}_I = \frac{d}{dt} \begin{bmatrix} \delta \psi_{im}^m \\ \delta \mathbf{v}_{im}^i \\ \delta \mathbf{r}_{im}^i \\ \mathbf{b}_a \\ \mathbf{b}_g \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \\ \hat{\mathbf{C}}_m^i \left[\hat{\mathbf{f}}^m \right]^\times & \mathbf{0}_3 & \mathbf{0}_3 & -\hat{\mathbf{C}}_m^i & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{T}_a^{-1} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{T}_g^{-1} \end{bmatrix} \begin{bmatrix} \delta \psi_{im}^m \\ \delta \mathbf{v}_{im}^i \\ \delta \mathbf{r}_{im}^i \\ \mathbf{b}_a \\ \mathbf{b}_g \end{bmatrix} = \mathbf{F}_I \delta \mathbf{x}_I \quad (5.22)$$

Equation (5.22) is the first element on the diagonal of Equation (5.2).

5.3 Barometer

5.3.1 Jacobian

The barometer measurement depends on the position:

$$\begin{aligned} \frac{\partial h_B}{\partial \delta \mathbf{r}_{im}^i} &= \frac{\partial}{\partial \delta \mathbf{r}_{im}^i} \sqrt{(\mathbf{r}_{is_B}^i - \mathbf{r}_{im}^i)^\top (\mathbf{r}_{is_B}^i - \mathbf{r}_{im}^i)} \\ &= \frac{2 (\mathbf{r}_{is_B}^i - \mathbf{r}_{im}^i)^\top}{2 \sqrt{(\mathbf{r}_{is_B}^i - \mathbf{r}_{im}^i)^\top (\mathbf{r}_{is_B}^i - \mathbf{r}_{im}^i)}} \frac{\partial (\mathbf{r}_{is_B}^i - \hat{\mathbf{r}}_{im}^i - \delta \mathbf{r}_{im}^i)}{\partial \delta \mathbf{r}_{im}^i} \\ &= \frac{(\mathbf{r}_{is_B}^i - \mathbf{r}_{im}^i)^\top}{\sqrt{(\mathbf{r}_{is_B}^i - \mathbf{r}_{im}^i)^\top (\mathbf{r}_{is_B}^i - \mathbf{r}_{im}^i)}} (-\mathbf{I}_3) \\ &= \mathbf{I}_{ms_B}^{i\top} \end{aligned} \quad (5.23)$$

The Jacobian corresponding to the bias element is simply

$$\frac{\partial h_B}{\partial b_B} = 1 \quad (5.24)$$

which give the total barometer Jacobian as

$$\begin{aligned} \mathbf{H}_B &= \begin{bmatrix} \frac{\partial h_B}{\partial \delta \psi_{im}^m} & \frac{\partial h_B}{\partial \delta \mathbf{v}_{im}^i} & \frac{\partial h_B}{\partial \delta \mathbf{r}_{im}^i} & \frac{\partial h_B}{\partial \delta \mathbf{b}_a} & \frac{\partial h_B}{\partial \delta \mathbf{b}_g} & \frac{\partial h_B}{\partial \delta \mathbf{x}_B} & \frac{\partial h_B}{\partial \delta \mathbf{x}_G} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{I}_{ms_B}^{i\top} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 1 & \mathbf{0}_{1 \times 2} \end{bmatrix} \end{aligned} \quad (5.25)$$

5.3.2 State transition model

Similarly to the IMU biases, the barometer bias is modeled as a first-order Markov model:

$$\dot{b}_B = -T_B^{-1}b_B \quad (5.26)$$

This give the matrix \mathbf{F}_B in Equation (5.2) as

$$\mathbf{F}_B = -T_B^{-1} \quad (5.27)$$

5.4 Magnetometer

5.4.1 Jacobian

The magnetometer measurement in Equation (4.22) can be rewritten in terms of the estimate and error DCM by the definition in Equation (4.2):

$$\begin{aligned} \mathbf{h}_M &= \mathbf{C}_i^m \mathbf{m}^i \\ &= \mathbf{C}_m^i \top \mathbf{m}^i \\ &= \delta \mathbf{C}_i^m \top \hat{\mathbf{C}}_i^m \mathbf{m}^i \end{aligned} \quad (5.28)$$

The small-angle approximation is applied to write the measurement as a function of the error state:

$$\begin{aligned} \mathbf{h}_M &\approx (\mathbf{I}_3 - [\delta \boldsymbol{\psi}_{im}^m]^\times)^\top \hat{\mathbf{C}}_m^i \top \mathbf{m}^i \\ &= \hat{\mathbf{C}}_m^i \top \mathbf{m}^i + [\delta \boldsymbol{\psi}_{im}^m]^\times \hat{\mathbf{C}}_m^i \top \mathbf{m}^i \\ &= \hat{\mathbf{C}}_m^i \top \mathbf{m}^i - [\hat{\mathbf{C}}_m^i \top \mathbf{m}^i]^\times \delta \boldsymbol{\psi}_{im}^m \end{aligned} \quad (5.29)$$

where the anticommutative property of the cross product has been used. This give the nonzero Jacobian elements as

$$\frac{\partial \mathbf{h}_M}{\partial \delta \boldsymbol{\psi}_{im}^m} = - [\hat{\mathbf{C}}_m^i \top \mathbf{m}^i]^\times \quad (5.30)$$

and the magnetometer Jacobian as

$$\begin{aligned} \mathbf{H}_M &= \begin{bmatrix} \frac{\partial \mathbf{h}_M}{\partial \delta \boldsymbol{\psi}_{im}^m} & \frac{\partial \mathbf{h}_M}{\partial \delta \mathbf{v}_{im}^i} & \frac{\partial \mathbf{h}_M}{\partial \delta \mathbf{r}_{im}^i} & \frac{\partial \mathbf{h}_M}{\partial \delta \mathbf{b}_a} & \frac{\partial \mathbf{h}_M}{\partial \delta \mathbf{b}_g} & \frac{\partial \mathbf{h}_M}{\partial \delta \mathbf{x}_B} & \frac{\partial \mathbf{h}_M}{\partial \delta \mathbf{x}_G} \end{bmatrix} \\ &= \begin{bmatrix} - [\hat{\mathbf{C}}_m^i \top \mathbf{m}^i]^\times & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 2} \end{bmatrix} \end{aligned} \quad (5.31)$$

5.5 Leveling

5.5.1 Jacobian

The leveling measurement Jacobian is derived similarly to the magnetometer. From Equation (4.24):

$$\begin{aligned} \mathbf{h}_A(\mathbf{x}) &= \Delta t \left(\left(\hat{\mathbf{C}}_m^i \delta \mathbf{C}_i^m \right)^\top (\mathbf{a}_{im}^i - \mathbf{g}^i) + \delta \mathbf{b}_a \right) \\ &\approx \Delta t \left((\mathbf{I}_3 - [\delta \boldsymbol{\psi}_{im}^m]^\times)^\top \hat{\mathbf{C}}_m^i \top (\mathbf{a}_{im}^i - \mathbf{g}^i) + \delta \mathbf{b}_a \right) \\ &= \Delta t \left(\hat{\mathbf{C}}_m^i \top (\mathbf{a}_{im}^i - \mathbf{g}^i) - \left[\hat{\mathbf{C}}_m^i \top (\mathbf{a}_{im}^i - \mathbf{g}^i) \right]^\times \delta \boldsymbol{\psi}_{im}^m + \delta \mathbf{b}_a \right) \end{aligned} \quad (5.32)$$

which give the nonzero elements of the Jacobian \mathbf{H}_A as

$$\frac{\partial \mathbf{h}_A}{\partial \delta \boldsymbol{\psi}_{im}^m} = -\Delta t \left[\hat{\mathbf{C}}_m^i \top (\mathbf{a}_{im}^i - \mathbf{g}^i) \right]^\times \quad (5.33)$$

$$\frac{\partial \mathbf{h}_A}{\partial \delta \mathbf{b}_a} = \Delta t \mathbf{I}_3 \quad (5.34)$$

and the leveling Jacobian as

$$\begin{aligned} \mathbf{H}_A &= \begin{bmatrix} \frac{\partial \mathbf{h}_A}{\partial \delta \boldsymbol{\psi}_{im}^m} & \frac{\partial \mathbf{h}_A}{\partial \delta \mathbf{v}_{im}^i} & \frac{\partial \mathbf{h}_A}{\partial \delta \mathbf{r}_{im}^i} & \frac{\partial \mathbf{h}_A}{\partial \delta \mathbf{b}_a} & \frac{\partial \mathbf{h}_A}{\partial \delta \mathbf{b}_g} & \frac{\partial \mathbf{h}_A}{\partial \delta \mathbf{x}_B} & \frac{\partial \mathbf{h}_A}{\partial \delta \mathbf{x}_G} \end{bmatrix} \\ &= \begin{bmatrix} -\Delta t \left[\hat{\mathbf{C}}_m^i \top (\mathbf{a}_{im}^i - \mathbf{g}^i) \right]^\times & \mathbf{0}_3 & \mathbf{0}_3 & \Delta t \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 2} \end{bmatrix} \end{aligned} \quad (5.35)$$

5.5.2 Acceleration calculation

Equation (5.33) shows that the acceleration \mathbf{a}_{im}^i is needed to compute the attitude. A common method is to assume the UAV is stationary or moving with constant velocity with respect to the local frame. This means the only acceleration measured by the accelerometers are the coriolis and centripetal acceleration described in Appendix A, and the acceleration is given by

$$\mathbf{a}_{im}^i \approx 2 [\boldsymbol{\omega}_{ie}^i]^\times \mathbf{v}_{im}^i - [\boldsymbol{\omega}_{ie}^i]^\times [\boldsymbol{\omega}_{ie}^i]^\times \mathbf{r}_{im}^i \quad (5.36)$$

With this approach, the measurement will be corrupted during maneuvers, with accelerations of 1m/s^2 corresponding to an error of approximately 5.7° [23]. To avoid this, it is possible to add a maneuver detection algorithm. The most simple method is to check the norm of the IMU output:

$$g^m = \frac{\|\hat{\mathbf{v}}^m\|}{\Delta t} \quad (5.37)$$

and use the attitude measurement if g^m is in an interval such that

$$g^m \in [g_{\min}, g_{\max}] \quad (5.38)$$

and discard it otherwise. This should be used with care, as the accelerometer bias errors will influence the value of g^m . The limits should be chosen conservatively, and it should be possible to deactivate this maneuver detection. For example, the it should be deactivated during initialization.

5.6 GNSS

5.6.1 Jacobian

The GNSS Jacobian depends on the integration structure.

Loosely coupled integration

First, rewrite the position measurement from Equation (4.28) in terms of the error states:

$$\begin{aligned} \mathbf{r}_{im}^i + \mathbf{C}_m^i \mathbf{r}_{ma}^m &= \mathbf{r}_{im}^i + \hat{\mathbf{C}}_m^i + \delta \mathbf{C}_m^i \mathbf{r}_{ma}^m \\ &\approx \mathbf{r}_{im}^i + \hat{\mathbf{C}}_m^i (\mathbf{I}_3 - [\delta \boldsymbol{\psi}_{im}^m]^\times) \mathbf{r}_{ma}^m \\ &= \hat{\mathbf{r}}_{im}^i + \delta \mathbf{r}_{im}^i + \hat{\mathbf{C}}_m^i \mathbf{r}_{ma}^m + \hat{\mathbf{C}}_m^i [\mathbf{r}_{ma}^m]^\times \delta \boldsymbol{\psi}_{im}^m \end{aligned} \quad (5.39)$$

which give the nonzero elements of the first three rows of the Jacobian as

$$\frac{\partial \mathbf{r}_{ia}^i}{\partial \delta \boldsymbol{\psi}_{im}^m} = \hat{\mathbf{C}}_m^i [\mathbf{r}_{ma}^m]^\times \quad (5.40)$$

$$\frac{\partial \mathbf{r}_{ia}^i}{\partial \delta \mathbf{r}_{im}^i} = \mathbf{I}_3 \quad (5.41)$$

Similarly, the velocity can be rewritten in terms of the error states as

$$\begin{aligned} \mathbf{v}_{im}^i + \mathbf{C}_m^i [\boldsymbol{\omega}_{im}^m]^\times \mathbf{r}_{ma}^m &\approx \mathbf{v}_{im}^i + \hat{\mathbf{C}}_m^i (\mathbf{I}_3 - [\delta \boldsymbol{\psi}_{im}^m]^\times) [\hat{\boldsymbol{\omega}}_{im}^m - \delta \mathbf{b}_g]^\times \mathbf{r}_{ma}^m \\ &= \hat{\mathbf{v}}_{im}^i + \delta \mathbf{v}_{im}^i + \hat{\mathbf{C}}_m^i [\hat{\boldsymbol{\omega}}_{im}^m]^\times \mathbf{r}_{ma}^m + \hat{\mathbf{C}}_m^i [\mathbf{r}_{ma}^m]^\times \delta \mathbf{b}_g + \dots \\ &\quad [\hat{\boldsymbol{\omega}}_{im}^m \times \mathbf{r}_{ma}^m]^\times \delta \boldsymbol{\psi}_{im}^m + [\delta \boldsymbol{\psi}_{im}^m]^\times (\delta \mathbf{b}_g \times \mathbf{r}_{ma}^m) \end{aligned} \quad (5.42)$$

This give the nonzero elements of the fourth through sixth rows of the Jacobian as

$$\begin{aligned} \frac{\partial \mathbf{v}_{ia}^i}{\partial \delta \boldsymbol{\psi}_{im}^m} &= \hat{\mathbf{C}}_m^i ([\hat{\boldsymbol{\omega}}_{im}^m \times \mathbf{r}_{ma}^m]^\times - [\delta \mathbf{b}_g \times \mathbf{r}_{ma}^m]^\times) \\ &= \hat{\mathbf{C}}_m^i [\hat{\boldsymbol{\omega}}_{im}^m \times \mathbf{r}_{ma}^m]^\times \end{aligned} \quad (5.43)$$

$$\frac{\partial \mathbf{v}_{ia}^i}{\partial \delta \mathbf{v}_{im}^i} = \mathbf{I}_3 \quad (5.44)$$

$$\begin{aligned} \frac{\partial \mathbf{v}_{ia}^i}{\partial \delta \mathbf{b}_g} &= \hat{\mathbf{C}}_m^i ([\mathbf{r}_{ma}^m]^\times - [\delta \boldsymbol{\psi}_{im}^m]^\times [\mathbf{r}_{ma}^m]^\times) \\ &= \hat{\mathbf{C}}_m^i [\mathbf{r}_{ma}^m]^\times \end{aligned} \quad (5.45)$$

where it has been exploited that the Jacobian is evaluated at the prior error state, which is zero due to filter reset. The two last rows of Equation (4.28) are trivial to rewrite, and the Jacobian can be calculated as

$$\frac{\partial b_\rho}{\partial \delta b_\rho} = 1 \quad (5.46)$$

$$\frac{\partial b_{\dot{\rho}}}{\partial \delta b_{\dot{\rho}}} = 1 \quad (5.47)$$

All the Jacobian elements related to attitude are dependant on the lever arm between the IMU and GNSS receiver antenna. If this value is small or the uncertainty is not negligible, it could be advantageous to set this to zero [23]. This give the loosely coupled measurement Jacobian as

$$\begin{aligned} \mathbf{H}_G &= \begin{bmatrix} \frac{\partial \mathbf{h}_G}{\partial \delta \psi_{im}^m} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{v}_{im}^i} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{r}_{im}^i} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{b}_a} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{b}_g} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{x}_B} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{x}_G} \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{C}}_m^i [\mathbf{r}_{ma}^m]^\times & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 2} \\ \hat{\mathbf{C}}_m^i [\hat{\boldsymbol{\omega}}_{im}^m \times \mathbf{r}_{im}^m]^\times & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{C}_m^i [\mathbf{r}_{ma}^m]^\times & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 1} & \mathbf{I}_2 \end{bmatrix} \end{aligned} \quad (5.48)$$

where the last two rows have been

Tightly coupled integration

Just as in the loosely coupled case, Equation (4.32) can be written in terms of the error states. To avoid double work, note that the satellite position does not contain any error states, and observe that

$$\frac{\partial \rho_j}{\partial \delta \mathbf{x}} = \frac{\partial \rho_j}{\partial \mathbf{r}_{ia}^i} \frac{\partial \mathbf{r}_{ia}^i}{\partial \delta \mathbf{x}} \quad (5.49)$$

where the second calculation has been carried out in the loosely coupled case. The first term is found as

$$\begin{aligned} \frac{\partial \rho_j}{\partial \mathbf{r}_{ia}^i} &= \frac{2 (\mathbf{r}_{is_j} - \mathbf{r}_{ia}^i)^\top}{2 \sqrt{(\mathbf{r}_{is_j} - \mathbf{r}_{ia}^i)^\top (\mathbf{r}_{is_j} - \mathbf{r}_{ia}^i)}} (-\mathbf{I}_3) \\ &= -\mathbf{l}_{as_j}^i{}^\top \end{aligned} \quad (5.50)$$

such that

$$\frac{\partial \rho_j}{\partial \delta \psi_{im}^m} = -\mathbf{l}_{as_j}^i{}^\top \hat{\mathbf{C}}_m^i [\mathbf{r}_{ma}^m]^\times \quad (5.51)$$

$$\frac{\partial \rho_j}{\partial \delta \mathbf{r}_{im}^i} = -\mathbf{l}_{as_j}^i{}^\top \quad (5.52)$$

and the clock bias is given by

$$\frac{\partial \rho_j}{\partial \delta b_\rho} = 1 \quad (5.53)$$

For the pseudorange-rate, note that the chain rule gives

$$\begin{aligned} \frac{\partial \dot{\rho}_j}{\partial \delta \mathbf{x}} &= (\mathbf{v}_{is_j}^i - \mathbf{v}_{ia}^i)^\top \frac{\partial \mathbf{l}_{as_j}^i}{\partial \delta \mathbf{x}} + \mathbf{l}_{as_j}^i{}^\top \frac{\partial}{\partial \delta \mathbf{x}} (\mathbf{v}_{is_j}^i - \mathbf{v}_{ia}^i) \\ &= (\mathbf{v}_{is_j}^i - \mathbf{v}_{ia}^i)^\top \frac{\partial \mathbf{l}_{as_j}^i}{\partial \delta \mathbf{x}} - \mathbf{l}_{as_j}^i{}^\top \frac{\partial \mathbf{v}_{ia}^i}{\partial \delta \mathbf{x}} \end{aligned} \quad (5.54)$$

The calculations in the second term has already been done. Since the LOS vector is a function of the antenna position, the Jacobian depends on the same terms as the

loosely coupled measurement. However, these terms are often ignored due to their low impact on the integrated navigation system [23]. This gives

$$\frac{\partial \dot{\rho}_j}{\partial \delta \psi_{im}^m} = -\mathbf{l}_{as_j}^i \top \hat{\mathbf{C}}_m^i [\hat{\boldsymbol{\omega}}_{im}^m \times \mathbf{r}_{ma}^m]^\times \quad (5.55)$$

$$\frac{\partial \dot{\rho}_j}{\partial \mathbf{v}_{im}^i} = -\mathbf{l}_{as_j}^i \top \quad (5.56)$$

$$\frac{\partial \dot{\rho}_j}{\partial \delta \mathbf{b}_g} = -\mathbf{l}_{as_j}^i \top \hat{\mathbf{C}}_m^i [\mathbf{r}_{ma}^m]^\times \quad (5.57)$$

and

$$\frac{\partial \dot{\rho}_j}{\partial \delta \dot{\rho}} = 1 \quad (5.58)$$

To summarize, the tightly coupled GNSS Jacobian is given by

$$\begin{aligned} \mathbf{H}_G &= \begin{bmatrix} \frac{\partial \mathbf{h}_G}{\partial \delta \psi_{im}^m} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{v}_{im}^i} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{r}_{im}^i} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{b}_a} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{b}_g} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{x}_B} & \frac{\partial \mathbf{h}_G}{\partial \delta \mathbf{x}_G} \end{bmatrix} \\ &= \begin{bmatrix} -\mathbf{l}_{as_1}^i \top \hat{\mathbf{C}}_m^i [\mathbf{r}_{ma}^m]^\times & \mathbf{0}_{1 \times 3} & -\mathbf{l}_{as_1}^i \top & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 1 & 0 \\ & & \vdots & & & & & \\ -\mathbf{l}_{as_N}^i \top \hat{\mathbf{C}}_m^i [\mathbf{r}_{ma}^m]^\times & \mathbf{0}_{1 \times 3} & -\mathbf{l}_{as_N}^i \top & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 1 & 0 \\ -\mathbf{l}_{as_1}^i \top \hat{\mathbf{C}}_m^i [\hat{\boldsymbol{\omega}}_{im}^m \times \mathbf{r}_{ma}^m]^\times & -\mathbf{l}_{as_1}^i \top & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 1 \\ & & \vdots & & & & & \\ -\mathbf{l}_{as_N}^i \top \hat{\mathbf{C}}_m^i [\hat{\boldsymbol{\omega}}_{im}^m \times \mathbf{r}_{ma}^m]^\times & -\mathbf{l}_{as_N}^i \top & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & 0 & 0 & 1 \end{bmatrix} \quad (5.59) \end{aligned}$$

5.6.2 State transition matrix

As mentioned in Section 4.6.2, the clock bias is the integral of the clock drift. Assuming a constant clock drift model gives

$$\dot{\mathbf{x}}_G = \frac{d}{dt} \begin{bmatrix} b_\rho \\ b_{\dot{\rho}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} b_\rho \\ b_{\dot{\rho}} \end{bmatrix} = \mathbf{F}_G \mathbf{x}_G \quad (5.60)$$

Which constitutes the last two states of Equation (5.2)

5.6.3 GNSS Standalone Solution

The position and velocity of the receiver must be calculated from the pseudoranges and pseudorange-rates to be used in the loosely coupled integration filter. An iterative least-squares method can be used [39]. However, an algebraic solution derived by [10] will be used here. This is known as the Bancroft method. We group the satellite positions and measured pseudoranges in a vector as

$$\mathbf{a}_j = \begin{bmatrix} \mathbf{r}_{es_j}^e \\ \rho_j \end{bmatrix} \quad 1 \leq j \leq N \quad (5.61)$$

where N is the total number of observed measurements, which is required to be at least 4. Define the matrix \mathbf{A} as

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_N^\top \end{bmatrix} \in \mathbb{R}^{N \times 4} \quad (5.62)$$

along with the vectors \mathbf{i}_0 and \mathbf{r} defined by

$$\mathbf{i}_0 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \quad \mathbf{r} = \begin{bmatrix} \langle \mathbf{a}_1, \mathbf{a}_1 \rangle \\ \langle \mathbf{a}_2, \mathbf{a}_2 \rangle \\ \vdots \\ \langle \mathbf{a}_N, \mathbf{a}_N \rangle \end{bmatrix} \quad (5.63)$$

where the *Lorentz product* of two 4-elements vectors are defined as

$$\langle \mathbf{a}, \mathbf{b} \rangle = a_1 b_1 + a_2 b_2 + a_3 b_3 - a_4 b_4 \quad (5.64)$$

Compute the generalized inverse of \mathbf{A} , \mathbf{A}^\dagger , as well as the vectors \mathbf{u} and \mathbf{w} from

$$\mathbf{A}^\dagger = (\mathbf{A}^\top \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{W} \quad (5.65)$$

$$\mathbf{u} = \mathbf{A}^\dagger \mathbf{i}_0 \quad (5.66)$$

$$\mathbf{w} = \mathbf{A}^\dagger \mathbf{r} \quad (5.67)$$

where \mathbf{W} is a symmetric, positive definite weighting matrix. The identity matrix is used here, although the inverse of the measurement covariance matrix can also be used [10]. Next, find the roots $\lambda_{1,2}$ of

$$\langle \mathbf{u}, \mathbf{u} \rangle \lambda^2 + 2(\langle \mathbf{u}, \mathbf{w} \rangle - 1) \lambda + \langle \mathbf{w}, \mathbf{w} \rangle = 0 \quad (5.68)$$

The two solution candidates are then given by

$$\mathbf{y}_{1,2} = \lambda_{1,2} \mathbf{u} + \mathbf{w} \quad (5.69)$$

where

$$\mathbf{r}_{ea,i}^e = \begin{bmatrix} y_{i,1} \\ y_{i,2} \\ y_{i,3} \end{bmatrix}, \quad b_\rho = -y_{i,4} \quad (5.70)$$

The correct position and clock bias can be found by inserting the solutions into the original equation for the pseudorange, Equation (4.29). There will be agreement only in one case.

With a position solution, the line-of-sight matrix from the antenna to the satellites can be found by normalizing the vector from the estimated position to each satellite:

$$\mathbf{L} = \begin{bmatrix} -\mathbf{r}_{as_1}^e{}^\top / \|\mathbf{r}_{as_1}^e\| & 1 \\ -\mathbf{r}_{as_2}^e{}^\top / \|\mathbf{r}_{as_2}^e\| & 1 \\ \vdots & \vdots \\ -\mathbf{r}_{as_N}^e{}^\top / \|\mathbf{r}_{as_N}^e\| & 1 \end{bmatrix} = \begin{bmatrix} -\mathbf{l}_1^\top & 1 \\ -\mathbf{l}_2^\top & 1 \\ \vdots & \vdots \\ -\mathbf{l}_N^\top & 1 \end{bmatrix} \quad (5.71)$$

The LOS-matrix is augmented with a column of ones to account for the receiver clock states. Note that in the ideal noiseless case, the LOS-matrix is equivalent to \mathbf{A} defined in Equation (5.62) normalized by the last column.

The generalized inverse of \mathbf{L} is found as in (5.65):

$$\mathbf{L}^\dagger = (\mathbf{L}^\top \mathbf{W} \mathbf{L})^{-1} \mathbf{L}^\top \mathbf{W} \quad (5.72)$$

which can be used to calculate the velocity of the receiver antenna by [39]:

$$\begin{bmatrix} \mathbf{v}_{ea}^e \\ b_{\dot{\rho}} \end{bmatrix} = \mathbf{L}^\dagger (\dot{\boldsymbol{\rho}} - \dot{\boldsymbol{\rho}}_{\text{sat}}) \quad (5.73)$$

where

$$\dot{\boldsymbol{\rho}} = \begin{bmatrix} \dot{\rho}_1 \\ \vdots \\ \dot{\rho}_{N_s} \end{bmatrix} \quad \dot{\boldsymbol{\rho}}_{\text{sat}} = \begin{bmatrix} \mathbf{I}_1^\top \mathbf{v}_{es1}^e \\ \mathbf{I}_2^\top \mathbf{v}_{es2}^e \\ \vdots \\ \mathbf{I}_N^\top \mathbf{v}_{esN}^e \end{bmatrix} \quad (5.74)$$

5.7 Filter Reset

In a closed-loop error state formulation, the estimated error is applied to the attitude, velocity and position in the reference navigation system. This means the error has to be reset, else it is accumulated both in the strapdown system and the integration filter. For this section, let \mathbf{x}_- denote a state before the reset, and \mathbf{x}_+ after reset.

Notice from Table 4.2 that all the error states except the attitude is a linear function of the true state. For brevity, only the position error is considered in the reset procedure, and the other errors are updated identically. The state update is given by

$$\hat{\mathbf{r}}_{im+}^i = \hat{\mathbf{r}}_{im+}^i + \delta \hat{\mathbf{r}}_{im}^i \quad (5.75)$$

$$\begin{aligned} \hat{\mathbf{C}}_{m+}^i &= \hat{\mathbf{C}}_{m-}^i \delta \hat{\mathbf{C}}_i^m \\ &\approx \hat{\mathbf{C}}_{m-}^i (\mathbf{I}_3 - [\delta \boldsymbol{\psi}_{im}^m]^\times) \end{aligned} \quad (5.76)$$

The error state is reset by

$$\delta \mathbf{x} = \mathbf{0}_{18 \times 1} \quad (5.77)$$

The position update is just a linear relationship, such that

$$\mathbb{E} [\delta \mathbf{r}_{im+}^i \delta \mathbf{r}_{im+}^{i \top}] = \mathbb{E} [\delta \mathbf{r}_{im-}^i \delta \mathbf{r}_{im-}^{i \top}] \quad (5.78)$$

and the angle error transforms as

$$\begin{aligned} \mathbb{E} [\delta \boldsymbol{\psi}_{im+}^m \delta \boldsymbol{\psi}_{im+}^{m \top}] &= \mathbb{E} [\mathbf{C}_{m-}^{m+} \delta \boldsymbol{\psi}_{im-}^m - (\mathbf{C}_{m-}^{m+} \delta \boldsymbol{\psi}_{im-}^m)^\top] \\ &= \mathbf{C}_{m-}^{m+} \mathbb{E} [\delta \boldsymbol{\psi}_{im-}^m \delta \boldsymbol{\psi}_{im-}^{m \top}] \mathbf{C}_{m-}^{m+ \top} \end{aligned} \quad (5.79)$$

where \mathbf{C}_{m-}^{m+} is estimated by given by the error state update as

$$\begin{aligned}
\mathbf{C}_{m-}^{m+} &= \mathbf{C}_i^{m+} \mathbf{C}_{m-}^i \\
&\approx \hat{\mathbf{C}}_{m+}^i \top \hat{\mathbf{C}}_{m-}^i \\
&\approx \left(\hat{\mathbf{C}}_{m-}^i (\mathbf{I}_3 - [\delta\boldsymbol{\psi}_{im}^m]^\times) \right)^\top \hat{\mathbf{C}}_{m-}^i \\
&= \mathbf{I}_3 + [\delta\boldsymbol{\psi}_{im}^m]^\times
\end{aligned} \tag{5.80}$$

Which give the covariance reset as

$$\mathbf{P}_+ = \mathbf{G} \mathbf{P}_- \mathbf{G}^\top \tag{5.81}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_3 + [\delta\boldsymbol{\psi}_{im}^m]^\times & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{15} \end{bmatrix} \tag{5.82}$$

Part III

Results

System Testing and Results

In order to test the system, a true trajectory for the UAV needs to be determined, and sensor data must be generated. The sensor data generation is the same for all cases, and is described in the following section. Some general considerations, such as parameters, are given after that. Finally, each test case is described and results are presented.

6.1 Sensor measurement generation

6.1.1 IMU

Given values of acceleration, angular velocity and position of the UAV, the IMU outputs of specific force and angular velocity are straightforward to calculate. The specific force are a linear combination of acceleration and gravity with added bias and measurement noise:

$$\mathbf{f}_I = \mathbf{a}_{im}^m - \mathbf{C}_i^m \mathbf{g}^i + \mathbf{b}_a + \mathbf{w}_a \quad (6.1)$$

Gravity model

The term \mathbf{g}^i is the acceleration due to gravity. The earth gravitational model 2008 (EGM08) will be used to model the gravity. Matlab software implements this model in the ECEF-frame [1], and the interface is shown in Figure 6.1. Given a position \mathbf{r}_{em}^e , the EGM model outputs gravity values \mathbf{g}^e . These can be rotated back to the ECI-frame by the coordinate transformations described in Appendix A.

Bias model

The bias are generated as constant terms, and are drawn from a normal distribution:

$$\mathbf{b}_a \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \sigma_{bas}^2 \mathbf{I}_3) \quad (6.2)$$

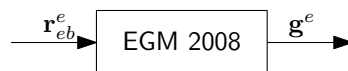


Figure 6.1: The EGM08, with interface from Matlab/Simulink implementation.

where the standard deviation σ_{bias} is called *bias repeatability*, and is usually found on the IMU data sheet.

The measurement noise \mathbf{w}_a is assumed to be WGN:

$$\mathbf{w}_a[n] \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \sigma_a^2 \mathbf{I}_3) \quad (6.3)$$

The angular velocity does not need to be corrected for gravity, such that the measurement generation is given directly by Equation (4.8). The bias is also considered constant:

$$\mathbf{b}_g \sim \mathcal{N}(\mathbf{0}_{3 \times 1}, \sigma_{bgs}^2 \mathbf{I}_3) \quad (6.4)$$

The measurement noise standard deviation is denoted σ_g .

The integrated specific force and integrated angular velocity can be calculated analytically if the closed-form expression for specific force and angular velocity is known. However, this is in most cases impractical, and a numerical integration method is preferred. A trapezoidal integration scheme is used, similar to the one used in the chosen IMU [8]. It is given by

$$\mathbf{v}_I[Dn] = \frac{T_s}{2} \sum_{d=0}^{D-1} \mathbf{f}_I[Dn-d] + \mathbf{f}_I[Dn-d-1] \quad (6.5)$$

where T_s is the internal sample rate, and $D > 0$ is the decimation rate. The output rate depends on the IMU sampling time and chosen decimation rate, and given by $\Delta t = DT_s$. The noise parameters can be applied before or after the summation. Rewrite the noise term of Equation (6.5) as

$$\begin{aligned} \mathbf{w}_v[Dn] &= \frac{T_s}{2} \sum_{d=0}^{D-1} \mathbf{w}_a[Dn-d] + \mathbf{w}_a[Dn-d-1] \\ &= \frac{T_s}{2} (\mathbf{w}_a[Dn] + \mathbf{w}_a[Dn-1] + \mathbf{w}_a[Dn-1] + \mathbf{w}_a[Dn-2] + \dots \\ &\quad \dots + \mathbf{w}_a[Dn-D+1] + \mathbf{w}_a[Dn-D+1] + \mathbf{w}_a[Dn-D]) \\ &= T_s \left(\frac{1}{2} \mathbf{w}_a[Dn] + \mathbf{w}_a[Dn-1] + \dots + \mathbf{w}_a[Dn-D+1] + \frac{1}{2} \mathbf{w}_a[Dn-D] \right) \end{aligned}$$

The covariance can then be expressed as the covariance of the sum:

$$\begin{aligned} \mathbf{R}_v &= T_s^2 \left(\frac{1}{2^2} \mathbf{R}_a + \mathbf{R}_a + \dots + \mathbf{R}_a + \frac{1}{2^2} \mathbf{R}_a \right) \\ &= T_s^2 \left(D - \frac{1}{2} \right) \mathbf{R}_a \end{aligned} \quad (6.6)$$

It should be noted that this trapezoidal integration scheme introduces correlation between samples, as the sample $\mathbf{w}_a[Dn]$ will appear in the expression for $\mathbf{w}_v[D(n+1)]$ as well.

Similar calculations can be done for the integrated angular velocity:

$$\boldsymbol{\alpha}_I = \frac{T_s}{2} \sum_{d=0}^{D-1} \boldsymbol{\omega}_I[Dn-d] + \boldsymbol{\omega}_I[Dn-d-1] \quad (6.7)$$

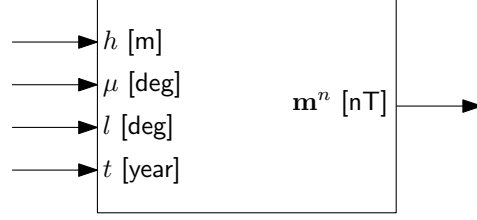


Figure 6.2: The WMM, with interface from the Matlab implementation.

The covariance of \mathbf{w}_α can be found using the same techniques as for the integrated specific force:

$$\mathbf{R}_\alpha = T_s^2 \left(D - \frac{1}{2} \right) \mathbf{R}_g \quad (6.8)$$

The accelerometer sensor data is also used in the leveling data.

6.1.2 Barometer

Barometer data is generated given the height of the UAV and the geoid height at the current location. Matlab software provides functions to calculate the geoid height, which is converted to a satellite position in the inertial frame by the transformations in Appendix A. The measurements is then calculated by Equation (4.17) with the height difference $h_m - h_0$ given by Equation (4.19).

6.1.3 Magnetometer

The magnetic measurements will be described by the World Magnetic Model (WMM), released by the U.S. National Oceanographic and Atmospheric Administration [7]. It describes the magnetic field by means of north, east and down intensity at a given position and time, and is released on a five-year basis.

Similarly to the gravity model, the WMM is also implemented in Matlab, and the interface is shown in Figure 6.2. This shows that the ECEF ellipsoidal coordinates and the current year is needed to calculate the local magnetic field, which is resolved in the NED-frame. The magnetic field varies slowly over time, and the year input is assumed to be constant.

After calculating the true local magnetic field with the true user position, the true DCM \mathbf{C}_n^m can be used to rotate the vector to the IMU-frame, as in Equation (4.22).

6.1.4 GNSS

To test the GNSS integration, a satellite model is needed. The constellation will follow simple circular orbits with constant velocity. The satellites orbital planes are assumed to be evenly distributed around the planet, and the satellites in a given orbital plane is assumed to be evenly spread around the orbit.

Given a constant circular orbit, the satellite positions can be given as

$$\mathbf{r}_{is_j}^i = \mathbf{C}_{s_j}^i \mathbf{r}_{is_j}^{s_j} \quad (6.9)$$

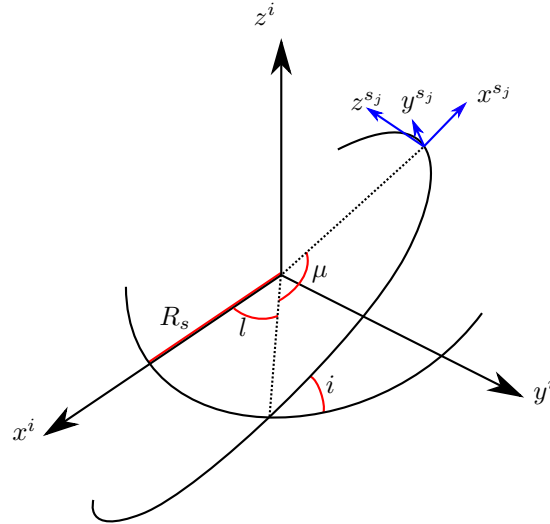


Figure 6.3: The satellite orbits are defined by a constant orbital radius and a set of three rotations.

where the satellite position in its own frame is given by the orbital radius of the constellation:

$$\mathbf{r}_{is_j}^{s_j} = \begin{bmatrix} R_s \\ 0 \\ 0 \end{bmatrix} \quad (6.10)$$

The DCM $\mathbf{C}_{s_j}^i$ can be found by a sequence of rotations. First, rotate the i -frame by an angle l about the z^i -axis. This angle is the *ascending node*. It describes the latitude where the orbital plane crosses the equator. Then, rotate by the inclination angle i , common to all the satellites, about the new x -axis. An inclination angle of 0 means the satellites follow the equator, which give low coverage at the poles. Last, rotate by an angle of μ about the new z -axis. This is called the *perigree* of the satellite. This gives

$$\mathbf{C}_{s_j}^i = \mathbf{C}_z(l)\mathbf{C}_x(i)\mathbf{C}_z(\mu) \quad (6.11)$$

The rotations are illustrated in Figure 6.3. The velocity of a satellite is given by

$$\begin{aligned} \mathbf{v}_{is_j}^i &= \frac{d}{dt} \mathbf{r}_{is_j}^i \\ &= \frac{d}{dt} \left(\mathbf{C}_{s_j}^i \mathbf{r}_{is_j}^{s_j} \right) \\ &= \dot{\mathbf{C}}_{s_j}^i \mathbf{r}_{is_j}^{s_j} + \mathbf{C}_{s_j}^i \mathbf{v}_{is_j}^{s_j} \\ &= \mathbf{C}_{s_j}^i \left[\boldsymbol{\omega}_{is_j}^{s_j} \right]^\times \mathbf{r}_{is_j}^{s_j} \\ &= \mathbf{C}_{s_j}^i \mathbf{v}_{is_j}^{s_j} \end{aligned} \quad (6.12)$$

Where the satellite angular velocity can be expressed as

$$\boldsymbol{\omega}_{is_j}^{s_j} = \begin{bmatrix} 0 \\ 0 \\ \omega_\mu \end{bmatrix} \quad (6.13)$$

Table 6.1: GNSS constellation parameters. The parameters are selected to mimic the american GPS constellation [25].

Sample rate	2.5 Hz
Number of orbital planes	6
Number of satellites per orbital plane	4
Satellite orbital radius	20200km
Satellite orbital period	12 hours
Orbit inclination angle	55°

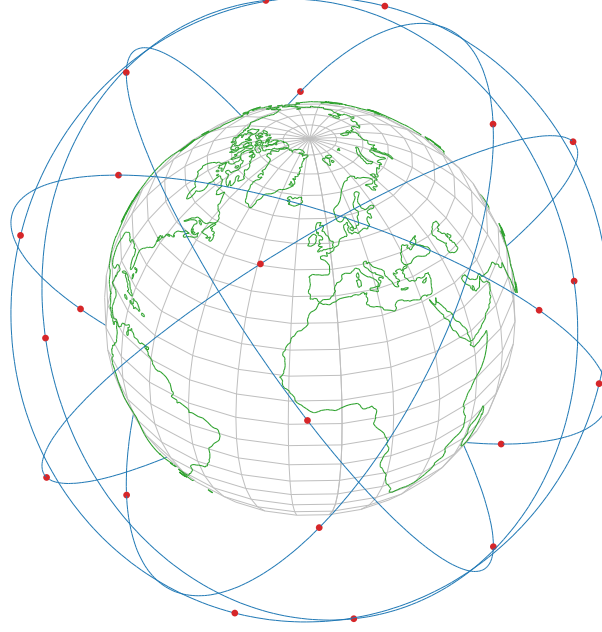


Figure 6.4: GNSS constellation generated by the parameters in Table 6.1. Initial satellite positions in red, satellite trajectories in blue.

due to constant inclination angle and ascending node. If T_μ is the orbital period of the satellite, $\omega_\mu = 2\pi/T_\mu$. This give

$$\mathbf{v}_{is_j}^i = \mathbf{C}_{s_j}^i \begin{bmatrix} 0 \\ \frac{2\pi R_s}{T_\mu} \\ 0 \end{bmatrix} \quad (6.14)$$

This describes the motion of a single satellite. To generate a full constellation, the parameters l and μ will be varied. The number of values of l will determine the number of satellite orbits, while μ will determine the number of satellites per orbit. This give the total number of satellites as $N_{sat} = N_l N_\mu$.

6.1.5 GNSS realization

The GNSS constellation is described by the parameters in Table 6.1. This give a total number of 24 satellites. To determine what satellites are able to provide measurements to the receiver, a mask angle will be set. It describes the maximum angle between the satellite position vector and the line-of-sight vector from the

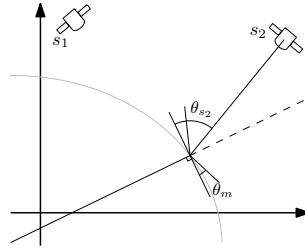


Figure 6.5: The mask angle determines if the satellite measurement is valid. Here, the measurement from s_1 will be rejected, and s_2 is valid.

antenna to the satellite. The criteria is given as

$$\theta_{s_j} > \theta_m \quad (6.15)$$

This is illustrated in Figure 6.5. Note that due to the flattening of the earth, the mask angle is not the minimum elevation required in the local NED-frame. However, this difference will be neglected.

6.2 Initialization test

This test simulates an initialization phase before takeoff. This is important to get a good estimate of the attitude, position and velocity before the UAV starts moving. It is also important to identify error parameters such as biases. It is known that there are three unobservables for stationary INS/GNS systems [38]. These three variables are combinations of accelerometer biases and attitude errors.

6.2.1 Setup and parameters

Since there are no information on the state of the system, all the navigation states will have to be initialized with some assumptions. If the location where the UAV is deployed is approximately known, the latitude and longitude can give an approximate initial position. In this particular case, the initial latitude and longitude were rounded down to the nearest integer, giving a position error of about 50km. With an approximate position, the initial velocity can be calculated by assuming the UAV is locally at rest. The coordinate transformation in Appendix A can be applied to calculate this velocity. Finally, the UAV attitude is assumed to align with the local NED-frame at the selected location. The bias parameters are initialized at zero.

Due to the large initial position error, the barometer has been disabled for the first 100 seconds. This is to let the GNSS handle the coarse positioning, and then adjust with the barometer.

To overcome the observability issue, motion is introduced to the system. The angle input can be seen in Figure 6.6, which simulates “rocking” the UAV back and forth. The position is assumed to be locally at rest throughout the simulation, which means the velocities and accelerations are zero.

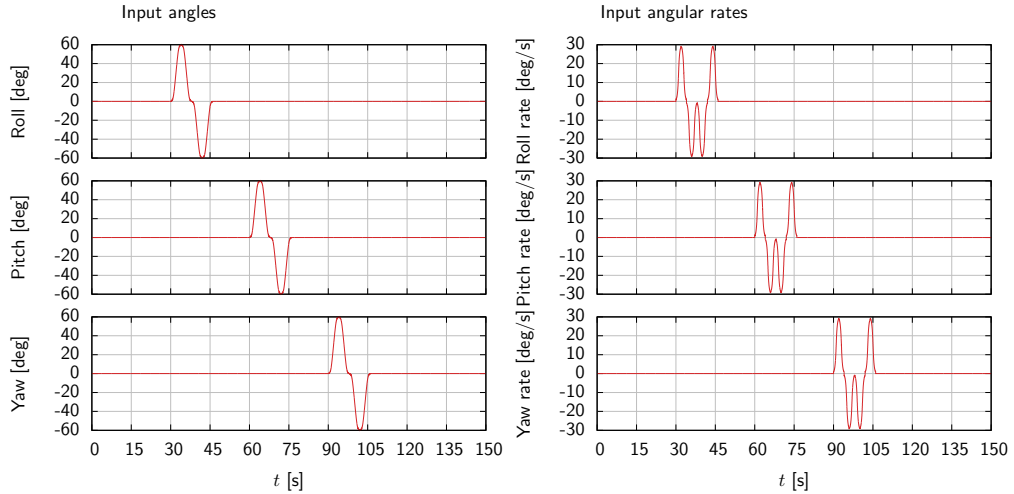


Figure 6.6: The input on the system in the initialization test.

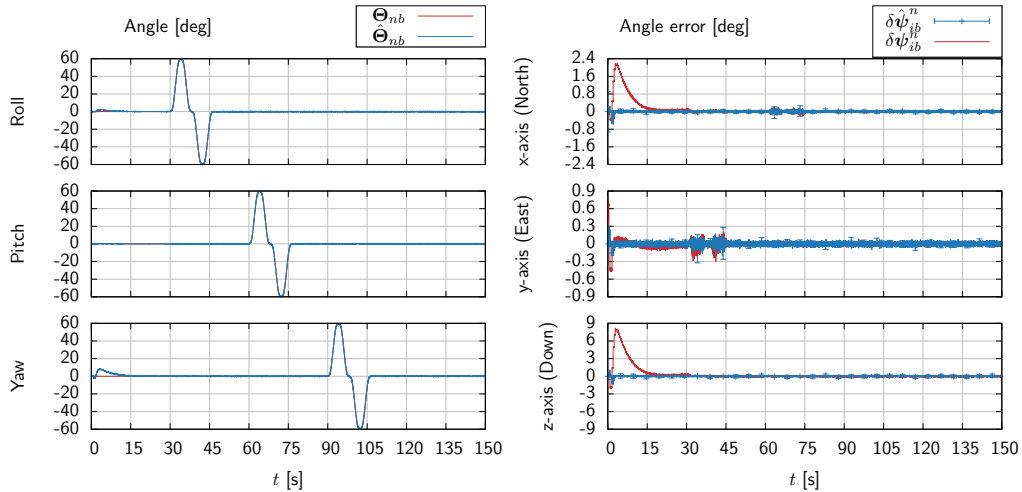


Figure 6.7: The euler angles and angle errors resolved about the NED-frame for the initialization test case.

6.2.2 Results

The estimated attitude and attitude errors can be seen in Figure 6.7. There is an initial transient of 9 degrees about the down-axis, which is resolved in the first 10 seconds. After this, there is a small steadystate error, less than a degree, about the north- and down-axis. This is resolved after the system starts moving after 30 seconds.

The velocity error can be seen in Figure 6.8. The transient velocity errors are large due to the initial position error, and the largest transients are gone after about 20 seconds. However, the transients are not completely gone until about 60 seconds into the simulation. The effect of enabling the barometer after 100 seconds is notable, and reduce the error estimates and covariance by about a third.

The position errors are seen in Figure 6.9. Once again, the large initial error cause quite large transient errors, and the estimates overshoot. Enabling the barometer is very effective for the position estimate and reduces the position error and covariance to about one tenth.

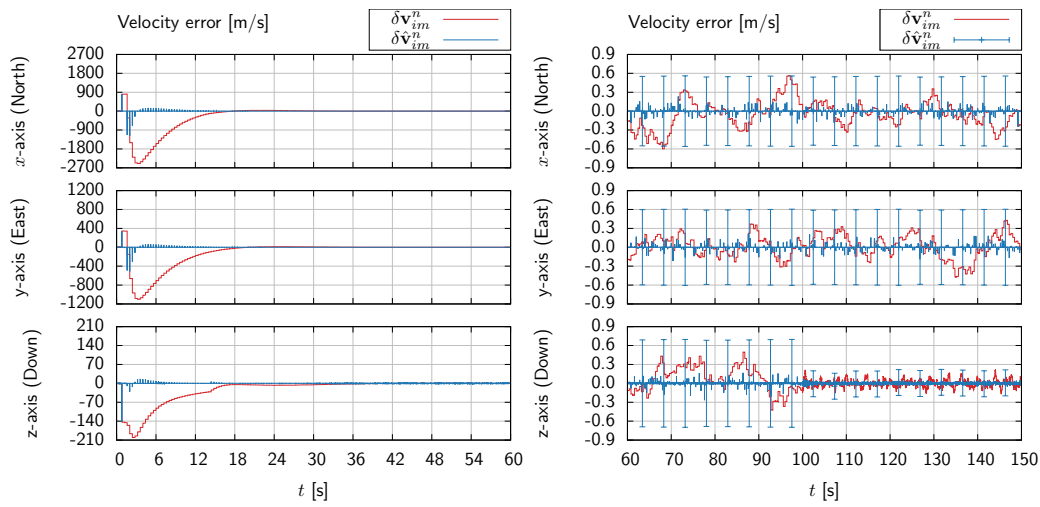


Figure 6.8: Velocity errors resolved in the NED-frame for the initialization test case, with the transient error on the left and steady-state on the right.

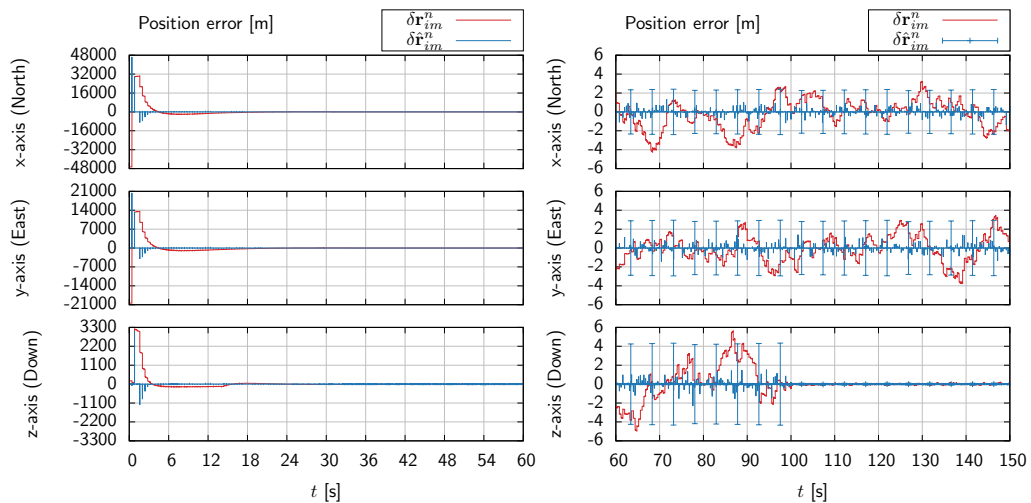


Figure 6.9: Position errors resolved in the NED-frame for the initialization test case, with the transient error on the left and steady-state on the right.

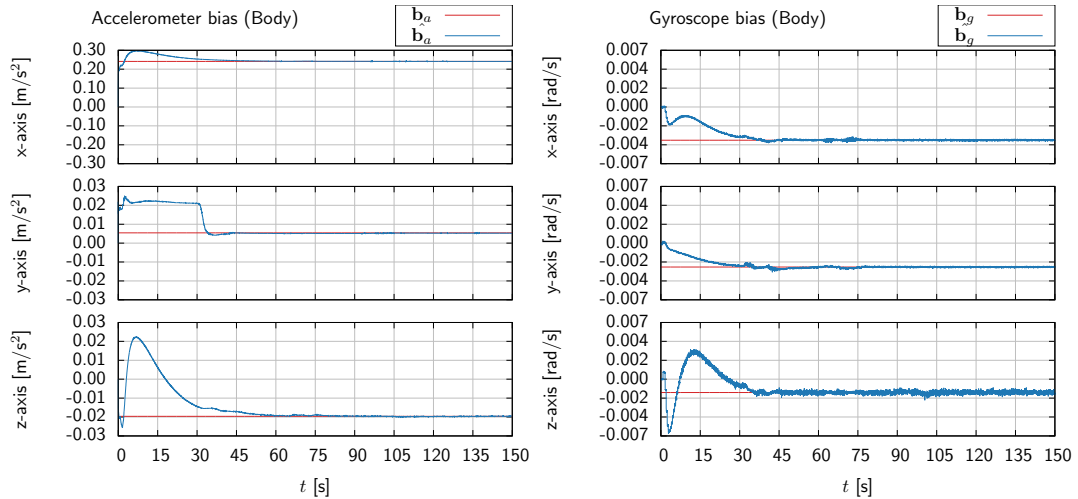


Figure 6.10: Bias estimates for the accelerometers and gyros for the initialization test case.

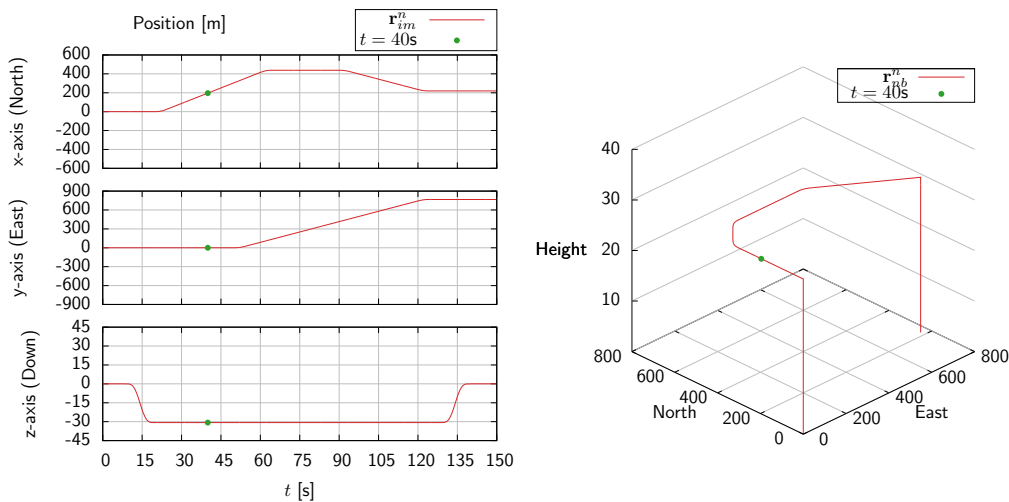


Figure 6.11: True position data for the reduced GNSS signal test.

The bias estimates are seen in Figure 6.10. There are notable transients in the accelerometer estimates, which can explain the attitude errors seen in Figure 6.7. The constant error up to 30 seconds in the y-axis accelerometer bias can also explain the constant attitude error. As with the angle errors, this offset is also resolved by tilting the UAV.

6.3 Reduced GNSS signal test

This test simulates a simple flight pattern, and the purpose is to compare loose and tight GNSS coupling.

6.3.1 Setup and parameters

The trajectory is seen in Figure 6.11, and lasts for 150 seconds. It is assumed that

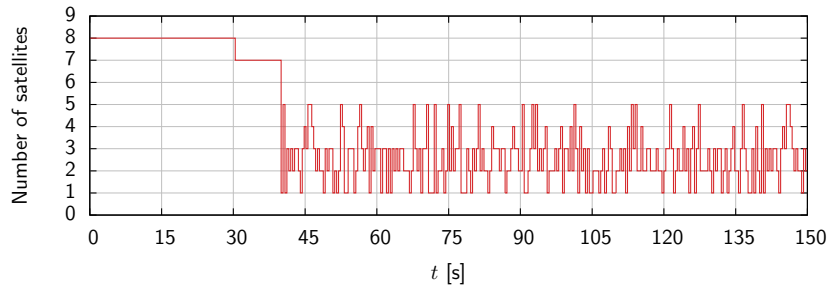


Figure 6.12: The available number of measurements for the GNSS receiver over time.

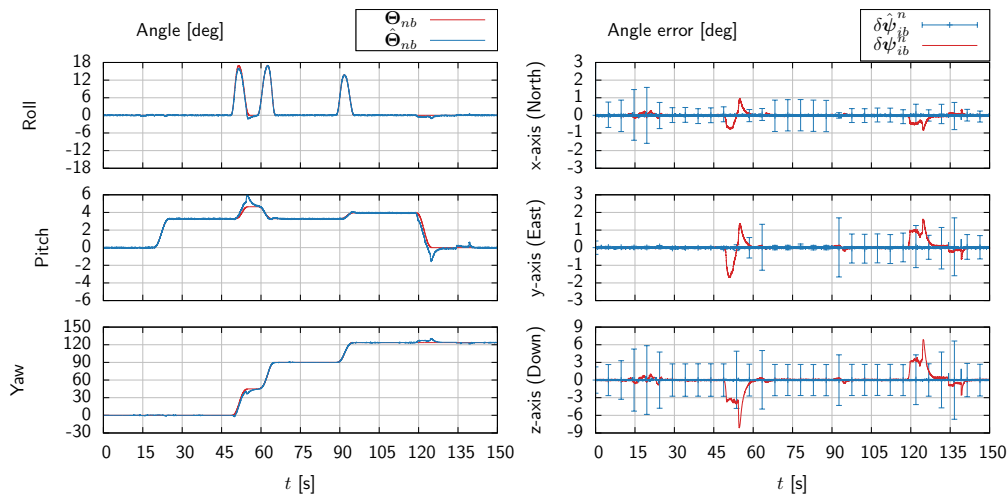


Figure 6.13: Angles and angle errors for the loose coupling. The error bars correspond to one standard deviation.

GNSS measurements are received from all satellites in view for the first 40 seconds. After this, the number measurements are reduced according to a pseudorandom pattern as shown in Figure 6.12.

Recall that 4 satellites are needed for a position and velocity fix in a loosely coupled system. This means there are several periods where there are no measurements available to the integration filter. To avoid the transient errors from the initialization test, the estimates are initialized at their true values.

6.3.2 Results

The angle and angle errors for the loosely coupled test case is shown in Figure 6.13. Maneuvers seems to affect the attitude estimate, with biggest error in the yaw angle. There are two notable errors, one between 50 and 60 seconds and one between 120 and 130 seconds. The uncertainty in the last error is detected by the filter, but the first error is not detected along the horizontal errors.

The velocity and velocity errors for the loosely coupled test is seen in Figure 6.14. The most notable errors seem to happen around the same time as the angle errors. This due to the transformation of integrated specific force from the body axis to the inertial axis. However, the errors seems to persist after the error angle has been corrected. The barometer seems to correct the errors along the vertical axis.

The position and position errors are seen in Figure 6.15. The constant velocity

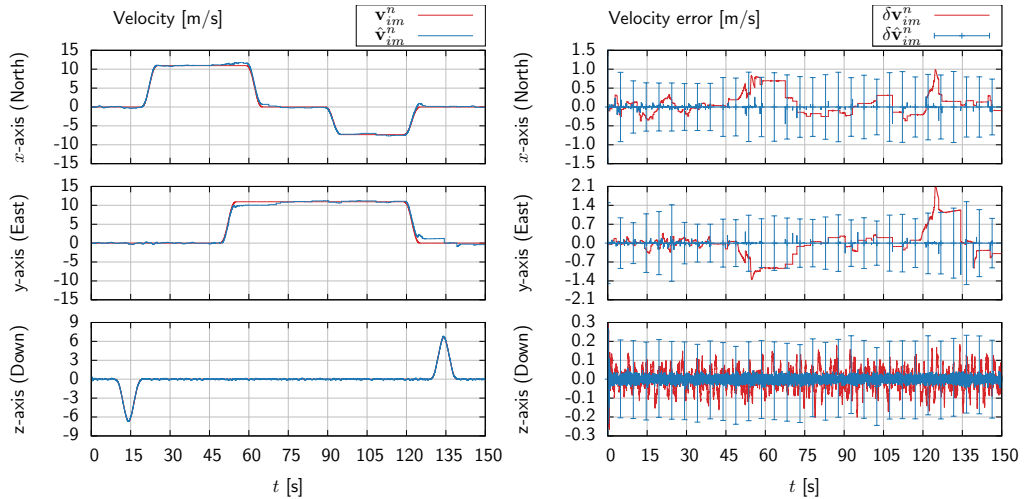


Figure 6.14: Velocity and velocity errors for the loose coupling. The error bars correspond to one standard deviation.

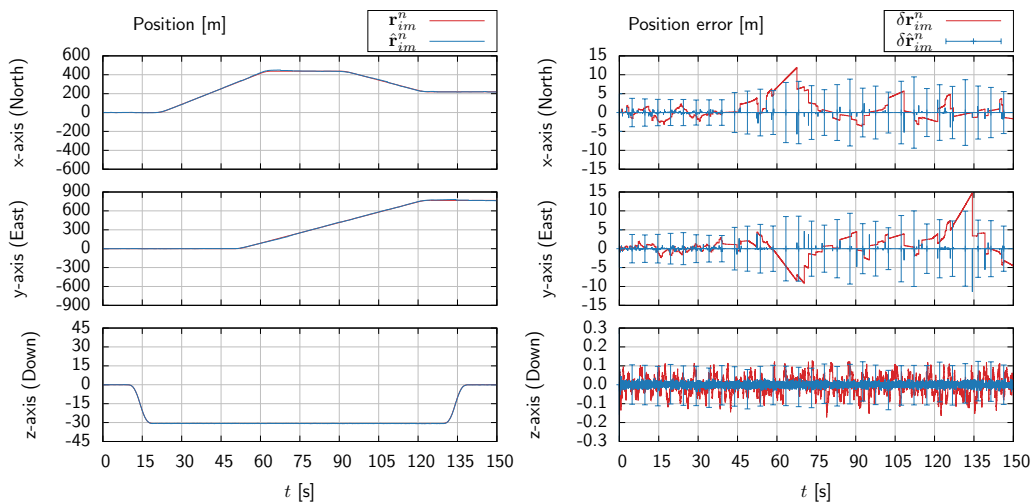


Figure 6.15: Position and position errors for the loose coupling. The error bars correspond to one standard deviation.

errors are seen as linearly increasing position errors. Other than these consequences from the attitude errors, the filter seems to capture the information loss. Once again, the barometer seems to aid in the vertical channel.

The angles and angle errors for the tightly coupled system is shown in Figure 6.16. The GNSS coupling does not influence the attitude measurement sensors, such that the behavior is expected to be similar to the loosely coupled test. The same attitude errors due to maneuvers are seen in the tightly coupled case.

The velocity and velocity errors for the tightly coupled system is shown in Figure 6.17. They are also influenced by the attitude errors. However, it seems the correct errors are estimated faster, and the transient after the misalignment is shorter. This is seen especially well during the attitude error at 125 seconds.

The position and position errors for the tightly coupled system is shown in Figure 6.18. The improved velocity correction give a noticeable performance increase in the position errors.

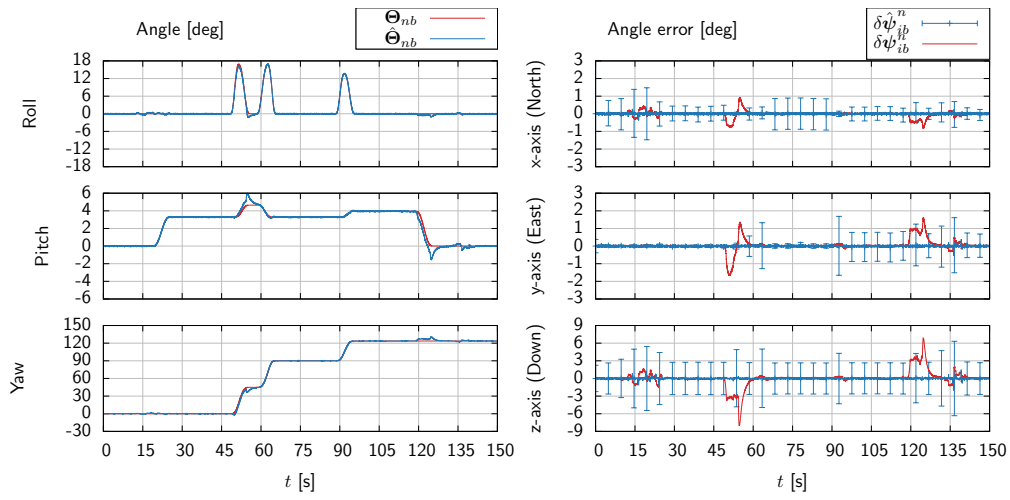


Figure 6.16: Angles and angle errors for the tight coupling. The error bars correspond to one standard deviation.

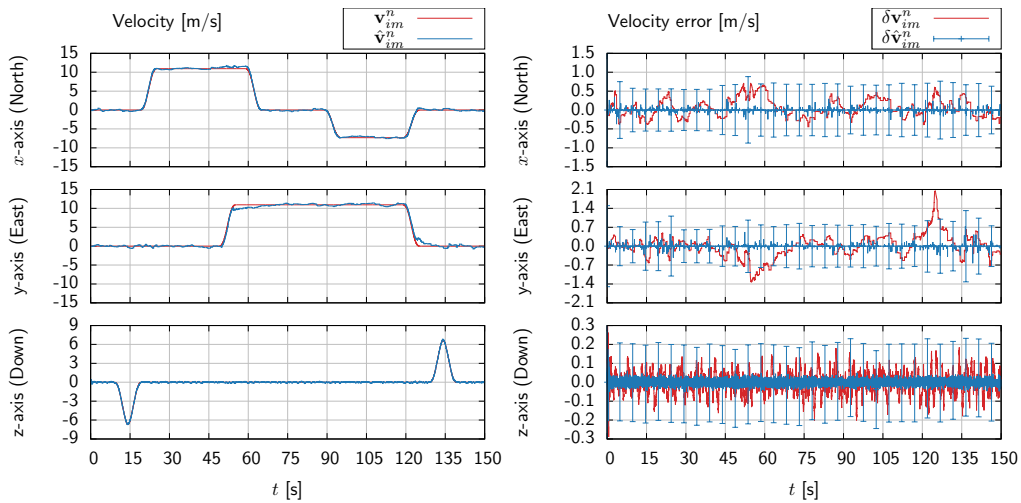


Figure 6.17: Velocity and velocity errors for the tight coupling. The error bars correspond to one standard deviation.

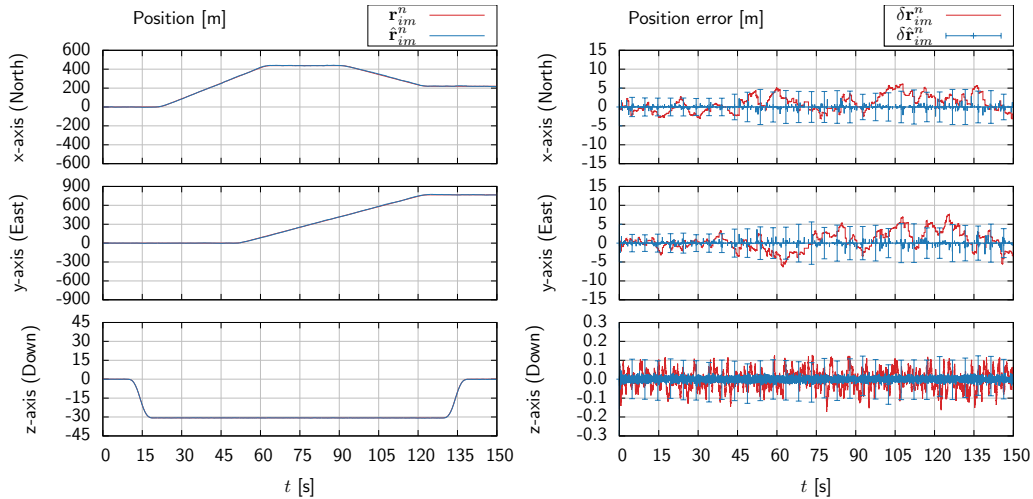


Figure 6.18: Position and position errors for the tight coupling. The error bars correspond to one standard deviation.

Figure 6.19 and Figure 6.20 compares the velocity and position for each case with a simulation with full coverage the entire simulation time. There are some errors that seems to coincide with the attitude error regardless of the signal strength, but the duration of the effects are notable in the reduced test case. Clearly, the tight integration manages to use the information from a reduced set of satellites to reduce the error much faster than the loose coupling.

6.4 Landing test

6.4.1 Setup and parameters

This test simulates the landing of the UAV under various conditions. The true trajectory is the same in all the cases, and can be seen in Figure 6.21. The system is initialized at the true state, and the position and velocity is perturbed by a zero-mean random value. The covariance for this variable is given by the steady-state covariance from the end of the initialization test.

The first case we consider is with all aids enabled. This will serve as a comparison for the later tests, and also be used to assert if the sensor configuration is sufficient for autonomous landing. The two following tests will disable all aiding sensors to assert the dead reckoning capabilities of the IMU and the strapdown equations.

6.4.2 Results

Figure 6.22 shows the velocity and velocity errors for the aided landing.

Figure 6.23 shows the position and position errors for the aided landing.

Next, the same maneuver is performed without any aids. Figure 6.24 shows the velocity and velocity errors. Since the velocity is initialized at an offset and there are no measurement updates, the estimates stay constant. However, there are not a big change, which means the inertial navigation system is able to detect the maneuver along the z-axis. The standard deviation of the error also grows, as should be

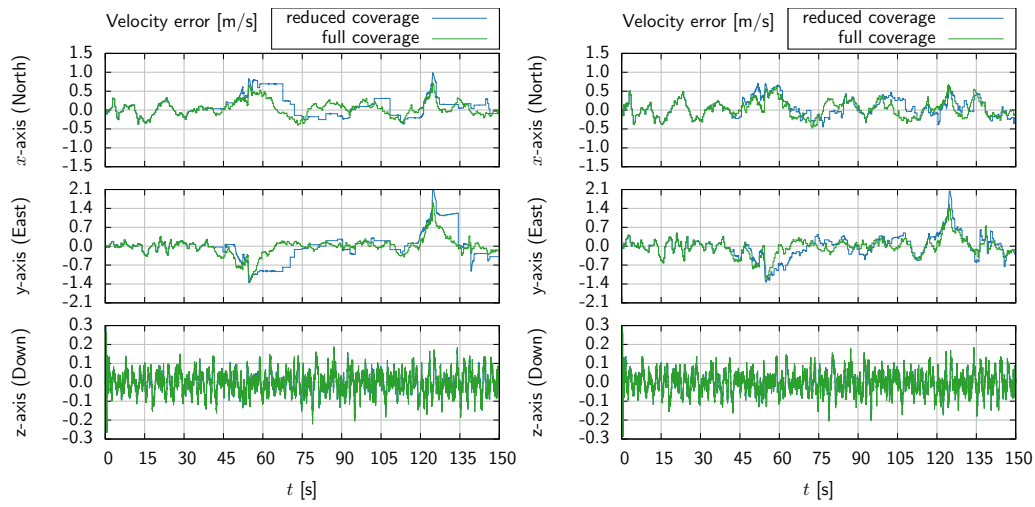


Figure 6.19: Velocity errors for the reduced GNSS simulations compared to a full coverage simulation. The loosely coupled system is on the left and the tightly coupled system on the right.

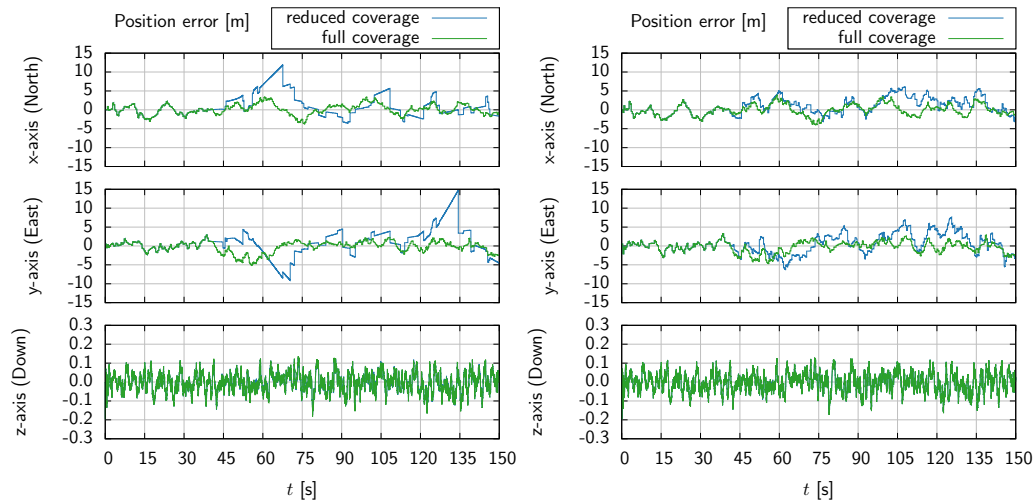


Figure 6.20: Position errors for the reduced GNSS simulations compared to a full coverage simulation. The loosely coupled system is on the left and the tightly coupled system on the right.

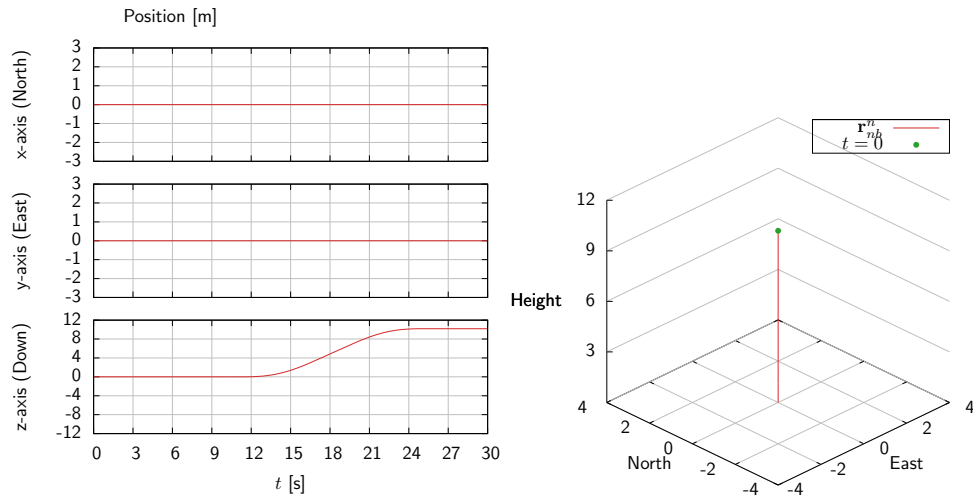


Figure 6.21: True trajectory for the landing test.

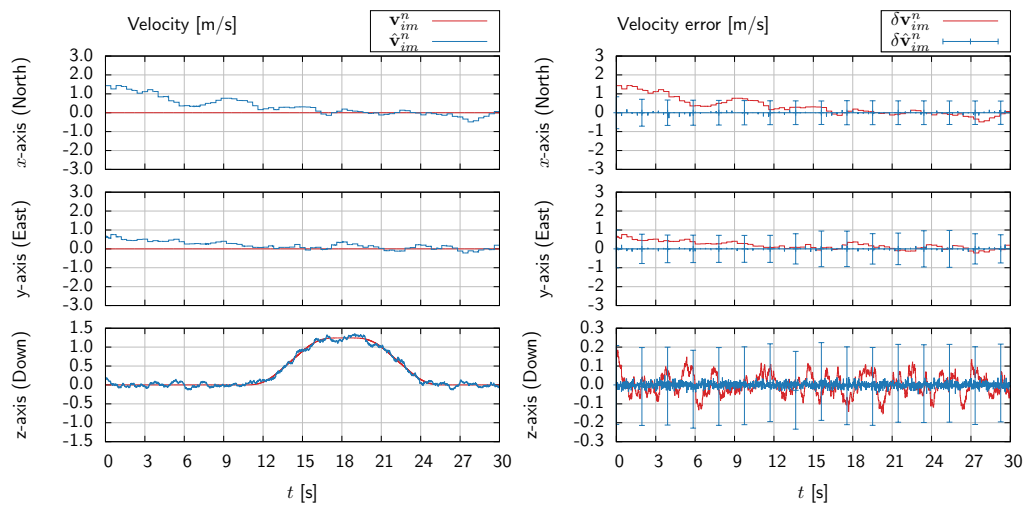


Figure 6.22: Velocity and velocity error estimates during landing with all aids enabled.

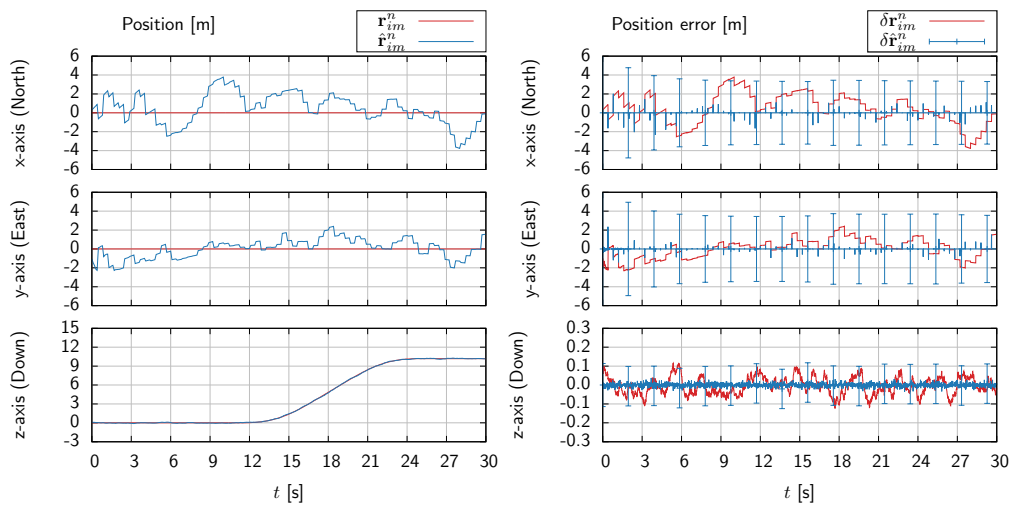


Figure 6.23: Position and position error estimates during landing with all aids enabled.

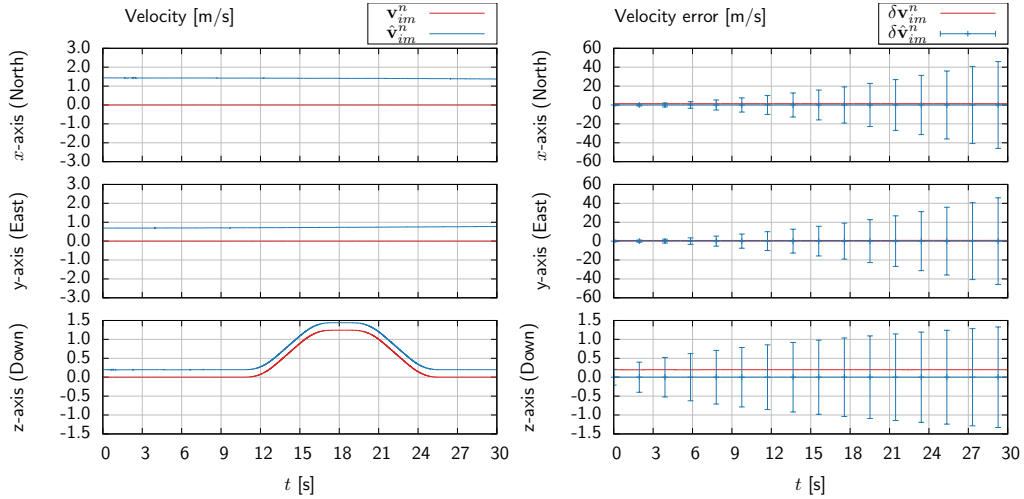


Figure 6.24: Velocity and velocity error estimates during landing without aids enabled.

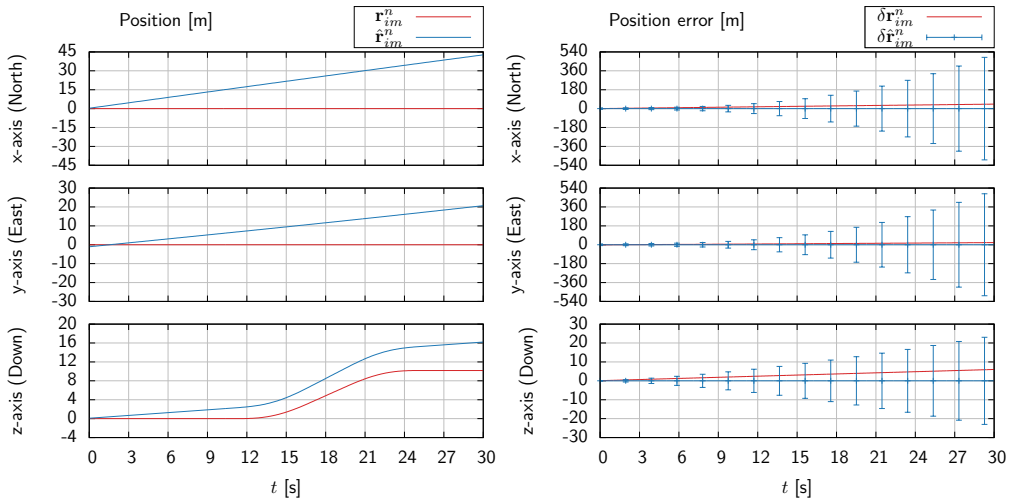


Figure 6.25: Position and position error estimates during landing without aids enabled.

expected when there are no measurements. The z-axis standard deviation grows a lot slower than the two others, due to the initial accuracy from the barometer.

Figure 6.25 shows the position and position errors. The constant velocity errors are integrated to linearly increasing position errors. Once again, the z-axis benefit from the added accuracy from the barometer.

Next, the initial standard deviation of the position and velocity estimates is reduced to a tenth. This is to test the potential of the inertial navigation algorithm with a more accurate sensor suite. Figure 6.26 shows the velocity and velocity errors. The constant offset is still present, but about a tenth of the magnitude. This is not really that surprising, given the reduction in standard deviation. The standard deviation of the error estimate, however, is significantly reduced.

Figure 6.27 shows the position and position errors. The reduced velocity error is directly influencing the position error, and the standard deviation is also significantly reduced.

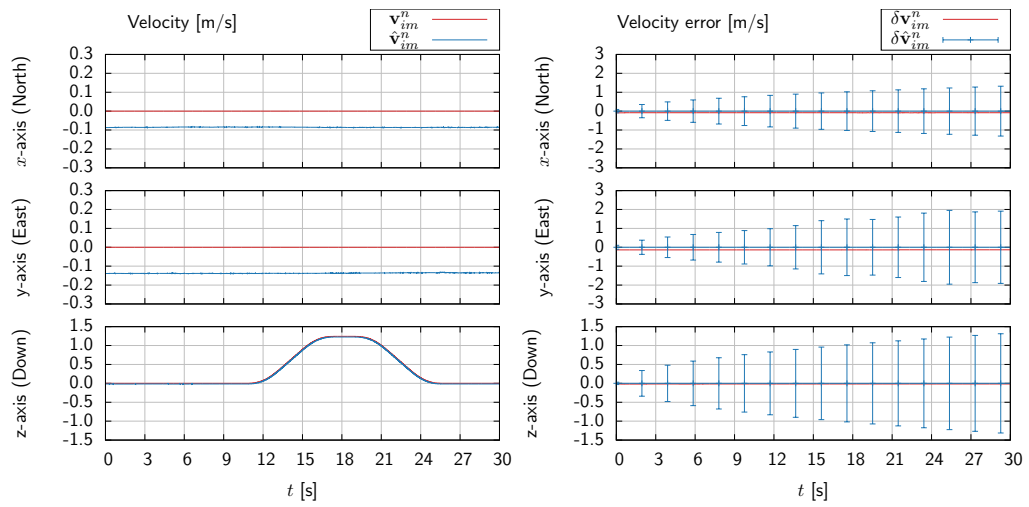


Figure 6.26: Velocity and velocity error estimates during landing without aids enabled. The standard deviation of the initial values have been reduced.

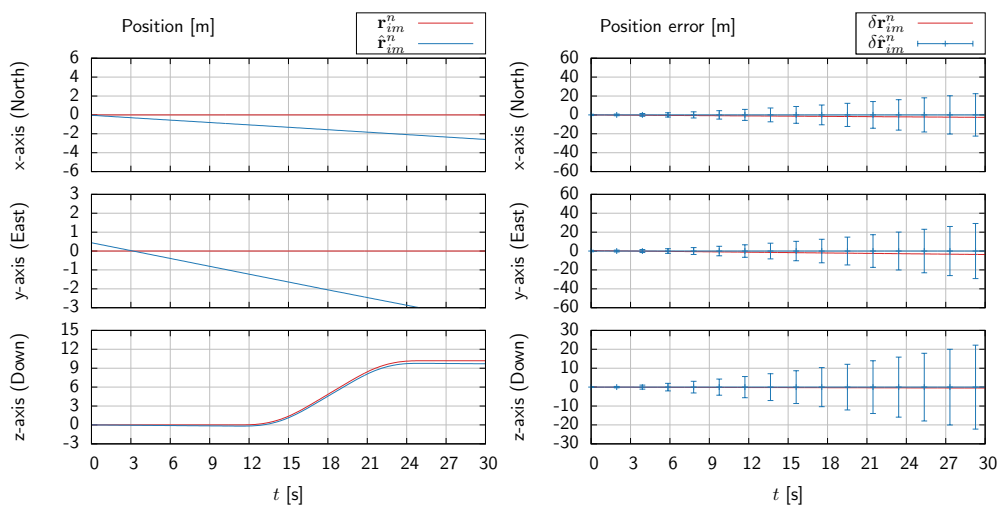


Figure 6.27: Position and position error estimates during landing without aids enabled. The standard deviation of the initial values have been reduced.

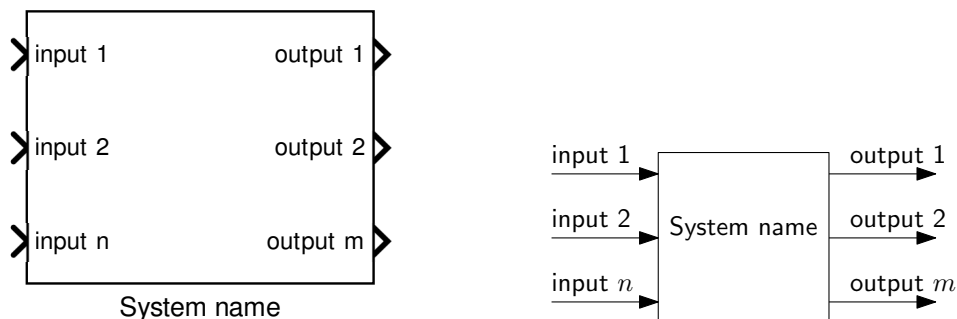
Simulink Models

This chapter will illustrate the components implemented in the Simulink framework to give a brief overview of the scope. It does not include every single operation, but the most essential components are described below. Constant parameters are set in the Matlab/Simulink workspace, and does not need to be input explicitly into the block. The blocks have been edited to be consistent with the notation used in the thesis and to remove unnecessary details. The relationship is shown in Figure 7.1.

The components of the individual sensor modules are shown in Table 7.2, Table 7.3, Table 7.4, and Table 7.5 or Table 7.6. The two GNSS interfaces have been configured with the same input/output to make changing between them easier.

The EKF components are shown in Table 7.7. Note that the sensor indices are not present in the EKF blocks. The sensors interface the EKF via modules that combines the measurement, the measurement prediction and the Jacobian in a single Simulink bus. This is to ensure the desired modularity, as changes in the sensor configuration is simply a change in the interface.

A few utility blocks are seen in Table 7.8. They have been used in various components of the implementation, mainly for rate transition, interpolation, reset and cumulative summation. Additionally, many of the basic operators used in Chapter 5 are native computations in Simulink, such as addition, multiplication and vector concatenation. These have been left out of this overview.



(a) A generic block from the Simulink software. (b) The corresponding block in this chapter.

Figure 7.1: The simulink block on the left correspond to the block on the right.

Table 7.1: Components of the strapdown computer.

Model	Notes
	Implements Equation (5.10).
	Implements Equation (5.12).
	Implements Equation (5.11) and Equation (5.13).
	Interface the gravity model described in Section 6.1.1 by a set of coordinate transformations.
	Decompose the error state in each component. This block also handles the rate transition between the strapdown system, the sensor blocks and the filter by inserting extra zeros.
	Implements the IMU correction.

Table 7.2: Components of the barometer module

Model	Notes
	Implements Equation (4.18).
	Implements Equation (4.19).
	Implements Equation (5.23).

Table 7.3: Components of the magnetometer module.

Model	Notes
	Interface the magnetic model described in Section 6.1.3 by a set of coordinate transformations.

Table 7.4: Components of the leveling module.

Model	Notes
	Implements Equation (5.36).
	Implements Equation (5.37) and Equation (5.38).

Table 7.5: Components of the loosely coupled GNSS module.

Model	Notes
	Demodulates the satellite data into vectors.
	Implements the Bancroft position method.
	Implements Equation (5.71).
	Implements Equation (5.73)

Table 7.6: Components of the tightly coupled GNSS module.

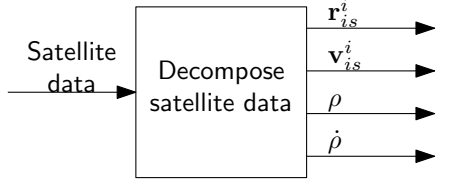
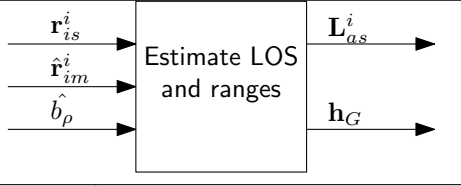
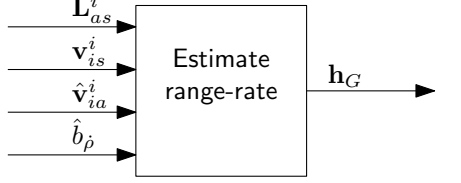
Model	Notes
 <p>Block diagram for "Decompose satellite data": An input arrow labeled "Satellite data" enters a rectangular block from the left. Four output arrows exit the block to the right, labeled from top to bottom as \mathbf{r}_{is}^i, \mathbf{v}_{is}^i, ρ, and $\dot{\rho}$.</p>	<p>Demodulates the satellite data into vectors.</p>
 <p>Block diagram for "Estimate LOS and ranges": Three input arrows enter a rectangular block from the left, labeled from top to bottom as \mathbf{r}_{is}^i, $\hat{\mathbf{r}}_{im}^i$, and \hat{b}_{ρ}. Two output arrows exit the block to the right, labeled from top to bottom as \mathbf{L}_{as}^i and \mathbf{h}_G.</p>	<p>Implements the range part of Equation (4.32)</p>
 <p>Block diagram for "Estimate range-rate": Four input arrows enter a rectangular block from the left, labeled from top to bottom as \mathbf{L}_{as}^i, \mathbf{v}_{is}^i, $\hat{\mathbf{v}}_{ia}^i$, and $\hat{b}_{\dot{\rho}}$. One output arrow exits the block to the right, labeled \mathbf{h}_G.</p>	<p>Implements the range-rate part of Equation (4.32)</p>

Table 7.7: Components of the Kalman Filter. All the blocks are found in Section 3.2.

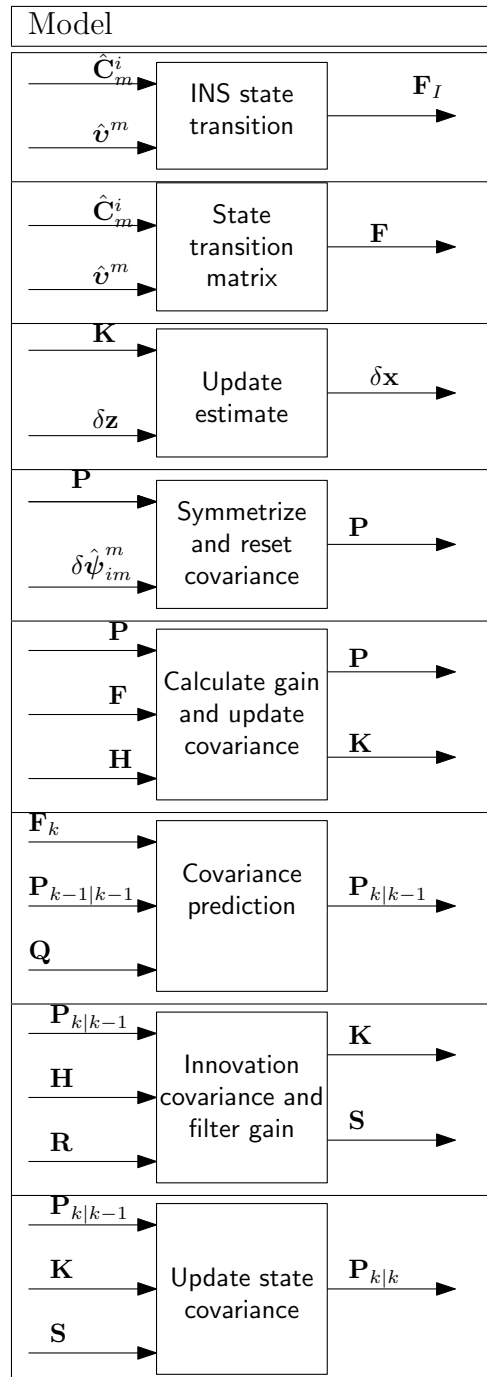
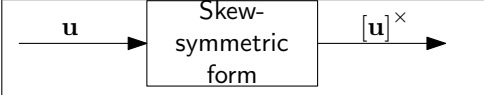
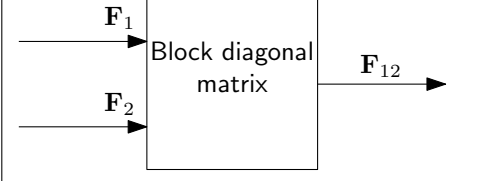
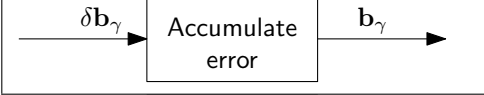
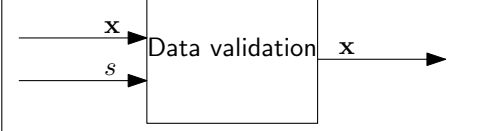


Table 7.8: Various utility blocks.

Model	Notes
	
	<p>Block diagonal form.</p> $\mathbf{F}_{12} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{0}_{m \times n} \\ \mathbf{0}_{n \times m} & \mathbf{F}_2 \end{bmatrix}$
	<p>Accumulates a general error signal, typically used for biases.</p>
	<p>The output is valid only if the flag s is set. This block is typically used to set measurements to zero between sampling.</p>

Part IV

Discussion and Conclusion

Discussion

The tests were only conducted using simple kinematic models, but they still provide some insight in the performance and challenges with the system.

8.1 Initialization

The gyro bias estimates seems to be determined regardless of the motion of the UAV. The accelerometer and magnetometer measurements indicate that the UAV is at rest, which allows the biases to converge to their true values.

However, the accelerometer and attitude errors are dependent on motion to be estimated. From a practical perspective, the idea of rotating the UAV might be undesirable. However, alternative is to allow the accelerometer bias and attitude measurement to have a constant offset during takeoff, which would hopefully be corrected during regular operations. Depending on the take-off method, this may not be crucial. For vertical take-off, the axes of primary interest are the x- and y-axes. The initial errors about these axes are not very large, such that a vertical take-off could still be possible with some unintended horizontal motion. This motion could be sufficient to correct the estimates.

8.2 Reduced GNSS signals

The loose and tight coupling seem to have comparable velocity and position errors when the receiver have signals from all the satellites in range, as seen in Figure 6.19 and Figure 6.20. The most notable errors occurs at about the 55 and 125 second mark, along the x- and y-axis. These are corrected relatively quickly, independent of coupling mode.

The same errors occur when a reduced number of satellite measurements are available. In this case, the difference between the loose and tight coupling is more notable. The loose coupling is unable to detect the velocity error due to few measurements in the time interval. This is demonstrated best in the north direction after 55 seconds and in the east direction after 55 and 125 seconds. This cause the velocity error to persist, and the position errors grows linearly, resulting in position errors up to about 15 meters.

The tight coupling is able to provide some corrections at each interval, and the velocity error is reduced more quickly. This improves the position estimates substantially, and the error is approximately halved compared to the loose coupling.

The velocity errors at the 55 and 125 second mark also occur when all the satellite measurements are available. This means these errors are caused by other effects. The attitude estimation is independent of the GNSS coupling, and Figure 6.13 and Figure 6.16 show the same effects. There are two main attitude errors which both occur at the same time the velocity is offset. An attitude error causes the strapdown computer to rotate the integrated specific force to the wrong frame, which causes an error in the velocity estimate.

It is evident that a good attitude estimate would reduce the velocity error during maneuvers, which again would reduce the effects of reduced satellite measurements. With accurate initial estimates of the attitude, velocity and position, a high quality IMU should be able to coast through the periods of reduced GNSS measurement availability without accumulating too large errors. This means that maneuver detection can be considered equally important for the purposes of mitigating velocity errors during periods of reduced GNSS measurement availability.

In Section 1.3.2, it was argued that the most critical phase of VTOL operations was during landing. The maneuvers during this phase should not be of the magnitude seen during these maneuvers, such that the effects during this phase will be less than the magnitude indicated here. However, the underlying problem of attitude errors during maneuvers should not be discarded.

Also note that a kinematic simulation model has been used to generate the data for testing, and that these combinations of maneuvers might be less of a problem with a more realistic UAV model.

8.3 Landing and dead reckoning

The third test trajectory concerned a simple landing model. With the aiding sensors enabled, the error along the z-axis were approximately 10 cm. This means the barometer could be able to provide sufficiently accurate data for autonomous landing. However, the simulation model has not considered environmental effects, which means the barometer is assumed to be calibrated before take-off, or has some relative measurement to provide precise estimates of the ground height. These relative measurements might be barometric data from a ground station (differential barometry), or some relative altitude measurements such as an ultrasonic range finder. The additional benefit of a range finder is another, independent height measurement, which improves the redundancy of the system.

Without the landing aids, the z-axis position error were about 6 meters after 30 seconds. This means that dead reckoning were unsuited for landing scenarios. However, this does not necessarily imply a low-quality strapdown system. The system were initialized with lower errors, and the horizontal errors were less than a meter after 30 seconds. The error is still too significant to be used for landing purposes, but the scenario shows the potential precision of the system without aiding sensors for longer time periods. Also note that the standard deviation of the estimates were considerably larger than accepted values during landing with dead reckoning.

Another aspect of autonomous landing is horizontal motion. Depending on the landing area, a moderate horizontal position error might not be problematic. However, it is important to have a good estimate of the horizontal error. Landing gears are usually not designed to handle large horizontal velocities, and large errors might cause damage to the UAV. With the accuracy of the current system, as seen in Figure 6.23, the horizontal velocity errors are probably on the borderline of what would be considered feasible. Dead reckoning landing with the current system is not possible. However, an improvement

Throughout this section, it has been argued that a better initial position estimate would significantly improve the dead reckoning results, as the inertial navigation system is quite capable. It is not immediately evident how to reduce these errors, especially in the horizontal direction. One possibility is to use differential or RTK GNSS systems, although that would require additional ground station hardware.

8.4 Further work

The next steps from this point is naturally divided into two parts: Further integration of the system into the LocalHawk framework, and developing methods to improve the navigation solution.

The first step of further integration is to set up the navigation system implementation with the guidance and control systems. This is a prerequisite for testing on the hardware platform and the airframe. When airframe design is finalized, the transformation between coordinate frames such as the body frame and the IMU-frame can be implemented. Designing an interface to the LocalHawk simulator would also be advantageous.

Of immediate research interest is attitude determination under maneuvers. Further research on horizontal velocity calculations exclusively with onboard sensors are also interesting. So far, most implemented landing systems utilize some form of aid from a ground station or vision-aided landing using markers.

Conclusion

This thesis has described an implementation of a navigation system for the Kongsberg LocalHawk UAV using an error state extended Kalman filter. The sensor configuration currently consists of an IMU as the reference navigation sensor and barometer, magnetometer and GNSS receiver as aiding sensors. The filter has been designed to allow changes in the sensor configuration, such that additional sensors are easily added.

The simulations indicate that the system is initialized correctly even under notable initial error, and the biases were correctly resolved by rotating the UAV. Tight coupling of GNSS improved the position and velocity estimates during reduced signals conditions. This was found to be especially important during maneuvers. The barometer provides valuable measurements of motion in the vertical direction, which is important during landing. However, horizontal motion estimation accuracy were bounded by GNSS.

The work in this thesis can be expanded in several ways, both in implementation and research. The implementation can be further integrated into the LocalHawk framework, such that the system can be validated experimentally. Additional sensors, such as an ultrasonic ranging sensor, would give both redundancy and additional accuracy for the purpose of detecting the height above ground. Another topic of future research is attitude determination during maneuvers, as accelerations disturb the leveling measurement. This attitude error affects the velocity estimates, and coping with them would be an interesting topic.

Appendices

Coordinate Transformations

The different sensor measurements are sometimes given in different coordinate frame. In addition, it might be advantageous to convert data to different frames for guidance and control purposes, for example in the body frame. The transformations are based on rigid body kinematics, which can be found in a classical mechanics textbook such as [16]. More specialized texts such as [23, 18, 39] provide calculations of the navigation-specific kinematics.

A.1 Earth-centered inertial and earth-centered earth-fixed coordinates

To convert between the ECEF and ECI-frame, we can utilize that the z-axes are coincident. This means that the DCM relating the ECI and ECEF-frames are the simple rotation about the z-axis:

$$\mathbf{C}_i^e = \mathbf{C}_z(\omega_e t) \tag{A.1}$$

where ω_e is the earth angular velocity, defined in [5], and t is the time since system initialization. This give the position, velocity and acceleration transformation as [23]

$$\mathbf{r}_{eb}^e = \mathbf{C}_i^e \mathbf{r}_{ib}^i \qquad \mathbf{r}_{ib}^i = \mathbf{C}_e^i \mathbf{r}_{eb}^e \tag{A.2}$$

$$\mathbf{v}_{eb}^e = \mathbf{C}_i^e \left(\mathbf{v}_{ib}^i - [\boldsymbol{\omega}_{ie}^i]^\times \mathbf{r}_{eb}^e \right) \qquad \mathbf{v}_{ib}^i = \mathbf{C}_e^i \left(\mathbf{v}_{eb}^e + [\boldsymbol{\omega}_{ie}^e]^\times \mathbf{r}_{eb}^e \right) \tag{A.3}$$

$$\mathbf{a}_{eb}^e = \mathbf{C}_i^e \left(\mathbf{a}_{ib}^i - 2 [\boldsymbol{\omega}_{ie}^i]^\times \mathbf{v}_{ib}^i + [\boldsymbol{\omega}_{ie}^i]^\times 2 \mathbf{r}_{ib}^i \right) \tag{A.4}$$

$$\mathbf{a}_{ib}^i = \mathbf{C}_e^i \left(\mathbf{a}_{eb}^e + 2 [\boldsymbol{\omega}_{ie}^e]^\times \mathbf{v}_{eb}^e + [\boldsymbol{\omega}_{ie}^e]^\times 2 \mathbf{r}_{eb}^e \right) \tag{A.5}$$

A.2 ECEF cartesian and ellipsoidal coordinates

For this section, let x , y and z be cartesian coordinates in an earth-centered, earth-fixed coordinate frame, and μ , l and h be the geodetic latitude, longitude and height above the ellipsoid for the same point. Conversion from ellipsoidal to cartesian

coordinates are straightforward [26]:

$$x = (N(\mu) + h) \cos \mu \cos l \quad (\text{A.6})$$

$$y = (N(\mu) + h) \cos \mu \sin l \quad (\text{A.7})$$

$$z = \left(\frac{r_p^2}{r_e^2} N(\mu) + h \right) \sin \mu \quad (\text{A.8})$$

where r_e and r_p denotes the equatorial and polar radius of the ellipsoid, and the normal radius of curvature N is given by

$$N(\mu) = \frac{r_e^2}{\sqrt{r_e^2 \cos^2 \mu + r_p^2 \sin^2 \mu}} \quad (\text{A.9})$$

For conversion from cartesian to ellipsoidal coordinates, the choice of geodetic versus geocentric latitude creates a nontrivial coupling between latitude and height. While closed-form solutions exist, an iterative algorithm is usually applied [39]. The equations are given by

$$\tan l = \frac{y}{x} \quad (\text{A.10})$$

$$\tan \mu = \frac{z}{\sqrt{x^2 + y^2}} \left(1 - \frac{(r_e^2 - r_p^2) N(\mu)}{r_e^2 (N(\mu) + h)} \right)^{-1} \quad (\text{A.11})$$

$$h = \frac{\sqrt{x^2 + y^2}}{\cos \mu} - N(\mu) \quad (\text{A.12})$$

which can be solved iteratively by the following procedure [18]:

1. Compute an approximate latitude $\mu_0 = \text{atan} \left[\frac{z(1-e^2)^{-1}}{\sqrt{x^2+y^2}} \right]$, where $e = \sqrt{1 - (r_p/r_e)^2}$.
2. Compute the corresponding normal radius of curvature $N(\mu_0)$ from Equation (A.9).
3. Compute a height estimate by $h = \frac{\sqrt{x^2+y^2}}{\cos \mu_0} - N(\mu_0)$.
4. Compute an improved latitude by $\mu = \text{atan} \left[\frac{z}{\sqrt{x^2+y^2}} \left(1 - e^2 \frac{N(\mu_0)}{N(\mu_0)+h} \right)^{-1} \right]$.
5. If $\mu \neq \mu_0$ within acceptable tolerance limits, let $\mu_0 = \mu$ and go back to step 2.
6. Compute the longitude by $l = \text{atan} \left[\frac{y}{x} \right]$.

A.3 ECEF to NED

The local NED plane depends on the location on the earth, and the transformation is given as [18]:

$$\mathbf{C}_n^e = \begin{bmatrix} -\cos l \sin \mu & -\sin l & -\cos l \cos \mu \\ -\sin l \sin \mu & \cos l & -\sin l \cos \mu \\ \cos \mu & 0 & -\sin \mu \end{bmatrix} \quad (\text{A.13})$$

A.4 NED to body

One of the most common methods to describe the body frame relative to the NED-frame is the roll-pitch-yaw convention [16]. This consists of a rotation about the initial z-axis, then a rotation about the resulting y-axis and finally a rotation about the resulting x-axis. This give the DCM from frame n to b as

$$\mathbf{C}_n^b = \mathbf{C}_z(\psi)\mathbf{C}_y(\theta)\mathbf{C}_x(\phi) \quad (\text{A.14})$$

In addition to having only three parameters, these three Euler angles are also easy to visualize. They have a disadvantage, however, in that they have a singularity at 90° pitch:

$$\begin{aligned} \mathbf{C}_z(\psi)\mathbf{C}_y(\pi/2)\mathbf{C}_x(\phi) &= \begin{bmatrix} 0 & \cos \psi \sin \phi - \sin \psi \cos \phi & \sin \psi \sin \phi + \cos \psi \cos \phi \\ 0 & \cos \psi \cos \phi + \sin \psi \sin \phi & \sin \psi \cos \phi - \cos \psi \sin \phi \\ -1 & 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \sin(\phi - \psi) & \cos(\phi - \psi) \\ 0 & \cos(\phi - \psi) & -\sin(\phi - \psi) \\ -1 & 0 & 0 \end{bmatrix} \end{aligned} \quad (\text{A.15})$$

The angles ϕ and ψ only appear through their difference, such that they cannot be determined with a pitch angle of 90° . Because of this weakness, the euler angels will not be used in the internal mechanization of the algorithm. The DCM will be used instead.

A.5 Transposition of navigation solutions

The different sensors are located in different positions on the UAV. To properly integrate the sensor data, they need to reference the same point. Assume that the sensor B has fixed a attitude and location relative to the airframe b . The position of the sensor in the ECI-frame is then given by

$$\begin{aligned} \mathbf{r}_{iB}^i &= \mathbf{r}_{ib}^i + \mathbf{r}_{bB}^i \\ &= \mathbf{r}_{ib}^i + \mathbf{C}_a^i \mathbf{r}_{bB}^b \end{aligned} \quad (\text{A.16})$$

and the velocity is found from differentiation:

$$\begin{aligned} \mathbf{v}_{iB}^i &= \frac{d}{dt} (\mathbf{r}_{ib}^i + \mathbf{C}_b^i \mathbf{r}_{bB}^b) \\ &= \mathbf{v}_{ib}^i + \dot{\mathbf{C}}_b^i \mathbf{r}_{bB}^b + \mathbf{C}_b^i \dot{\mathbf{r}}_{bB}^b \\ &= \mathbf{v}_{ib}^i + \mathbf{C}_b^i [\boldsymbol{\omega}_{ib}^b]^\times \mathbf{r}_{bB}^b \end{aligned} \quad (\text{A.17})$$

where it has been used that the lever arm is constant in the body frame.

Bibliography

- [1] Matlab Aerospace Toolbox User's Guide. http://se.mathworks.com/help/pdf_doc/aerotbx/aerotbx_ug.pdf. Accessed: 13.05.2015.
- [2] Matlab Function Reference. http://se.mathworks.com/help/releases/R2015a/pdf_doc/matlab/matlab_refbook.pdf. Accessed: 22.06.2015.
- [3] MTi-G User Manual and Technical Documentation. https://www.xsens.com/wp-content/uploads/2013/11/MTi-G_User_Manual_and_Technical_Documentation.pdf. Accessed: 25.06.2015.
- [4] U.S. Standard Atmosphere. Technical report, National Oceanic and Atmospheric Administration, 1976.
- [5] Department of defense world geodetic system 1984. its definition and relationship with local geodetic systems. Technical report, National Imagery and Mapping Agency, 2004.
- [6] IEEE Standard for Inertial Systems Terminology. *IEEE Std. 1559-2009*, 2009.
- [7] The US/UK World Magnetic Model for 2010-2015. Technical report, National Oceanographic and Atmospheric Administration, 2010.
- [8] Analog Devices. *ADIS 16488A Datasheet*, 2014. Rev. A.
- [9] Antonio Angrisano, Mark Petovello, and Giovanni Pugliano. Benefits of Combined GPS/GLONASS with Low-Cost MEMS IMUs for Vehicular Urban Navigation. *Sensors*, 12(4):5134–5158, April 2012.
- [10] Stephen Bancroft. An Algebraic Solution of the GPS Equations. *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(7):56–59, 1985.
- [11] Yaakov Bar-Shalom, X. Rong Li, and Thiagalingam Kirubarajan. *Estimation with Application To Tracking and Navigation*. John Wiley & Sons, Inc., 2001.
- [12] Robert G. Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, Inc., 2012.
- [13] HaiYang Chao, YongCan Cao, and YangQuan Chen. Autopilots for small unmanned aerial vehicles: A survey. *International Journal of Control, Automation and Systems*, 8(1):36–44, February 2010.

- [14] Joseph Conroy, Gregory Gremillion, Badri Ranganathan, and J. Sean Humbert. Implementation of wide-field integration of optic flow for autonomous quadrotor navigation. *Autonomous Robots*, 27(3):189–198, August 2009.
- [15] John L. Crassidis, F. Landis Markley, and Yang Cheng. Survey of Nonlinear Attitude Estimation Methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.
- [16] Olav Egeland and Jan Tommy Gravdahl. *Modeling and Simulation for Automatic Control*. Tapir Trykkeri, Norway, 2002.
- [17] Mark Euston, Paul Coote, Robert Mahony, Jonghyuk Kim, and Tarek Hamel. A complementary filter for attitude estimation of a fixed-wing UAV. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 340–345, 2008.
- [18] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Inc., 2011.
- [19] Kenneth Gade. A Non-singular Horizontal Position Representation. *Journal of Navigation*, 63(03):395–417, 2010.
- [20] Håvard Fjær Grip, Thor I. Fossen, Tor a. Johansen, and Ali Saberi. Attitude estimation based on time-varying reference vectors with biased gyro and vector measurements. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 18(PART 1):8497–8502, 2011.
- [21] Jason Gross, Yu Gu, Srikanth Gururajan, Brad Seanor, and Marcello Napolitano. A Comparison of Extended Kalman Filter, Sigma-Point Kalman Filter, and Particle Filter in GPS/INS Sensor Fusion. In *AIAA Guidance, Navigation, and Control Conference*, Reston, Virginia, August 2010. American Institute of Aeronautics and Astronautics.
- [22] Jason N. Gross, Yu Gu, Matthew B. Rhudy, Srikanth Gururajan, and Marcello R. Napolitano. Flight-Test Evaluation of Sensor Fusion Algorithms for Attitude Estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 48(3):2128–2139, July 2012.
- [23] Paul D. Groves. *Principles of GNSS, Inertial and Multisensor Integrated Navigation Systems*. Artech House, 2013.
- [24] Fredrik Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, Sweden, 2012.
- [25] Bernard Hofmann-Wellenhof, Herbert Lichtenegger, and Elmar Wasle. *GNSS — Global Navigation Satellite Systems*. Springer Verlag Wien, 2008.
- [26] Bernard Hofmann-Wellenhof and Helmut Moritz. *Physical Geodesy*. Springer Verlag Wien, 2006.
- [27] J.C. Hung, J.S. Hunter, W.W. Stripling, and H. V. White. Size effect on navigation using a strapdown IMU. Technical report, U.S army missile research and development command, 1979.

- [28] Rudolf Emil Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 1960.
- [29] Myron Kayton and Willis G. Wing. Attitude and heading references. In *Avionics Navigation Systems*, pages 426–448. John Wiley & Sons, Inc., 2007.
- [30] Farid Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378, 2012.
- [31] Jong Hyuk Kim, Salah Sukkarieh, and Stuart Wishart. Real-time navigation, guidance, and control of a UAV using low-cost sensors. *Springer Tracts in Advanced Robotics*, 24:299–309, 2006.
- [32] Robert C. Leishman, Timothy W. McLain, and Randal W. Beard. Relative Navigation Approach for Vision-Based Aerial GPS-Denied Navigation. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 74(1-2):97–111, October 2014.
- [33] Robert Mahony, Tarek Hamel, and Jean Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, 2008.
- [34] Marco Mammarella, Giampiero Campa, Mario L. Fravolini, and Marcello R. Napolitano. Comparing optical flow algorithms using 6-DOF motion of real-world rigid objects. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 42(6):1752–1762, 2012.
- [35] Torsten Merz, Simone Duranti, and Gianpaolo Conte. Autonomous landing of an unmanned helicopter based on vision and inertial sensing. *Springer Tracts in Advanced Robotics*, 21(c):343–352, 2006.
- [36] Cleve Moler and Charles Van Loan. Nineteen Dubious Ways to Compute the Exponential of a Matrix, Twenty-Five Years Later. *SIAM Review*, 45(1):3–49, 2003.
- [37] T. Puls and a. Hein. Outdoor position estimation and autonomous landing algorithm for quadrocopters using a wireless sensor network. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pages 285–290, 2010.
- [38] Ihnsoek Rhee, Mamoun F. Abdel-Hafez, and Jason L. Speyer. Observability of an integrated GPS/INS during maneuvers. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2):526–535, 2004.
- [39] Bjørnar Vik. *Integrated Satellite and Inertial Navigation Systems*. Department of Engineering Cybernetics, Norwegian University of Science and Technology, 2014.
- [40] S.I. Roumeliotis, G.S. Sukhatme, and G.a. Bekey. Circumventing dynamic modeling: evaluation of the error-state Kalman filter applied to mobile robot localization. *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, 2, 1999.

- [41] Srikanth Saripalli, James F Montgomery, and Gaurav S Sukhatme. Visually-Guided Landing of an Unmanned Aerial Vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–381, 2003.
- [42] Paul G. Savage. *Strapdown Analytics*. Strapdown Associates, Inc, 2000.
- [43] Mohammad Shabani, Asghar Gholami, and Narjes Davari. Asynchronous direct Kalman filtering approach for underwater integrated navigation system. *Nonlinear Dynamics*, 80:71–85, 2014.
- [44] D Michael Sobers. Indoor Navigation for Unmanned Aerial Vehicles. *AIAA Guidance, Navigation, and Control Conference*, (August), 2009.
- [45] David Titterton and Jessie Weston. *Strapdown Inertial Navigation Technology*. American Institute of Aeronautics and Astronomics, 2004.
- [46] 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*, 53(1):21–31, 2006.
- [47] Erik Wilthil. A singularity-free inertial navigation system for an unmanned aerial vehicle. Project report, Norwegian University of Science and Technology, 2014.
- [48] Zhiqiang Xing and Demoz Gebre-Egziabher. Comparing Non-Linear Filters for Aided Inertial Navigators. In *Proceedings of the 2009 International Technical Meeting of The Institute of Navigation*, pages 1048–1053, 2001.
- [49] Yi Yao, Zhi Gang Huang, and Rui Li. Data fusion algorithm of BARO/GPS based on delta-altitude. *Proceedings - 2009 IEEE Youth Conference on Information, Computing and Telecommunication, YC-ICT2009*, pages 522–525, 2009.
- [50] A.S. Zaidi and M.R. Suddle. Global Navigation Satellite Systems : A Survey. *Advances in Space Technologies, 2006 International Conference on 2-3 Sept.*, vol., no.:pp. 84–87, 2006.